TEAS
Internet-Draft
Intended status: Standards Track
Expires: 27 April 2023

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

24 October 2022

A YANG Data Model for the IETF Network Slice Service ~~YANG Model~~

draft-ietf-teas-ietf-network-slice-nbi-yang-03

Abstract

This document defines a YANG data model for the IETF Network Slice ~~service~~Service. The model can be used ~~by an~~in the ~~–~~IETF Network Slice Service Interface ~~customer to manage IETF Network Slices~~between a customer and a provider that ~~.~~offers IETF Network Slices.

Status of This Memo

This Internet-Draft will expire on 27 April 2023.

Copyright Notice

Table of Contents

1. Introduction 3
2. Conventions used in this document 3
2.1. Acronyms 4
3. IETF Network Slice Service Overview 5
4. IETF Network Slice Service Model Usage 8
5. IETF Network Slice Service Modeling Description 9
5.1. IETF Network Slice Service SLO and SLE Templates 15
5.2. IETF Network Slice Service 16
5.2.1. IETF Network Slice Service Demarcation Point 17
5.2.2. IETF Network Slice Service Connectivity Construct . . 22
5.2.3. IETF Network Slice Service SLO and SLE Policy 23
5.2.4. IETF Network Slice Service Monitoring 25
5.2.5. IETF Network Slice Service on Abstract TE topology . 26
6. IETF Network Slice Service Module 27
7. Security Considerations 54
8. IANA Considerations 55
9. Acknowledgments 55
10. Contributors 55
11. References 56
11.1. Normative References 56
11.2. Informative References 57
Appendix A. Examples of Network Slice Services 58
A.1. Example-1: Two Any-to-any Slice Services with different match approachs 58
A.2. Example-2: Two P2P slice services with different match approaches 68
A.3. Example-3: A Hub-and-spoke Slice Service with a P2MP connectivity construct 74
A.4. Example-4: An Any-to-any Slice service with multiple SLOs and DSCP Matching 80
A.5. Example-5: An any-to-any Network Slice Service with SLO Precedence Policies 85
A.6. Example-6: SDP at CE, L3 any-to-any Slice Service 94
A.7. Example-7: SDP at CE, L3 any-to-any Slice Service with Network abstraction 102
Appendix B. Comparison with Other Possible Design choices for IETF Network Slice Service Interface 106
B.1. ACTN VN Model Augmentation 107
B.2. RFC8345 Augmentation Model 108
Authors' Addresses 108

1. Introduction

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

~~The YANG model discussed in this document is defined based on the description of the IETF Network Slice service in [I-D.ietf-teas-ietf-network-slices],~~This YANG module can be used in the IETF Network Slice Service Interface exposed by a provider to its customers in order to manage (e.g., subscribe, delete, or change) IETF Network Slice Services. The agreed service will then trigger appropriate IETF Network Slice operation, such as instantiating, modifying, or deleting an IETF Network Slice. ~~which is used to operate IETF~~

~~Network Slices during the IETF Network Slice instantiation~~. As discussed in [I-D.ietf-teas-ietf-network-slices], the mapping between an IETF Network Slice Service and its realization is implementation and deployment specific. ~~This YANG model supports various operations on IETF Network Slices such as creation, modification, deletion, and monitoring.~~

T~~The~~ he IETF Network Slice Service (NSSM) YANG ~~model discussed in this document~~Model ~~describes~~ focuses on the requirements of an IETF Network Slice ~~service~~ Service from the point of view of the customer, not how it is implemented by a provider. ~~It~~ The module is ~~thus~~ classified as customer service model (Section 2 of~~ in~~ [RFC8309]).

The NSSM conforms to the Network Management Datastore Architecture ~~The IETF Network Slice operational state is included in the same tree as the configuration consistent with~~ Network Management Datastore Architecture (NMDA) defined in [RFC8342].

Editorial Note: (To be removed by RFC Editor)

This draft contains several placeholder values that need to be replaced with finalized values at the time of publication. Please apply the following replacements:

* "XXXX" --> the assigned RFC value for this draft both in this draft and in the YANG models under the revision statement.

* The "revision" date in model, in the format XXXX-XX-XX, needs to be updated with the date the draft gets approved.

2. Conventions used in this document

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP14, [RFC2119], [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following terms are defined in [RFC6241] and are used in this specification:

* client

* configuration data
* state data

This document makes use of the terms defined in [RFC7950].

The tree ~~diagram~~ diagrams used in this document follow the notation defined in [RFC8340].

This document also makes use of the terms ~~introduced~~ defined in ~~the Framework for IETF Network Slices~~ [I-D.ietf-teas-ietf-network-slices]:

* Attachment Circuit (AC): ~~as defined in section~~ See Section 3.1

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

- Service Demarcation Point (SDP): ~~The point at which an IETF Network Slice service is delivered by a service provider to a customer, as defined in~~See Sections 2.1 and 4.2 [I-D.ietf-teas-ietf-network-slices] ~~section 4.2.~~

Commenté [BMI1]: To align with Section 2.1 of the framework

Commenté [BMI2]: No need to redefine it here. A pointer is sufficient.

