Workgroup: TEAS
Internet-Draft:
draft-ietf-teas-ietf-network-slice-nbi-yang-10
Published: 16 March 2024
Intended Status: Standards Track
Expires: 17 September 2024
Authors: B. Wu
Huawei Technologies
D. Dhody
Huawei Technologies
R. Rokui
Ciena
T. Saad
Cisco Systems, Inc
J. Mullooly
Cisco Systems, Inc

A YANG Data Model for the RFC 9543 Network Slice Service

Abstract

This document defines a YANG data model for RFC 9543 Network Slice Service. The model can be used in the Network Slice Service interface between a customer and a provider that offers RFC 9543 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 17 September 2024.

Table of Contents

1. Introduction

This document defines a YANG [RFC7950] data model for [RFC9543] Network Slice Service. [RFC9543] discusses common framework and interface for Network Slice using IETF technologies. The Network Slice Services may be referred to as RFC 9543 Network Slice Services. In this document, we simply use the term "Network Slice Service" to refer to this concept.

The Network Slice Service Model (NSSM) can be used in the 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) Network Slice Services. The agreed service will then trigger the appropriate Network Slice operation, such as instantiating, modifying, or deleting a Network Slice.

The NSSM focuses on the requirements of a Network Slice Service from the point of view of the customer, not how it is implemented within a provider network. The module is classified as customer service model (Section 2 of [RFC8309]). As discussed in [RFC9543], the mapping between a Network Slice Service and its realization is implementation and deployment specific.

The NSSM 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:

* "AAAA" -- 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 [RFC9543]:

*Attachment Circuit (AC): See Section 3.2 of [RFC9543].

*Connectivity Construct: See Sections 3.2 and 4.2.1 of [RFC9543].

*Customer: See Section 3.2 of [RFC9543].

*Customer Higher-level Operation System: See Section 6.3.1 of [RFC9543].

*Service Demarcation Point (SDP): See Sections 3.2 and 5.2 [RFC9543].

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
MTU Maximum Transmission Unit
NSC Network Slice Controller

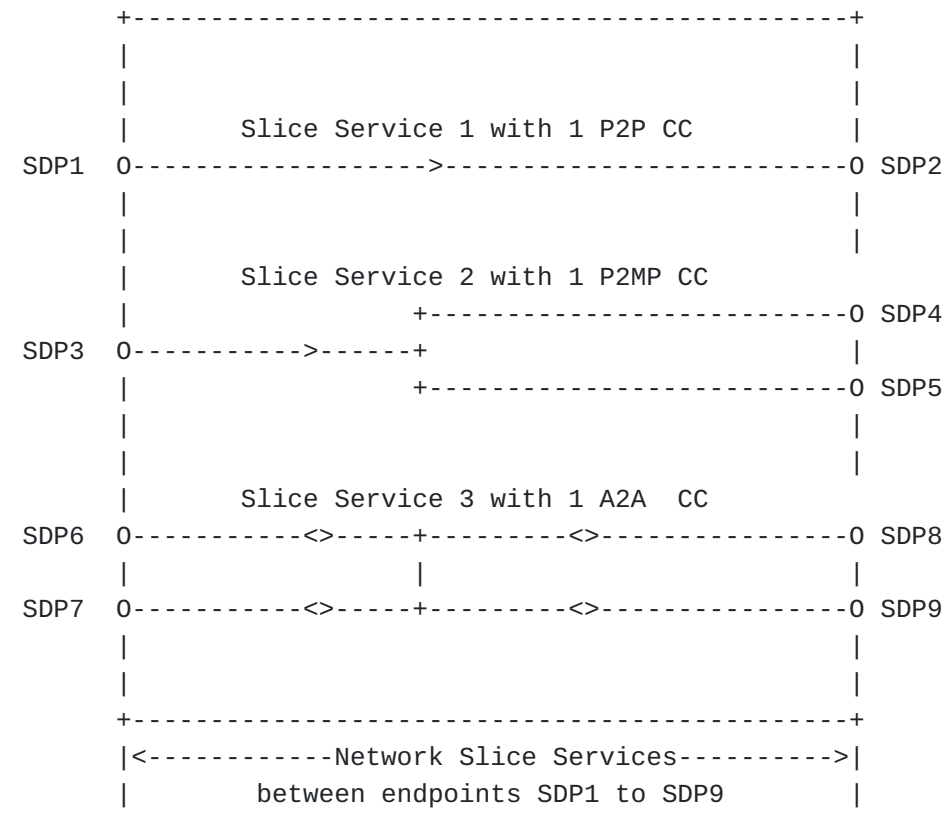
NSSM Network Slice Service Model
P2P Point-to-point
P2MP Point-to-multipoint
PE Provider Edge
QoS Quality of Service
SDP Service Demarcation Point
SLE Service Level Expectation
SLO Service Level Objective

3. Network Slice Service Overview

As defined in Section 3.2 of [RFC9543], a 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 Network Slice Service ingress/egress points. An SDP is identified by a unique identifier in the context of a Network Slice Service.

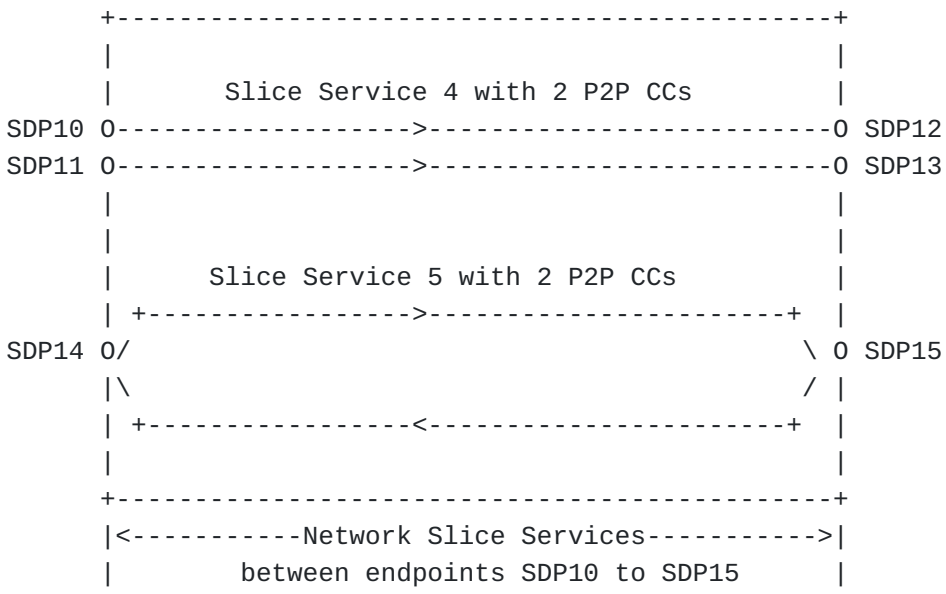
Examples of Network Slice Services that contain only one connectivity construct are shown in Figure 1.



CC: Connectivity construct
O: Represents an SDP
----: Represents connectivity construct
< > : Inbound/outbound directions

Figure 1: Examples of Network Slice Services of Single Connectivity Construct

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



Slice Service: Network Slice Service
CC: Connectivity construct
O: Represents an SDP
----: Represents connectivity construct
< > : Inbound/outbound directions

Figure 2: Examples of Network Slice Services of Multiple Connectivity Constructs

As shown in Figure 2, the Network Slice Service 4 contains two P2P connectivity constructs between the set of SDPs. The Network Slice Service 5 is a bidirectional unicast service between SDP14 and SDP15 that consists of two unidirectional P2P connectivity constructs.

4. Network Slice Service Model (NSSM) Usage

The NSSM can be used by a provider to expose its Network Slice Services, and by a customer to manage its 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 Network Slices realization are hidden from customers.

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

An Network Slice Controller (NSC) is an entity that exposes the Network Slice Service Interface to customers to manage 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 Network Slice Service customer provides, describing the needed Network Slices Service in terms 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 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 Network Slices are put into effect are out of scope for this document.

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

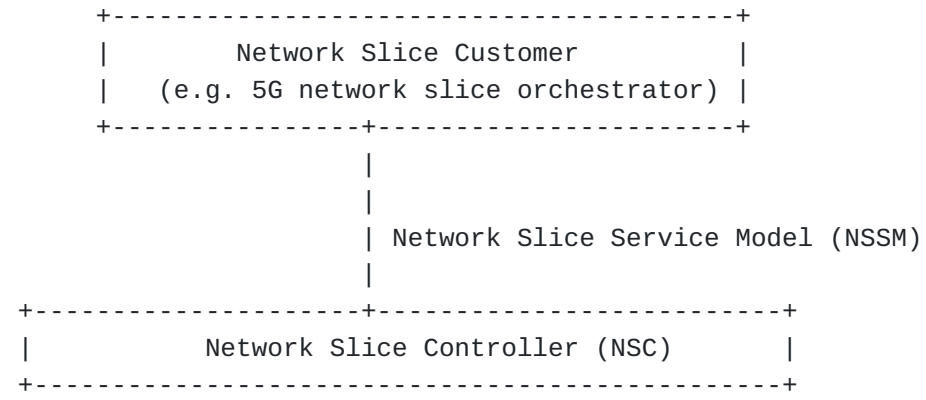


Figure 3: Network Slice Service Reference Architecture

Note: The NSSM can be used recursively (hierarchical mode), i.e., an NSS can map to child NSSes. As described in Section A.5 of [RFC9543], the Network Slice Service can support a recursive composite architecture that allows one layer of Network Slice Services to be used by other layers.

5. Network Slice Service Model (NSSM) Description

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

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 Network Slice Services. Refer to Section 5.1 for further details on the properties of a NSS templates.

The "slice-service" list includes the set of Network Slice Services that are maintained by a provider. "slice-service" is the data structure that abstracts the 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 for further details on the properties of a NSS.

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 Overall Tree Structure

5.1. SLO and SLE Templates

The "slo-sle-templates" container (Figure 5) is used by a Network Slice Service provider to define and maintain a set of common Network Slice Service templates that apply to one or several Network Slice Services. The templates are assumed to be known to both the customers and the provider. The exact definition of the templates is deployment specific to each provider.

```
+--rw slo-sle-templates
|  +--rw slo-sle-template* [id]
|     +--rw id              string
|     +--rw description?    string
|     +--rw template-ref?   slice-template-ref
|     +--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?            uint32
|     +--rw sle-policy
|        +--rw security*              identityref
|        +--rw isolation*             identityref
|        +--rw max-occupancy-level?   uint8
|        +--rw path-constraints
|           +--rw service-functions
|           +--rw diversity
|              +--rw diversity-type?
|                      te-types:te-path-disjointness
```

Figure 5: Slo Sle Templates Subtree Structure

The NSSM provides the identifiers of SLO and SLE templates and the common attributes defined in Section 5.1 of [RFC9543]. Considering that there are many attributes defined and some attributes could vary with service requirements, e.g., bandwidth, or latency, standard templates as well as custom "service-slo-sle-policy" are defined. Customer can choose either a standard template provided by the operator or a custom "service-slo-sle-policy".