Commenté [BMI3]: As core definitions are provided there.

- Connectivity Construct: ~~A set of SDPs together with a communication type that defines how traffic flows between the SDPs, as defined in~~See ~~section~~ Sections 2.1 and 3.2 [I-D.ietf-teas-ietf-network-slices].

In addition, ~~This~~ this document defines the following term~~s~~:

- Connection Group: ~~Connection group is a~~An arbitrary collection of one or more connectivity constructs, which can be used for the following possible purposes:

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

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

  Share specific SLO limits within multiple connectivity constructs.

2.1. Acronyms

The following acronyms are used in the document:

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

3. IETF Network Slice Service Overview

As defined in Section 3.2 of [I-D.ietf-teas-ietf-network-slices], an IETF Network Slice S~~s~~ervice is specified in terms of a set of SDPs, a set of one or more connectivity constructs between subsets of these SDPs, and a set of SLOs and SLEs ~~(see Section 4)~~ for each SDP sending to each connectivity construct. A communication type (point- to-point (P2P),

Commenté [BMI4]: A copy/paste bug from the framework.

point-to-multipoint (P2MP), or any-to-any (A2A)) is specified for each connectivity construct.

An IETF Network ~~slice~~ Slice S~~s~~ervice may include ~~only~~ one connectivity construct or multiple connectivity constructs that associate sets of SDPs. The SDPs serve as the IETF Network Slice ingress/egress points. An SDP is identified by a unique identifier in the context of an IETF Network Slice ~~service~~Service.

Commenté [BMI5]: I woudl delete this as this is redundant with “one or More ..” in the previous para.

~~An example~~Examples ~~-~~of IETF ~~network~~ Network ~~slice~~ Slice ~~services~~ Services that ~~containing~~ contain each only one connectivity construct ~~is~~ are shown in Figure 1.

```
          +----------------------------------------------+
          |                                              |
          |                                              |
          |       Slice Service 1 with 1 P2P CC          |
    SDP1  O------------------->--------------------------O  SDP2
          |                                              |
          |                                              |
          |       Slice Service 2 with 1 P2MP CC
          |                  +---------------------------O  SDP4
    SDP3  O----------->------+                           |
          |                  +---------------------------O  SDP5
          |                                              |
          |                                              |
          |       Slice Service 3 with 1 A2A  CC
    SDP6  O-----------<>-----+---------<>----------------O  SDP8
          |                  |                           |
    SDP7  O-----------<>-----+---------<>----------------O  SDP9
          |                                              |
          |                                              |
          +----------------------------------------------+
          |<----------IETF Network Slice Services------->|
~~        |          between endpoints SDP1 to SDP9      |~~

~~  SDP:  Service Demarcation Point~~
~~  P2P:  Point-to-point~~
~~  P2MP: Point-to-multipoint~~
~~  A2A:  Any-to-any~~
    CC:   Connectivity Construct
    O   : Represents Service Demarcation Point
   ----: Represents Connectivity Construct
   < > : Inbound/Outbound ~~Direction~~directions
```

Commenté [BMI6]: You may indicate in the text that the service, as defined, is not a bidirectional unicast service.

Figure 1: ~~An~~ Examples of IETF Network Slice Services with Single Connectivity Construct~~s~~

~~An example of~~ Examples of IETF network S~~s~~lice ~~services~~ Services that ~~containing~~ each contain multiple connectivity constructs ~~is~~ are shown in Figure 2.

```
          +----------------------------------------------+
          |                                              |
          |       Slice Service 4 with 2 P2P CCs         |
    SDP10 O------------------->--------------------------O  SDP12
    SDP11 O------------------->--------------------------O  SDP13
```

```
        |                                              |
        |                                              |
        |      Slice Service 5 with 2 P2P CCs          |
        | +------------------>-----------------------+  |
  SDP14 o/                                           \ o SDP15
        |\                                           / |
        | +-----------------<------------------------+  |
        |                                              |
  Application of                                       |
  match criteria                                       |
        |                                              |
        |      Slice Service 6 with 2 P2P CCs          |
        v +--x-x-x-x-x-x---->---x-x-x-x-x-x-x-x-x---+  |
  SDP16 o/                                           \ o SDP17
        |\                                           / |
        | +--%-%-%-%-%-%---->---%-%-%-%-%-%-%-%-%---+  |
        |                                              |
        +----------------------------------------------+
        |<----------IETF Network Slice Services------->|
~~        |         between endpoints SDP10 to SDP17       |~~

  Slice Service: IETF Network Slice Service
~~  SDP:  Service Demarcation Point~~
~~  P2P:  Point-to-point~~
  CC:   Connectivity Construct
  o   : Represents Service Demarcation Point
 ----: Represents Connectivity Construct
 -x-x-x-: Represents connectivity construct applying match-criteria 1
 -%-%-%-: Represents connectivity construct applying match-criteria 2
 <–> : Direction
```