1. Standard template: The exact definition of the templates is deployment specific to the provider. The attributes configuration of a standard template is optional. When specifying attributes, a standard template can use "template-ref" to inherit some attributes of the predefined standard template and override the specific attributes.

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

Figure 6 shows an example where two standard network slice templates can be retrieved by the customers.

```
========== 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": "ietf-nss:two-way-delay-percentile",
                "metric-unit": "milliseconds",
                "percentile-value": "95.000",
                "bound": "50"
              }
            ]
          },
          "sle-policy": {
            "isolation": ["ietf-nss:traffic-isolation"]
          }
        },
        {
          "id": "GOLD-template",
          "description": "Two-way bandwidth: 1 Gbps,\
                          maximum latency 100ms",
          "slo-policy": {
            "metric-bound": [
              {
                "metric-type": "ietf-nss:two-way-delay-maximum",
                "metric-unit": "milliseconds",
                "bound": "100"
              }
            ]
          },
          "sle-policy": {
            "isolation": ["ietf-nss:traffic-isolation"]
          }
        }
      ]
    }
  }
}
```

Figure 6

Figure 6 use folding as defined in [RFC8792] for long lines.

5.2. Network Slice Services

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

A Network Slice Service has the following main data nodes:

* "description": Provides a textual description of an 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 Network Slices. It might be used by a Network Slice Service provider to provide additional information to an NSC during the operation of the Network Slices. E.g. adding tags with "customer-name" when multiple actual customers use a same Network Slice Service. Another use-case for "service-tag" might be for a provider to provide additional attributes to an NSC which might be used during the realization of Network Slice Services such as type of services (e.g., Layer 2 or Layer 3 technology). These additional attributes can also be used by an NSC for various purposes such as monitoring and assurance of the Network Slice Services where the NSC can issue notifications to the customer system. Note that all these attributes are optional.

* "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 both operational and administrative status of a Network Slice Service. It can be used as indicator to detect Network Slice Service anomalies.

* "sdps": Represents a set of SDPs that are involved in the Network Slice Service. More details are provided in Section 5.2.1.

* "connection-groups": Abstracts the connections to the set of SDPs of the Network Slice Service.

* "custom-topology": Represents custom topology constraints for the Network Slice Service. More details are provided in Section 5.2.5

5.2.1. Service Demarcation Points

A Network Slice Service involves two or more SDPs. A Network Slice Service can be modified by adding new "sdp"s.

```
+--rw sdps
   +--rw sdp* [id]
      +--rw id                          string
      +--rw description?                string
      +--rw geo-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 ce-mode?                    boolean
      +--rw attachment-circuits
      |     ...
      +--rw status
      |     ...
      +--ro sdp-monitoring
            ...
```

Figure 7: SDP Subtree Structure

Section 5.2 of [RFC9543] 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.

In the four options, the Attachment Circuit (AC) may be part of the Network Slice Service or may be external to it. Based on the AC

definition in Section 5.2 of [RFC9543], the customer and provider may agree on a per {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 a 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:

* "id": Uniquely identifies the SDP within an NSC. The identifier is a string that allows any encoding for the local administration of the Network Slice Service.

* "geo-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. This document assumes that higher-level systems can obtain the node information, PE and CE, prior to the service requests. For example, SAP Network [RFC9408] can obtain PE-related node information. The implementation details are left to the NSC provider.

* "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.

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

* "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 [RFC9543].

*"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. As shown in Figure 8

```
+--rw sdp-peering
|  +--rw peer-sap-id*   string
|  +--rw protocols
```

Figure 8: SDP Peering Subtree Structure

-"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 gives the example protocols of BGP, static routing, etc.

*"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.

*"ce-mode": A flag node that marks the SDP as CE type.

*"attachment-circuits": Specifies the list of ACs by which the service traffic is received. 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.

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

*"sdp-monitoring": Provides SDP bandwidth statistics.

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 9. Traffic needs to be explicitly mapped into the 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 {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 are no additional connectivity constructs where the "sdp-id" source is specified, then that traffic will be dropped.

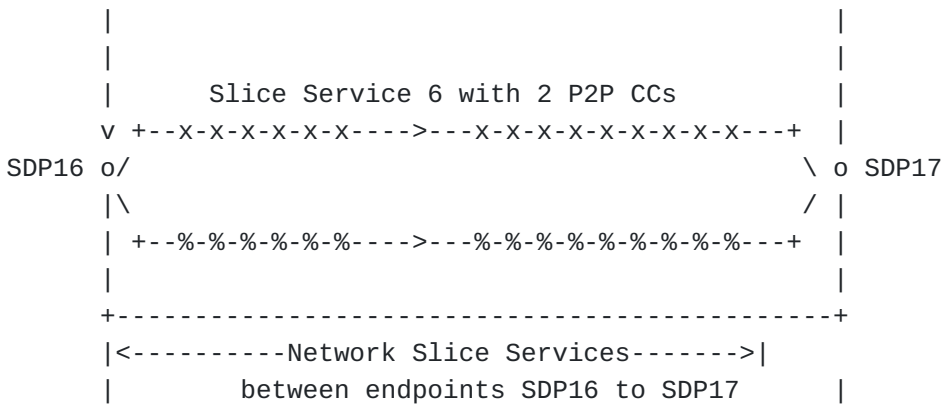
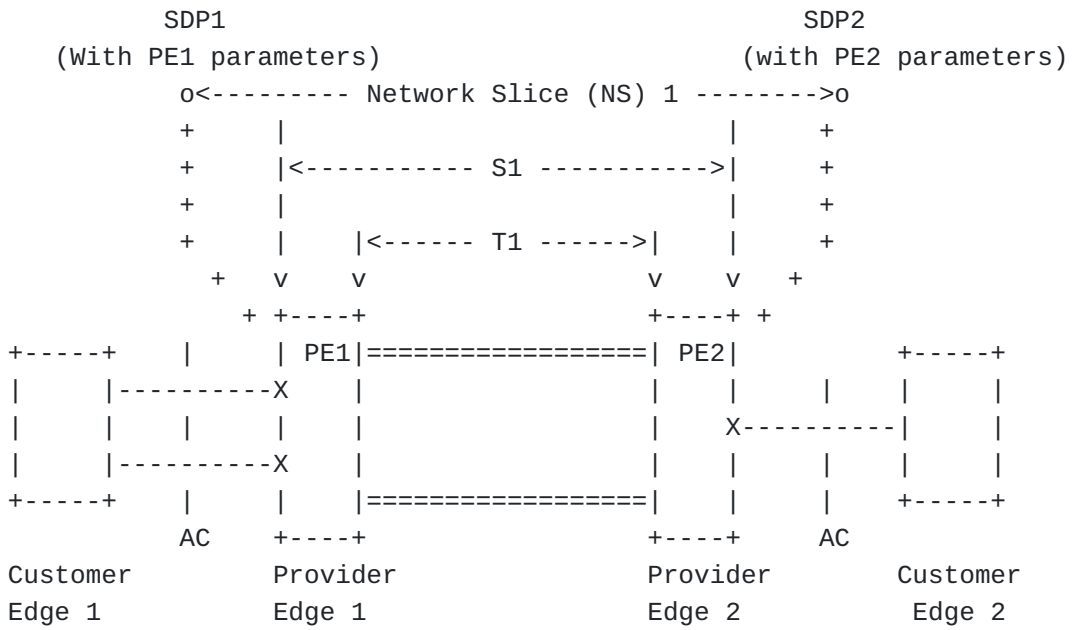


Figure 9: Application of Match Criteria

If an SDP is placed at the port of a CE or PE, and there is only one single connectivity construct with a source at the SDP, traffic can be implicitly mapped to this connectivity construct since the AC information (e.g., VLAN tag) 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, Figure 10 and Figure 11 are two examples. How an NSC realize the mapping is out of scope for this document.

*SDPs at customer-facing ports on the PEs: As shown in Figure 10 , a customer of the Network Slice Service would like to connect two SDPs to satisfy specific service needs, e.g., network wholesale services. In this case, the Network Slice SDPs are mapped to customer-facing ports of PE nodes. The NSC uses "node-id" (PE device ID), "attachment-circuits" (ACs) or "ac-svc-name" to map SDPs to the customer-facing ports on the PEs.

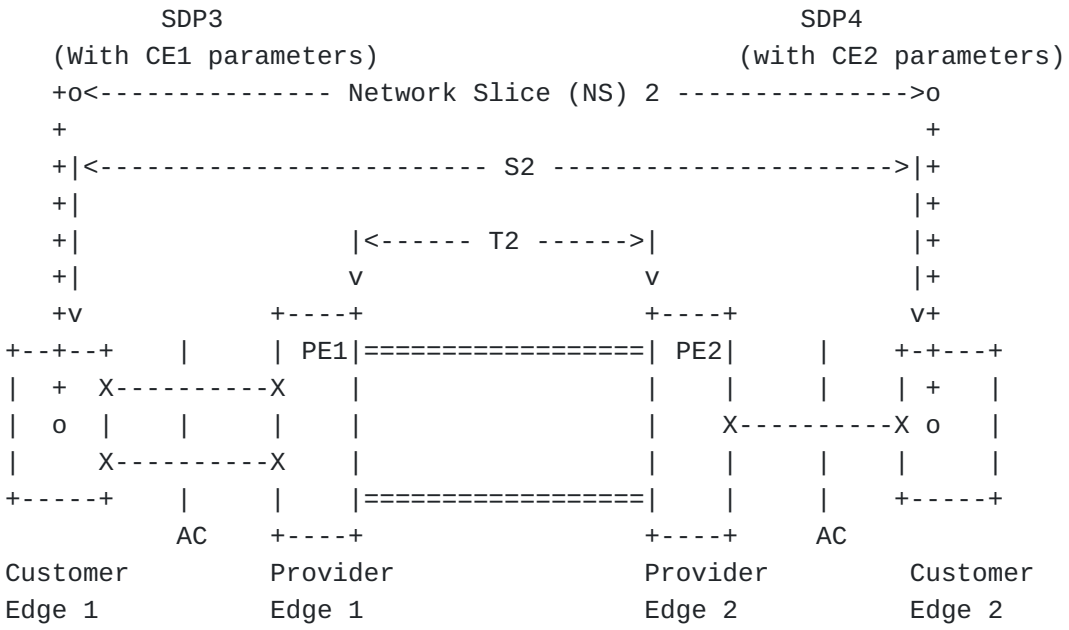


Legend:
o: Representation of an SDP
+: Mapping of an SDP to customer-facing ports on the PE
X: Physical interfaces used for realization of the NS Service
S1: L0/L1/L2/L3 services used for realization of NS Service
T1: Tunnels used for realization of NS Service

Figure 10: An Example of SDPs Placing at PEs

*SDPs within CEs: As shown in Figure 11 , a customer of the 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), "geo-location", "sdp-ip-address" (IP address of SDP for management), "service-match-criteria" (VLAN tag), "attachment-circuits" or or "ac-svc-name" (CE ACs) to map SDPs to the CE. The NSC can use these CE parameters (and optionally other information to uniquely identify a CE within an NSC, such as

"peer-sap-id" [RFC9408]) to retrieve the corresponding PE device, interface and AC mapping details to complete the Network Slice Service provisioning.



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

Figure 11: An Example of SDPs Placing at CEs

5.2.2. Connectivity Constructs

Section 4.2.1 of [RFC9543] defines the basic connectivity construct (CC) and CC types of a Network Slice Service, including P2P, P2MP, and A2A.

A Network Slice Service involves one or more connectivity constructs. The "connection-groups" container is used to abstract CC, CC groups, and their SLO-SLE policies and the structure is shown in Figure 12.

```
+--rw connection-groups
   +--rw connection-group* [id]
      +--rw id                                    string
      +--rw connectivity-type?
      |       identityref
      +--rw (slo-sle-policy)?
      |  +--:(standard)
      |  |     ...
      |  +--:(custom)
      |        ...
      +--rw service-slo-sle-policy-override?
      |       identityref
      +--rw connectivity-construct* [id]
      |  +--rw id
      |  |       uint32
      |  +--rw (type)?
      |  |     ...
      |  +--rw (slo-sle-policy)?
      |  |     ...
      |  +--rw service-slo-sle-policy-override?
      |  |       identityref
      |  +--rw status
      |  |     ...
```

Figure 12: Connection Groups Subtree Structure

The description of the "connection-groups" data nodes is as follows:

* "connection-group": Represents a group of CCs. In the case of hub and spoke connectivity of the Slice Service, it may be inefficient when there are a large number of 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]. Another use is for optimizing "slo-sle-policy" configurations, treating CCs with the same SLO and SLE characteristics as a connection group such that the connectivity construct can inherit the SLO/SLE from the group if not explicitly defined.

* "connectivity-type": Indicates the type of the connection group, extending "vpn-common:vpn-topology" specified [RFC9181] with the NS connectivity type, e.g., P2P and P2MP.

* "connectivity-construct": Represents single connectivity construct, and "slo-sle-policy" under it represents the per-connectivity construct SLO and SLE requirements.

*"slo-sle-policy" and "service-slo-sle-policy-override": The details of "slo-sle-policy" is defined in Section 5.2.3. In addition to "slo-sle-policy" nodes of "connection-group" and "connectivity-construct", a leaf node "service-slo-sle-policy-override" is provided for scenarios with complex SLO-SLE requirements to completely override all or part of an "slo-sle-policy" 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.

5.2.3. SLO and SLE Policy

As defined in Section 5 of [RFC9543], the SLO and SLE policy of the Network Slice Services define some common attributes.

"slo-sle-policy" is used to represent these SLO and SLE policies. During the creation of a Network Slice Service, the policy can be specified either by a standard SLO and SLE 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 include performance metric attributes, availability, and MTU. The SLO structure is shown in Figure 13.

```
+--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
```

Figure 13: SLO Policy Subtree Structure

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 the bandwidth is unidirectional.

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

"one-way-delay-maximum": Indicates the maximum one-way latency between two SDPs, defined in [RFC7679].

"two-way-delay-maximum": Indicates the maximum round-trip latency between two SDPs, defined in [RFC2681]. .

"one-way-delay-percentile": Indicates the percentile objective of the one-way latency between two SDPs (See [RFC7679]).

"two-way-delay-percentile": Indicates the percentile objective of the round-trip latency between two SDPs (See [RFC2681])..

"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, as defined in [RFC3393]

"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, as defined in [RFC3393].

"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, as defined in [RFC3393].

"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, as defined in [RFC5481].

"one-way-packet-loss": Indicates maximum permissible packet loss rate (See [RFC7680], 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 (See [RFC7680], 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 Network Slice is available in accordance with the SLOs associated with it.

"mtu": Refers to the service MTU. If the customer sends packets that are longer than the requested service MTU, the network may discard it (or for IPv4, fragment it). Depending on the service layer, the value can be an L3 service MTU (Section 7.6.6 [RFC9182]) or an L2 service MTU (Section 7.4 [RFC9291]).

As shown in Figure 14, the following SLEs data nodes are defined.

"security": The security leaf-list defines the list of security functions the customer requests the operator to apply to traffic between the two SDPs, including authentication, encryption, etc, which is defined in Section 5.1.2.1 [RFC9543].

"isolation": Specifies the isolation types that a customer expects, as defined in Section 8 [RFC9543].

"max-occupancy-level": Specifies the number of flows that the operator admits (See Section 5.1.2.1 [RFC9543]).

"path-constraints": Specifies the path constraints the customer requests for the Network Slice Service, including geographic restrictions and diversity which is defined in Section 5.1.2.1 [RFC9543].

```
+--rw sle-policy
   +--rw security*              identityref
   +--rw isolation*             identityref
   +--rw max-occupancy-level?   uint8
   +--rw path-constraints
      +--rw service-functions
      +--rw diversity
         +--rw diversity-type?
                 te-types:te-path-disjointness
```

Figure 14: SLE Policy Subtree Structure

Figure 15 shows an example with a network slice "slo-policy".

```
{
  "slice-services": {
    "slice-service": {
      "id": "exp-slice",
      "service-slo-sle-policy": {
        "description": "video-service-policy",
        "slo-policy": {
          "metric-bound": [
            {
              "metric-type": "ietf-nss:one-way-bandwidth",
              "metric-unit": "Mbps",
              "bound": "1000"
            },
            {
              "metric-type": "ietf-nss:two-way-delay-maximum",
              "metric-unit": "milliseconds",
              "bound": "10"
            }
          ],
          "availability": "ietf-nss:level-4",
          "mtu": "1500"
        }
      }
    }
  }
}
```

Figure 15: An Example of a Slice Service of SLO Policies

5.2.4. Network Slice Service Performance Monitoring

The operation and performance status of Network Slice Services is also a key component of the NSSM. The model provides SLO monitoring information with the following granularity:

- Per SDP: The incoming and outgoing bandwidths of an SDP are specified in "sdp-monitoring" under the "sdp".

- Per connectivity construct: The delay, delay variation, and packet loss status are specified in "connectivity-construct-monitoring" under the "connectivity-construct".

- Per connection group: The delay, delay variation, and packet loss status are specified in "connection-group-monitoring" under the "connection-group".

[RFC8639] and [RFC8641] define a subscription mechanism and a push mechanism for YANG datastores. These mechanisms currently allow the user to subscribe to notifications on a per-client basis and specify either periodic or on-demand notifications. By specifying subtree filters or xpath filters to "sdp", "connectivity-construct", or "connection-group", so that only interested contents will be sent. The example in Figure 24 shows the way for a customer to subscribe to the monitoring information for a particular Network Slice Service. .

Additionaly, a customer can use the NSSM to obtain a snapshot of the Network Slice Service performance status through [RFC8040] or [RFC6241] interfaces. For example, retrieve the per-connectivity-construc data by specifying "connectivity-construct" as the filter in the RESTCONF GET request.

5.2.5. Custom Topology Constraints

The Slice Service customer might 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. Section 1 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 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 serves as an augmentation target for the Slice Service topology context, which can be multiple. This node is located directly under the "slice-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 a Network Slice Service, e.g., a Service Attachment Points (SAPs) topology to request SDP feasibility checks on a SAPs network described in Section 3 of [RFC9408], an abstract Traffic Engineering (TE) topology defined in section-3.13 of [RFC8795] to customize the service paths in a Network Slice Service.

"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 example TPs could be parent termination points of the SAP topology.

5.2.6. Network Slice Service Compute

A Network Slice Service is, by default, provisioned so that it can instantiate and trigger service delivery. A Network Slice Service customer may request to check the feasibility of a request before instantiating or modifying a Network Slice Service . In such a case, the 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 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 "admin-status" and the connection groups or connectivity construct list are used to convey the result. For example, "admin-compute-only" can be used to show the status. Customers can query the "compute-only" connectivity constructs attributes, or can subscribe to be notified when the connectivity constructs status change.

The "compute-only" applies only if the data model is used with a protocol that does not natively support such operation, e.g. [RFC8040]. When using NETCONF, <edit-config> operation (Section 7.2 of [RFC6241]), "test-only" of the <test-option> parameter also applies.

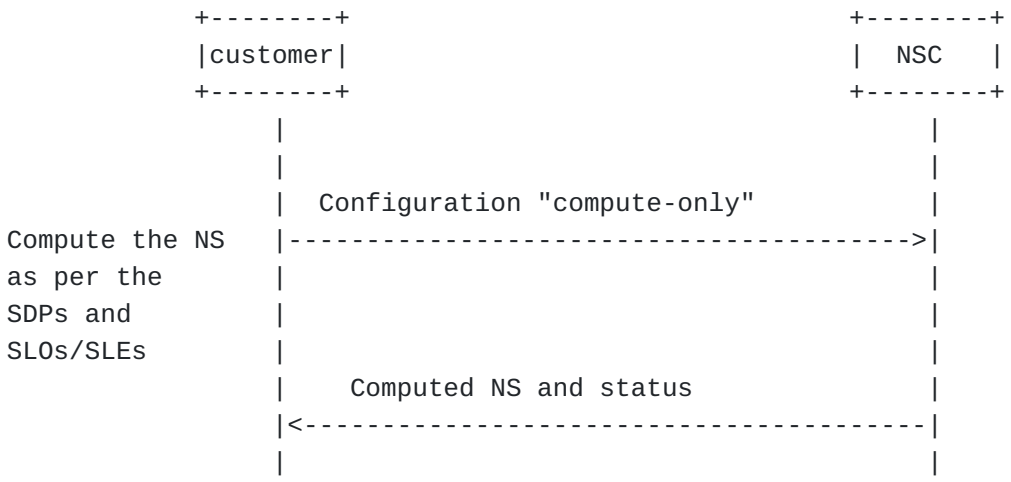


Figure 16: An Example of Network Slice Service Compute

6. Network Slice Service Module

The "ietf-network-slice-service" module uses types defined in [RFC6991], [RFC8345], [RFC9179], [RFC9181], [RFC8776], and [RFC7640].

```
<CODE BEGINS> file "ietf-network-slice-service@2024-03-17.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-yang-types {
    prefix yang;
    reference
      "RFC 6991: Common YANG Data Types";
  }
  import ietf-geo-location {
    prefix geo;
    reference
      "RFC 9179: A YANG Grouping for Geographic Locations";
  }
  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
      "RFC 8345: A YANG Data Model for Network
                 Topologies, Section 6.2";
  }
  import ietf-te-types {
    prefix te-types;
    reference
      "RFC 8776: Traffic Engineering Common YANG Types";
  }

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

```
contact
  "WG Web:  <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 RFC 9543 Network Slice
   Service.

   Copyright (c) 2024 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 AAAA; see the
   RFC itself for full legal notices.";

revision 2024-03-17 {
  description
    "Initial revision.";
  reference
    "RFC AAAA: A YANG Data Model for Network Slice Service";
}

/* Identities */

identity service-tag-type {
  description
    "Base identity of Network Slice Service tag type.";
}

identity service-tag-customer {
  base service-tag-type;
  description
    "The Network Slice Service customer name tag type,
     e.g. adding tags with 'customer-name' when multiple actual
```

```
     customers use a same Network Slice Service.";
}

identity service-tag-service {
  base service-tag-type;
  description
    "The Network Slice Service tag type, which can indicate the
     technical constraints used during service realization,
     for example, Layer 2 or Layer 3 technologies.";
}

identity service-tag-opaque {
  base service-tag-type;
  description
    "An opaque type, which can be used for future use,
     such as filtering of services.";
}

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, 802.1Q dot1Q.";
  reference
    "IEEE Std 802.1Q: IEEE Standard for Local and Metropolitan
                      Area Networks--Bridges and Bridged
                      Networks";
}

identity cvlan-id {
  base attachment-circuit-tag-type;
  description
    "Identity for C-VLAN ID tag type, 802.1ad QinQ VLAN IDs.";
  reference
    "IEEE Std 802.1ad: IEEE Standard for Local and Metropolitan
                       Area Networks---Virtual Bridged Local
                       Area Networks---Amendment 4: Provider
                       Bridges";
}

identity svlan-id {
  base attachment-circuit-tag-type;
  description
    "Identity for S-VLAN ID tag type, 802.1ad QinQ VLAN IDs.";
  reference
```

```
    "IEEE Std 802.1ad: IEEE Standard for Local and Metropolitan
                       Area Networks---Virtual Bridged Local
                       Area Networks---Amendment 4: Provider
                       Bridges";
}

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

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

identity traffic-isolation {
  base service-isolation-type;
  description
    "Specify the requirement for separating the traffic of the
     customer's 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
    "Base identity for 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 Network Slice
     Service connectivity.";
}

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

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

identity multipoint-to-point {
  base vpn-common:vpn-topology;
  description
    "Identity for multipoint-to-point 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 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
    "RFC7679: 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
    "RFC7679: 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
    "RFC2681: A Round-trip Delay Metric for IPPM";
}

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
    "RFC2681: 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
    "RFC3393: 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
    "RFC3393: 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
    "RFC5481: 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
    "RFC5481: 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.";
  reference
    "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.";
  reference
    "RFC7680: A One-Way Loss Metric for IP Performance
     Metrics (IPPM)";
}

/*
 * Identity for availability-type
 */

identity availability-type {
  description
    "Base identity for availability.";
}

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 Network Slice Service traffic
     match type.";
}

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

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

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

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

identity 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 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 full-override {
  base slo-sle-policy-override;
  description
    "The SLO/SLE policy defined at the child level overrides a
     parent SLO/SLE policy, which means that no SLO/SLEs are
     inherited from parent if a child SLO/SLE policy exists.";
}

identity partial-override {
  base slo-sle-policy-override;
  description
    "The SLO/SLE policy defined at the 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 percentage {
  type decimal64 {
    fraction-digits 5;
    range "0..100";
  }
  description
    "Percentage to 5 decimal places.";
}

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.";
}

typedef slice-template-ref {
  type leafref {
    path "/ietf-nss:network-slice-services"
       + "/ietf-nss:slo-sle-templates"
       + "/ietf-nss:slo-sle-template"
       + "/ietf-nss:id";
  }
  description
    "This type is used by data models that need to reference
     Network Slice template.";
}

/* Groupings */

grouping service-slos {
  description
    "A reusable grouping for 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 SLO metric type of the Slice Service.";
  }
  leaf metric-unit {
    type string;
    mandatory true;
    description
      "The metric unit of the parameter. For example,
       for time units, where the options are hours, minutes,
       seconds, milliseconds, microseconds, and nanoseconds;
       for bandwidth units, where the options are bps, Kbps,
       Mbps, Gbps; for the packet loss rate unit,
       the options can be percentage.";
  }
  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;
    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 uint32;
  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
    "A reusable grouping for indirectly measurable objectives of
     a Slice Service.";
  container sle-policy {
    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 and optionally a maximum number of
         countable resource units (e.g., IP or MAC addresses)
         a Network Slice Service can consume.";
    }
    container path-constraints {
      description
        "Container for the policy of path constraints
         applicable to the Slice Service.";
      container service-functions {
        description
          "Container for the policy of service function
           applicable to the Slice Service.";
```

```
      }
      container diversity {
        description
          "Container for the policy of disjointness
           applicable to the Slice Service.";
        leaf diversity-type {
          type te-types:te-path-disjointness;
          description
            "The type of disjointness on Slice Service, i.e.,
             across all connectivity constructs.";
        }
      }
    }
  }
}

grouping slice-service-template {
  description
    "A reusable 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 slice-template-ref;
        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;
    }
  }
```

```
}

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 slice-template-ref;
        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 bw-rate-limits {
  description
    "Grouping for bandwidth rate limits.";
  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
    "Grouping for the Slice Service QoS policy.";
  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 classes {
      description
        "Container for service class bandwidth control.";
      list cos {
        key "cos-id";
        description
          "List of Class of Services.";
        leaf cos-id {
          type uint8;
          description
            "Identifier of the CoS, indicated by
             a Differentiated Services Code Point
             (DSCP) or a CE-CLAN CoS (802.1p)
             value in the service frame.";
          reference
            "IEEE Std 802.1Q: Bridges and Bridged
                              Networks";
        }
        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;
      container classes {
        description
          "Container for classes.";
        list cos {
          key "cos-id";
          description
            "List of Class of Services.";
          leaf cos-id {
            type uint8;
            description
              "Identifier of the CoS, indicated by
               a Differentiated Services Code Point
               (DSCP) or a CE-CLAN CoS (802.1p)
               value in the service frame.";
            reference
              "IEEE Std 802.1Q: Bridges and Bridged
                                Networks";
          }
          uses bw-rate-limits;
        }
      }
    }
  }
}

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:full-override";
    description
      "SLO/SLE policy override option.";
  }
}

grouping one-way-performance-metrics {
  description
    "One-way PM metrics grouping.";
  leaf one-way-min-delay {
    type yang:gauge64;
    description
      "One-way minimum delay or latency in microseconds.";
  }
  leaf one-way-max-delay {
    type yang:gauge64;
```

```
    description
      "One-way maximum delay or latency in microseconds.";
    reference
      "RFC7679: A One-Way Delay Metric for IP Performance
       Metrics (IPPM)";
  }
  leaf one-way-delay-variation {
    type yang:gauge64;
    description
      "One-way delay variation in microseconds.";
    reference
      "RFC3393: IP Packet Delay Variation Metric for IP Performance
       Metrics (IPPM)";
  }
  leaf one-way-packet-loss {
    type percentage;
    description
      "The ratio of packets dropped to packets transmitted between
       two endpoints.";
    reference
      "RFC7680: A One-Way Loss Metric for IP Performance
       Metrics (IPPM)";
  }
}

grouping two-way-performance-metrics {
  description
    "Two-way packet PM metrics grouping.";
  leaf two-way-min-delay {
    type yang:gauge64;
    description
      "Two-way minimum delay or latency in microseconds.";
    reference
      "RFC2681: A Round-trip Delay Metric for IPPM";
  }
  leaf two-way-max-delay {
    type yang:gauge64;
    description
      "Two-way maximum delay or latency in microseconds.";
    reference
      "RFC2681: A Round-trip Delay Metric for IPPM";
  }
  leaf two-way-delay-variation {
    type yang:gauge64;
    description
      "Two-way delay variation in microseconds.";
    reference
      "RFC5481: Packet Delay Variation Applicability Statement";
  }
```

```
  leaf two-way-packet-loss {
    type percentage;
    description
      "The ratio of packets dropped to packets transmitted between
       two endpoints.";
  }
}

grouping connectivity-construct-monitoring-metrics {
  description
    "Grouping for connectivity construct monitoring metrics.";
  uses one-way-performance-metrics;
  uses two-way-performance-metrics;
}

/* Main Network Slice Services Container */

container network-slice-services {
  description
    "Contains a list of Network Slice Services";
  uses slice-service-template;
  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 within an NSC.";
    }
    leaf description {
      type string;
      description
        "Textual description of the Slice Service.";
    }
    container service-tags {
      description
        "Container for a list of service tags for management
         purposes, such as policy constraints
         (e.g., Layer 2 or Layer 3 technology realization),
         classification (e.g., customer names, opaque values).";
      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, e.g. realization technology
         constraints, customer name, or other customer-defined
         opaque types.";
    }
    leaf-list value {
      type string;
      description
        "The tag values, e.g., 5G customer names when multiple
         customers share the same Slice Service in 5G scenario,
         or Slice realization technology (such as Layer 2 or
         Layer 3).";
    }
  }
}
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;
    description
      "List of SDPs in this Slice Service.";
    leaf id {
      type string;
      description
        "The unique identifier of the SDP within the scope of
         an NSC.";
    }
    leaf description {
      type string;
      description
        "Provides a description of the SDP.";
    }
    uses geo:geo-location;
    leaf node-id {
      type string;
      description
        "A unique identifier of an edge node of the SDP
         within the scope of the NSC.";
    }
```

```
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 of 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 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:connection-groups"
        + "/ietf-nss:connection-group[ietf-nss: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 SAPs
       of provider equipments when requesting a service with
       CE based SDP attributes.";
    reference
      "RFC 9408: 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, 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.";
  reference
    "draft-ietf-opsawg-teas-attachment-circuit-02:
     YANG Data Models for
     'Attachment Circuits'-as-a-Service (ACaaS)";
}
leaf ce-mode {
  type boolean;
  description
    "Indicates that SDP is on the CE.";
}
container attachment-circuits {
  description
    "List of attachment circuits.";
  list attachment-circuit {
    key "id";
    description
      "The Network Slice Service SDP attachment circuit
       related parameters.";
    leaf id {
      type string;
      description
        "The identifier of attachment circuit.";
    }
    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.";
    reference
      "draft-ietf-opsawg-teas-attachment-circuit-02:
       YANG Data Models for
       'Attachment Circuits'-as-a-Service (ACaaS)";
  }
  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 {
    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 uint32;
    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-tag {
```

```
        key "tag-type";
        description
          "The attachment circuit tag 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
             multiple VLAN identifiers.";
        }
      }
    }
    uses service-qos;
    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
          "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.";
      }
    }
    uses vpn-common:service-status;
  }
}
uses vpn-common:service-status;
container sdp-monitoring {
  config false;
```

```
      description
        "Container for SDP monitoring metrics.";
      leaf incoming-bw-value {
        type yang:gauge64;
        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 percentage;
        units "percent";
        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 yang:gauge64;
        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 percentage;
        units "percent";
        description
          "Indicates a percentage of the outgoing bandwidth
           at an SDP towards the customer network or towards
           another provider's network.";
      }
    }
  }
}
container connection-groups {
  description
    "Contains connection groups.";
  list connection-group {
    key "id";
    description
      "List of connection groups.";
    leaf id {
      type string;
      description
        "The connection group identifier.";
    }
```

```
leaf connectivity-type {
  type identityref {
    base vpn-common:vpn-topology;
  }
  default "vpn-common:any-to-any";
  description
    "Connection group connectivity type.";
}
uses service-slo-sle-policy;
/* Per connection group SLO/SLE policy
 * overrides the per Slice SLO/SLE policy.
 */
uses service-slo-sle-policy-override;
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 {
```

```
          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 SLO/SLE policy
   * overrides the per slice 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;
  }
}
container connection-group-monitoring {
  config false;
  description
    "SLO status per connection group.";
  uses connectivity-construct-monitoring-metrics;
```

```
          }
        }
      }
      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>

Figure 17: Network Slice Service YANG Module

7. Security Considerations

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

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in these YANG modules 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) and delete operations to these data nodes without proper protection or authentication can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/ vulnerability in the "ietf-network-slice-service" module:

* /ietf-network-slice-service/network-slice-services/slo-sle-templates

This subtree specifies the Network Slice Service SLO templates and SLE templates. Modifying the configuration in the subtree will change the related Network Slice Service configuration in the future. By making such modifications, a malicious attacker may degrade the Slice Service functions configured at a certain time in the future.

* /ietf-network-slice-service/network-slice-services/slice-service

The entries in the list above include the whole network configurations corresponding with the Network 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.

Some of the readable data nodes in these YANG modules may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and

data nodes and their sensitivity/vulnerability in the "ietf-network-slice-service" module:

* /ietf-network-slice-service/network-slice-services/slo-sle-templates

Unauthorized access to the subtree may disclose the SLO and SLE templates of the Network Slice Service.

* /ietf-network-slice-service/network-slice-services/slice-service

Unauthorized access to the subtree may disclose the operation status information of the Network Slice Service.

8. IANA Considerations

This document request to register the following URI in the IETF XML registry [RFC3688]:

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 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-ser
Prefix: ietf-nss
Maintained by IANA: N
Reference: RFC AAAA
```

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>.

[RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Notifications", RFC 8639, DOI 10.17487/RFC8639, September 2019, <https://www.rfc-editor.org/info/rfc8639>.

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

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

[RFC9179] Hopps, C., "A YANG Grouping for Geographic Locations", RFC 9179, DOI 10.17487/RFC9179, February 2022, <https://www.rfc-editor.org/info/rfc9179>.

[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>.

[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>.

[RFC9543]
Farrel, A., Ed., Drake, J., Ed., Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "A Framework for Network Slices in Networks Built from IETF Technologies", RFC 9543, DOI 10.17487/RFC9543, March 2024, <https://www.rfc-editor.org/info/rfc9543>.

11.2. Informative References

[I-D.ietf-opsawg-teas-attachment-circuit]
Boucadair, M., Roberts, R., Dios, O. G. D., Giraldo, S. B., and B. Wu, "YANG Data Models for Bearers and 'Attachment Circuits'-as-a-Service (ACaaS)", Work in Progress, Internet-Draft, draft-ietf-opsawg-teas-attachment-circuit-08, 16 March 2024, <https://datatracker.ietf.org/api/v1/doc/document/draft-ietf-opsawg-teas-attachment-circuit/>.

[I-D.ietf-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-ietf-opsawg-teas-common-ac-05, 9 February 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-opsawg-teas-common-ac-05>.

[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-24, 16 March 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-teas-actn-vn-yang-24>.

[I-D.ietf-teas-ietf-network-slice-use-cases]
Contreras, L. M., Homma, S., Ordonez-Lucena, J. A., Tantsura, J., and H. Nishihara, "IETF Network Slice Use Cases and Attributes for the Slice Service Interface of IETF Network Slice Controllers", Work in Progress, Internet-Draft, draft-ietf-teas-ietf-network-slice-use-cases-01, 24 October 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-teas-ietf-network-slice-use-cases-01>.

[RFC2681] Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, DOI 10.17487/RFC2681, September 1999, <https://www.rfc-editor.org/info/rfc2681>.

[RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC

3393, DOI 10.17487/RFC3393, November 2002, <https://www.rfc-editor.org/info/rfc3393>.

[RFC5481] Morton, A. and B. Claise, "Packet Delay Variation Applicability Statement", RFC 5481, DOI 10.17487/RFC5481, March 2009, <https://www.rfc-editor.org/info/rfc5481>.

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

[RFC7679] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <https://www.rfc-editor.org/info/rfc7679>.

[RFC7680] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Loss Metric for IP Performance Metrics (IPPM)", STD 82, RFC 7680, DOI 10.17487/RFC7680, January 2016, <https://www.rfc-editor.org/info/rfc7680>.

[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>.

[RFC8792] Watsen, K., Auerswald, E., Farrel, A., and Q. Wu, "Handling Long Lines in Content of Internet-Drafts and RFCs", RFC 8792, DOI 10.17487/RFC8792, June 2020, <https://www.rfc-editor.org/info/rfc8792>.

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

[RFC9182] Barguil, S., Gonzalez de Dios, O., Ed., Boucadair, M., Ed., Munoz, L., and A. Aguado, "A YANG Network Data Model for Layer 3 VPNs", RFC 9182, DOI 10.17487/RFC9182, February 2022, <https://www.rfc-editor.org/info/rfc9182>.

[RFC9291] Boucadair, M., Ed., Gonzalez de Dios, O., Ed., Barguil, S., and L. Munoz, "A YANG Network Data Model for Layer 2 VPNs", RFC 9291, DOI 10.17487/RFC9291, September 2022, <https://www.rfc-editor.org/info/rfc9291>.

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.ietf-opsawg-teas-common-ac].

```
module: ietf-network-slice-service-proto-ex
  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
    |  +--rw name?             string
    |  +--ro local-as?         inet:as-number
    |  +--rw peer-as?          inet:as-number
    |  +--rw address-family?   identityref
    +--rw static-routing-ipv4
    |  +--rw lan?        inet:ipv4-prefix
    |  +--rw lan-tag?    string
    |  +--rw next-hop?   union
    |  +--rw metric?     uint32
    +--rw static-routing-ipv6
       +--rw lan?        inet:ipv6-prefix
       +--rw lan-tag?    string
       +--rw next-hop?   union
       +--rw metric?     uint32
```

Figure 18: Example YANG Tree Augmenting SDP Peering Protocols

In some scenarios, for example, when multiple Slice Services share one or more ACs, independent AC services, defined in [I-D.ietf-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
     dedicated network resources, similar to a dedicated
     private network.";
}
```

Figure 19: Example "isolation" Identity Augmentation

Appendix B. Examples of Network Slice Services

B.1. Example-1: Two A2A Slice Services with Different Match Approaches

Figure 20 shows an example of two Network Slice Service instances where the SDPs are the customer-facing ports on the PE:

* 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.

* 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. The "slo-sle-templates" templates are known to the customer.

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.

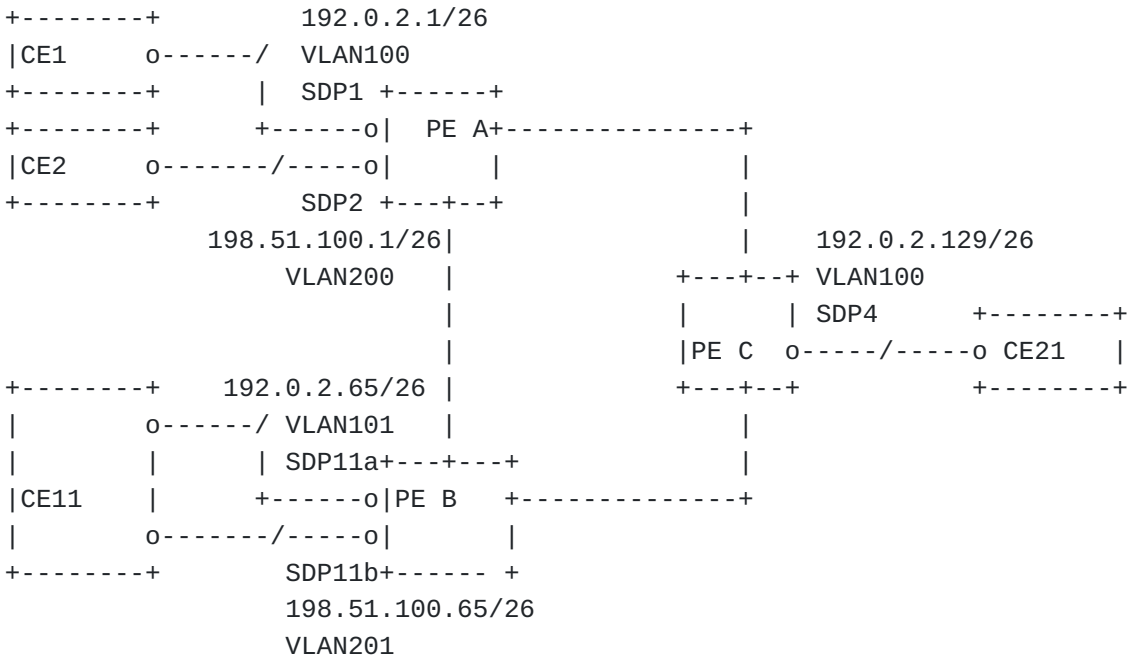


Figure 20: Example of Two A2A Slice Services

Figure 21 shows an example YANG JSON data for the body of the Network Slice Service instances request.

```
{
  "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": {},
        "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-tag": [
            {
              "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-tag": [
            {
              "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-tag": [
              {
                "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-tag": [
            {
              "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-tag": [
            {
              "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": {}
              }
            ]
          }
        ]
      }
    }
  ]
}
```

Figure 21: Example of a Message Body to Create Two A2A Slice Services

B.2. Example-2: Two P2P Slice Services with Different Match Approaches

Figure 22 shows an example of two Network Slice Service instances where the SDPs are the customer-facing ports on the PE:

- *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-based metric bound of 10ms which will apply to both connectivity-constructs.

- *Network Slice 4 on SDP6 and SDP7b, with P2P connectivity type. This is a L2 Slice Service and using 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.

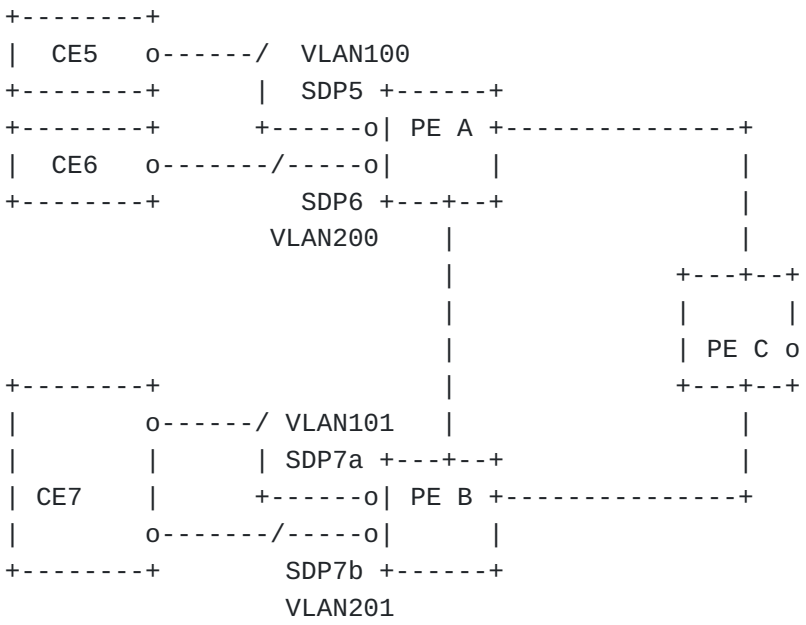


Figure 22: Example of Two P2P Slice Services

Figure 23 shows an example YANG JSON data for the body of the Network Slice Service instances request.

```
{
  "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-tag": [
                        {
                          "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-tag": [
                {
                  "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-tag": [
```

```
                {
                  "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-tag": [
                {
                  "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": {}
              }
            ]
          }
        ]
      }
    ]
  }
}
```

Figure 23: Example of a Message Body to Create Two P2P Slice Services

The example shown in Figure 24 illustrates how a customer subscribes to the monitoring information of "slice3". The customer is interested in the operational and performance status of SDPs and connectivity constructs.

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

POST /restconf/operations/ietf-subscribed-notifications:establish-\
                                     subscription
Host: example.com
Content-Type: application/yang-data+json
{
  "ietf-subscribed-notifications:input": {
    "stream-subtree-filter": {
      "ietf-network-slice-service:network-slice-services": {
        "slice-service": [
          {
            "id": "slice3",
            "sdps": {
              "sdp": [
                {
                  "id": "5",
                  "status": {
                    "oper-status": {
                      "status": {}
                    }
                  },
                  "sdp-monitoring": {
                    "incoming-bw-value": {},
                    "outgoing-bw-value": {}
                  }
                },
                {
                  "id": "7a",
                  "status": {
                    "oper-status": {
                      "status": {}
                    }
                  },
                  "sdp-monitoring": {
                    "incoming-bw-value": {},
                    "outgoing-bw-value": {}
                  }
                }
              ]
            },
            "connection-groups": {
              "connection-group": [
                {
                  "id": "matrix3",
                  "connectivity-type": "ietf-nss:point-to-point",
                  "connectivity-construct": [
                    {
                      "id": 1,
```

```
                    "p2p-sender-sdp": "5",
                    "p2p-receiver-sdp": "7a",
                    "status": {
                      "oper-status": {
                        "status": "{}"
                      }
                    },
                    "connectivity-construct-monitoring": {
                      "one-way-min-delay": {},
                      "one-way-max-delay": {}
                    }
                  },
                  {
                    "id": 2,
                    "p2p-sender-sdp": "7a",
                    "p2p-receiver-sdp": "5",
                    "status": {
                      "oper-status": {
                        "status": {}
                      }
                    },
                    "connectivity-construct-monitoring": {
                      "one-way-min-delay": {},
                      "one-way-max-delay": {}
                    }
                  }
                ]
              }
            ]
          }
        }
      ]
    },
    "ietf-yang-push:periodic": {
      "period": "500"
    }
  }
}
```

Figure 24: Example of a Message Body to Subscribe Monitoring Information of the Slice Service

The example Figure 25 shows a snapshot of YANG JSON data for the body of operational and performance status of the Network Slice Service "slice3".

```
{
  "slice-service": [
    {
      "id": "slice3",
      "description": "example slice3",
      "slo-sle-template": "low-latency-template",
      "status": {
        "oper-status": {
          "status": "ietf-vpn-common:op-up"
        }
      },
      "sdps": {
        "sdp": [
          {
            "id": "5",
            "node-id": "PE-A",
            "status": {
              "oper-status": {
                "status": "ietf-vpn-common:op-up"
              }
            },
            "sdp-monitoring": {
              "incoming-bw-value": "10000",
              "outgoing-bw-value": "10000"
            }
          },
          {
            "id": "7a",
            "node-id": "PE-B",
            "status": {
              "oper-status": {
                "status": "ietf-vpn-common:op-up"
              }
            },
            "sdp-monitoring": {
              "incoming-bw-value": "10000",
              "outgoing-bw-value": "10000"
            }
          }
        ]
      },
      "connection-groups": {
        "connection-group": [
          {
            "id": "matrix3",
            "connectivity-type": "ietf-nss:point-to-point",
            "connectivity-construct": [
              {
                "id": 1,
```

```
            "p2p-sender-sdp": "5",
            "p2p-receiver-sdp": "7a",
            "status": {
              "oper-status": {
                "status": "ietf-vpn-common:op-up"
              }
            },
            "connectivity-construct-monitoring": {
              "one-way-min-delay": "15",
              "one-way-max-delay": "20"
            }
          },
          {
            "id": 2,
            "p2p-sender-sdp": "7a",
            "p2p-receiver-sdp": "5",
            "status": {
              "oper-status": {
                "status": "ietf-vpn-common:op-up"
              }
            },
            "connectivity-construct-monitoring": {
              "one-way-min-delay": "15",
              "one-way-max-delay": "20"
            }
          }
        ]
      }
    ]
  }
},
{
  "id": "slice4",
  "description": "example slice4",
  "slo-sle-template": "high-BW-template",
  "status": {
    "oper-status": {
      "status": "ietf-vpn-common:op-up"
    }
  },
  "sdps": {
    "sdp": [
      {
        "id": "6",
        "node-id": "PE-A",
        "status": {
          "oper-status": {
            "status": "ietf-vpn-common:op-up"
          }
```

```
      },
      "sdp-monitoring": {
        "incoming-bw-value": "10000000",
        "outgoing-bw-value": "10000000"
      }
    },
    {
      "id": "7b",
      "node-id": "PE-B",
      "status": {
        "oper-status": {
          "status": "ietf-vpn-common:op-up"
        }
      },
      "sdp-monitoring": {
        "incoming-bw-value": "10000000",
        "outgoing-bw-value": "10000000"
      }
    }
  ]
},
"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",
          "status": {
            "oper-status": {
              "status": "ietf-vpn-common:op-up"
            }
          },
          "connectivity-construct-monitoring": {
            "one-way-min-delay": "150",
            "one-way-max-delay": "200"
          }
        },
        {
          "id": 2,
          "p2p-sender-sdp": "7b",
          "p2p-receiver-sdp": "6",
          "status": {
            "oper-status": {
              "status": "ietf-vpn-common:op-up"
            }
```

```
                },
                "connectivity-construct-monitoring": {
                  "one-way-min-delay": "150",
                  "one-way-max-delay": "200"
                }
              }
            ]
          }
        ]
      }
    }
  ]
}
```

Figure 25: Example of a Message Body of a Snapshot of Monitoring of the Slice Service

B.3. Example-3: A Hub and Spoke Slice Service with a P2MP Connectivity Construct

Figure 26 shows an example of one Network Slice Service instance where the SDPs are the customer-facing ports on the PE:

> 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.

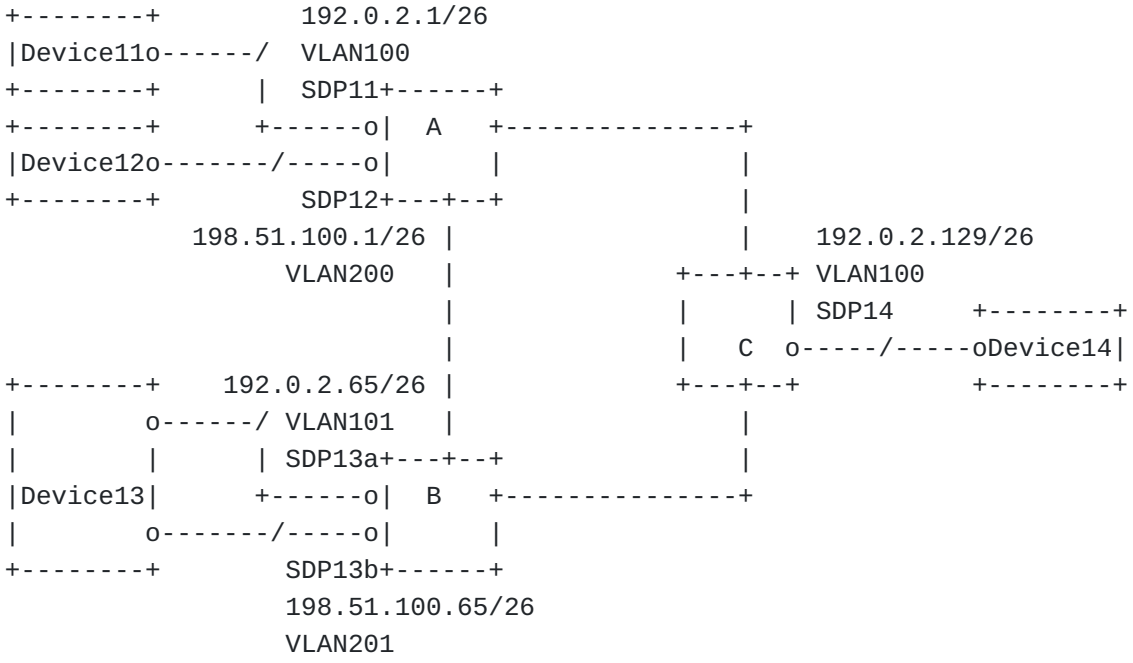


Figure 26: Example of A Hub and Spoke Slice Service

Figure 27 shows an example YANG JSON data for the body of the hub-spoke Network Slice Service instances request.

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

{
  "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-tag": [
                        {
                          "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-tag": [
```

```
                        {
                          "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-tag": [
                        {
                          "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-tag": [
                        {
                          "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-tag": [
                        {
                          "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": {}
              }
            ]
          }
        ]
      }
    }
  ]
}
```

Figure 27: Example of a Message Body to Create A Hub and Spoke Slice Service

B.4. Example-4: An A2A Slice Service with Multiple SLOs and DSCP Matching

Figure 28 shows an example of a Network slice instance where the SDPs are the customer-facing ports on the PE:

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.

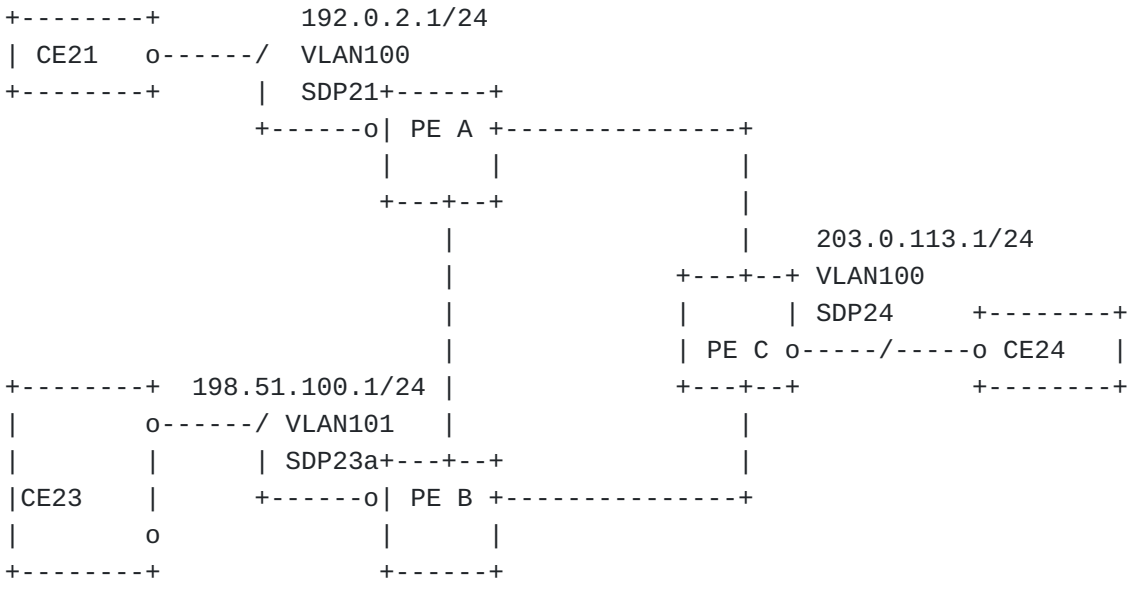


Figure 28: Example of An A2A Slice Service with DSCP Matching

Figure 29 shows an example YANG JSON data for the body of the Network Slice Service instances request.

```
{
  "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-tag": [
              {
                "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-tag": [
            {
              "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-tag": [
                {
                  "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": {}
              }
            ]
          }
        ]
      }
    ]
  }
}
```

Figure 29: Example of a Message Body to Create An A2A Slice Service with DSCP Matching

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

Figure 30 shows an example of a 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 construct level takes precedence over the one specified at the group level.

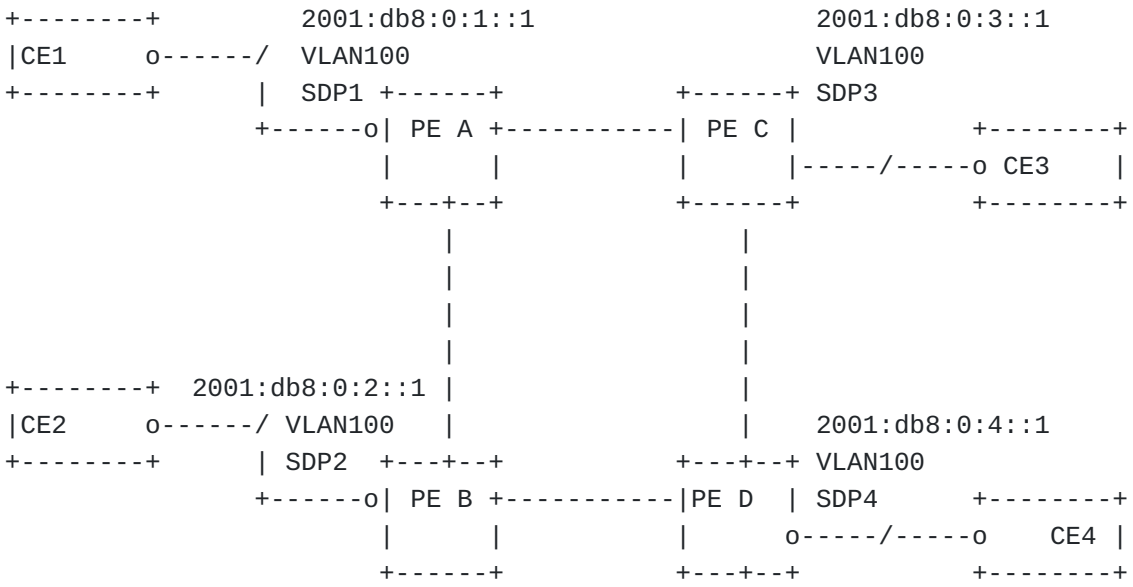


Figure 30: Example of An A2A Slice Service with SLO Precedence

Figure 31 shows an example YANG JSON data for the body of the Network Slice Service instances request.

```
{
  "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-tag": [
              {
                "tag-type": "ietf-nss:vlan-id",
                "value": [
                  "100"
                ]
              }
            ]
          },
          "incoming-qos-policy": {
            "qos-policy-name": "QoS-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-tag": [
              {
                "tag-type": "ietf-nss:vlan-id",
                "value": [
                  "100"
                ]
              }
            ]
          },
          "incoming-qos-policy": {
            "qos-policy-name": "QoS-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-tag": [
                {
                  "tag-type": "ietf-nss:vlan-id",
                  "value": [
                    "100"
                  ]
                }
              ]
            },
            "incoming-qos-policy": {
              "qos-policy-name": "QoS-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-tag": [
            {
              "tag-type": "ietf-nss:vlan-id",
              "value": [
                "100"
              ]
            }
          ]
        },
        "incoming-qos-policy": {
          "qos-policy-name": "QoS-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": {}
              }
            ]
          }
        ]
      }
    }
  ]
}
```

Figure 31: Example of a Message Body to Create an A2A Slice Service with SLO Precedence

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

Figure 32 shows an example of one Network slice instance where the SDPs are located at the PE-facing ports on the CE:

* 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

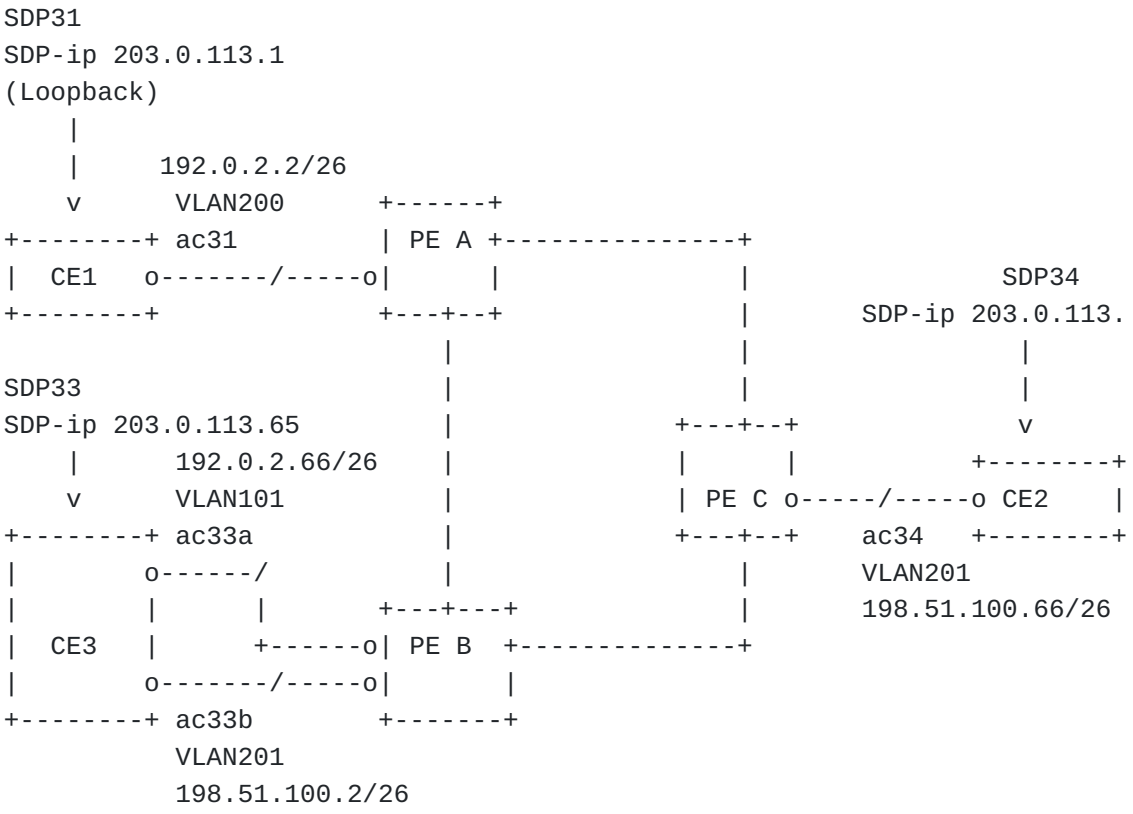


Figure 32: Example of an A2A Slice Service with CE Based SDP

Figure 33 shows an example YANG JSON data for the body of the Network Slice Service instances request.

```
{
  "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-tag": [
              {
                "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-tag": [
            {
              "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-tag": [
            {
              "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-tag": [
                {
                  "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": {}
```

```
                }
              ]
            }
          ]
        }
      }
    ]
  }
}
```

Figure 33: Example of a Message Body to Create an CE based A2A Slice Services

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

Figure 34 shows an example of one Network slice instance 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.

*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

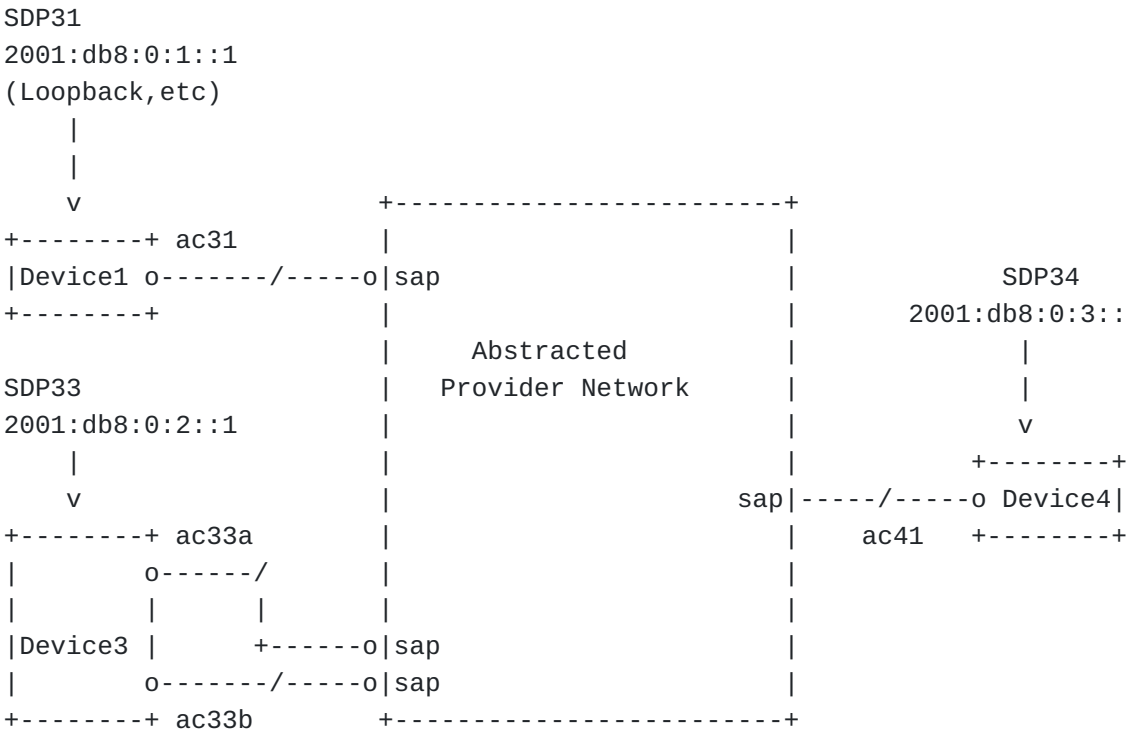


Figure 34: Example of a Message Body to Create an A2A CE Based Slice Service with Abstraction

Figure 35 shows an example YANG JSON data for the body of the Network Slice Service instances request.

```
{
  "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": {}
        }
```

```
                ]
              }
            ]
          }
        }
      ]
    }
}
```

Figure 35: Example of a Message Body to Create an A2A Slice Service with Abstraction

Appendix C. Complete Model Tree Structure

```
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?   slice-template-ref
     |     +--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?            uint32
     |     +--rw sle-policy
     |        +--rw security*              identityref
     |        +--rw isolation*             identityref
     |        +--rw max-occupancy-level?   uint8
     |        +--rw path-constraints
     |           +--rw service-functions
     |           +--rw diversity
     |              +--rw diversity-type?
     |                      te-types:te-path-disjointness
     +--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?         slice-template-ref
        |  +--:(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?            uint32
        |        +--rw sle-policy
        |           +--rw security*              identityref
```

```
|           +--rw isolation*             identityref
|           +--rw max-occupancy-level?   uint8
|           +--rw path-constraints
|              +--rw service-functions
|              +--rw diversity
|                 +--rw diversity-type?
|                         te-types:te-path-disjointness
+--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 geo-location
|     |  +--rw reference-frame
|     |  |  +--rw alternate-system?    string
|     |  |  |       {alternate-systems}?
|     |  |  +--rw astronomical-body?   string
|     |  |  +--rw geodetic-system
|     |  |     +--rw geodetic-datum?    string
|     |  |     +--rw coord-accuracy?    decimal64
|     |  |     +--rw height-accuracy?   decimal64
|     |  +--rw (location)?
|     |  |  +--:(ellipsoid)
|     |  |  |  +--rw latitude?    decimal64
|     |  |  |  +--rw longitude?   decimal64
|     |  |  |  +--rw height?      decimal64
|     |  |  +--:(cartesian)
|     |  |     +--rw x?           decimal64
|     |  |     +--rw y?           decimal64
|     |  |     +--rw z?           decimal64
|     |  +--rw velocity
|     |  |  +--rw v-north?   decimal64
|     |  |  +--rw v-east?    decimal64
|     |  |  +--rw v-up?      decimal64
|     |  +--rw timestamp?         yang:date-and-time
|     |  +--rw valid-until?       yang:date-and-time
|     +--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 classes
|     |        +--rw cos* [cos-id]
|     |           +--rw cos-id    uint8
|     |           +--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 classes
|     |        +--rw cos* [cos-id]
|     |           +--rw cos-id    uint8
|     |           +--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 ce-mode?                  boolean
|     +--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?                     uint32
|     |     +--rw ac-tags
|     |     |  +--rw ac-tag* [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 classes
|     |     |        +--rw cos* [cos-id]
|     |     |           +--rw cos-id    uint8
|     |     |           +--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 classes
|     |     |        +--rw cos* [cos-id]
|     |     |           +--rw cos-id    uint8
```

```
|     |     |          +--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?     yang:gauge64
|        +--ro incoming-bw-percent?   percentage
|        +--ro outgoing-bw-value?     yang:gauge64
|        +--ro outgoing-bw-percent?   percentage
+--rw connection-groups
|  +--rw connection-group* [id]
|     +--rw id                                string
|     +--rw connectivity-type?
|     |       identityref
|     +--rw (slo-sle-policy)?
|     |  +--:(standard)
|     |  |  +--rw slo-sle-template?
|     |  |          slice-template-ref
|     |  +--:(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?            uint32
|     |       +--rw sle-policy
|     |          +--rw security*
|     |          |       identityref
|     |          +--rw isolation*
|     |          |       identityref
|     |          +--rw max-occupancy-level?   uint8
|     |          +--rw path-constraints
|     |             +--rw service-functions
|     |             +--rw diversity
|     |                +--rw diversity-type?
|     |                        te-types:te-path-disjointness
|     +--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?
|     |  |           |          slice-template-ref
|     |  |           +--:(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?
|     |  |                |          uint32
|     |  |                +--rw sle-policy
|     |  |                   +--rw security*
|     |  |                   |       identityref
|     |  |                   +--rw isolation*
|     |  |                   |       identityref
|     |  |                   +--rw max-occupancy-level?
|     |  |                   |       uint8
|     |  |                   +--rw path-constraints
|     |  |                      +--rw service-functions
|     |  |                      +--rw diversity
|     |  |                         +--rw diversity-type?
|     |  |                                 te-types:te-path-dis
|     |  +--rw (slo-sle-policy)?
|     |  |  +--:(standard)
|     |  |  |  +--rw slo-sle-template?
|     |  |  |          slice-template-ref
|     |  |  +--:(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?            uint32
|     |  |        +--rw sle-policy
|     |  |           +--rw security*
|     |  |           |       identityref
|     |  |           +--rw isolation*
|     |  |           |       identityref
|     |  |           +--rw max-occupancy-level?   uint8
|     |  |           +--rw path-constraints
|     |  |              +--rw service-functions
|     |  |              +--rw diversity
|     |  |                 +--rw diversity-type?
|     |  |                         te-types:te-path-disjointness
```

```
|     |  +--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?         yang:gauge64
|     |     +--ro one-way-max-delay?         yang:gauge64
|     |     +--ro one-way-delay-variation?   yang:gauge64
|     |     +--ro one-way-packet-loss?       percentage
|     |     +--ro two-way-min-delay?         yang:gauge64
|     |     +--ro two-way-max-delay?         yang:gauge64
|     |     +--ro two-way-delay-variation?   yang:gauge64
|     |     +--ro two-way-packet-loss?       percentage
|     +--ro connection-group-monitoring
|        +--ro one-way-min-delay?         yang:gauge64
|        +--ro one-way-max-delay?         yang:gauge64
|        +--ro one-way-delay-variation?   yang:gauge64
|        +--ro one-way-packet-loss?       percentage
|        +--ro two-way-min-delay?         yang:gauge64
|        +--ro two-way-max-delay?         yang:gauge64
|        +--ro two-way-delay-variation?   yang:gauge64
|        +--ro two-way-packet-loss?       percentage
+--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 Network Slice Service requirements is that the Network Slice Service interface is a technology-agnostic interface, whereas the VN model is bound to the TE Topologies. The realization of the Network Slice does not necessarily require the slice network to support the TE technology.

The ACTN VN (Virtual Network) model introduced in[I-D.ietf-teas-actn-vn-yang] is the abstract customer view of the TE network. Its YANG structure includes four components:

*VN: A Virtual Network (VN) is a network provided by a service provider to a customer for use and two types of VN has defined. The Type 1 VN can be seen as a set of edge-to-edge abstract links. Each link is an abstraction of the underlying network which can encompass edge points of the customer's network, access links, intra-domain paths, and inter-domain links.

*AP: An AP is a logical identifier used to identify the access link which is shared between the customer and the IETF scoped Network.

*VN-AP: A VN-AP is a logical binding between an AP and a given VN.

*VN-member: A VN-member is an abstract edge-to-edge link between any two APs or VN-APs. Each link is formed as an E2E tunnel across the underlying networks.

The Type 1 VN can be used to describe 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 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 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