Figure 2: ~~An~~ Examples of IETF Network Slice Services, each with multiple Connectivity Constructs

As shown in the example, ~~The~~ the IETF ~~network~~ Network ~~slice~~ Slice ~~service~~ Service 4 contains two P2P connectivity constructs between ~~various~~different sets of SDPs. The IETF ~~network~~ Network ~~slice~~ Slice ~~service~~ Service 5 is a bidirectional ~~P2P~~ unicast service between ~~on~~ SDP14 and SDP15 that is decomposed into~~. The service consists of~~ two unidirectional P2P connectivity Constructs between these two SDPs. For the traffic ~~from~~ received via each SDPs, there is only one connectivity construct. ~~For the~~ The IETF N~~n~~etwork ~~slice~~ Slice S~~s~~ervice 6~~, an P2P service on SDP 16 and SDP 17, this service~~ includes two unidirectional P2P connectivity constructs with different set of SLOs between ~~the~~ these two SDPs. Since SDP16 is the source of two connectivity constructs, this requires non-overlapping matching criteria be applied to traffic ~~of~~on SDP16 to ~~distinguish~~ demultiplex flows and inject them on the appropriate ~~between flows constructed by two~~ connectivity construct~~connections~~.

~~One~~ An example of the demultiplexing criteria at ~~example could be that~~ the SDP16 is to use~~s~~ the ~~-~~DSCP/Traffic Class (TC) marking ~~s~~ ~~in the packets~~ to

Commenté [BMI7]: This is not defined yet at this stage

Commenté [BMI8]: Not sure why match criteria are specifically called out here. Such criteria will apply for all services. I would simplify the figure

Commenté [BMI9]: Which one ?

Commenté [BMI10]: Not sure to get the intent here.

Commenté [BMI11]: The traffic is NOT necessarily originated by SDP differentiate ~~traffic~~ flows and, then, ~~-~~steer~~s~~ the traffic among to different connectivity constructs ~~to ensure different SLOs.~~.

4. IETF Network Slice Service Model Usage

~~The intention of t~~The ~~IETF Network Slice service model~~NSSM can be used by a provider to expose its Slice Service, and by a customer ~~is to allow the customer~~ to manage its IETF Network Slices Services (e.g., request, delete, or change). ~~In particular, the model allows customers to operate in an abstract and technology-agnostic manner, with~~ The details about how service requests are handled by the provider, including which network operations are triggered, are internal the provided. The details of the IETF Network Slices realization are hidden customer.

~~According to the [I-D.ietf-teas-ietf-network-slices] description,~~The IETF Network Slices are applicable to use cases, such as (but not limited to) network wholesale services, network infrastructure sharing among operators, NFV (Network Function Virtualization) connectivity, Data Center Interconnect, and 5G E2E network slice.

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

Commenté [BMI12]: You may move this text to be positioned right after the text introducing the NSC.

The ~~IETF Network Slice Controller (NSC)~~ is an~~ logical~~ entity ~~that Allows~~that exposes the IETF Network Slice Service Interface to customers to manage IETF ~~network~~ Network ~~slices~~Slice Services. Typically, ~~The~~ the NSC receives requests from its customer-facing interface (e.g., from a management system). This interface carries data objects the IETF Network Slice service customer provides, describing the needed IETF network slices service in terms of a set of SDPs, the associated connectivity constructs and the service objectives that the customer wishes to be fulfilled. These service requirements are then translated into technology-specific actions that are implemented in the underlying network using a network-facing interface. The details of how the IETF Network Slices are put into effect are out of scope for this document.

Commenté [BMI13]: You may clarify that this is for the service creation.

The details may be different for deleting a slice or for GET-like operations.

```
             +----------------------------------------+
             |    IETF Network Slice Customer         |
             |                                        |
             +----------------+-----------------------+
                              |
                              |
                              |IETF Network Slice ~~service~~ Service M~~m~~odel
(NSSM)
                              |
        +---------------------+--------------------------+
        |      IETF Network Slice Controller (NSC)       |
        +------------------------------------------------+
```

Figure 3: IETF Network Slice Service Reference Architecture

5. IETF Network Slice Service Modeling Description

The 'ietf-network-slice-service' module ~~uses~~ includes two main data nodes: ~~list~~ 'slice-service' and ~~container~~ 'slo-sle-templates' (see Figure 4).

The 'slice-service' list includes the set of IETF Network Slice ~~services~~ Services that are ~~managed~~ maintained by ~~within~~ a provider ~~network~~. 'slice-service' is the data structure that abstracts ~~an~~ the IETF Network Slice ~~service~~Service. Under the ~~"~~'slice-service~~",~~ ', ~~list~~ the ~~"~~'sdp~~"~~ ' list is used to abstract the SDPs. ~~And~~ ~~list~~The ~~"~~'connection-group~~"~~ ' is used to abstract connectivity constructs between SDPs.

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

The figure below describes the overall structure of the YANG module:

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

Commenté [BMI14]: Add a note that the model can be recursive (hierarchical mode). An NSSM may map an child NSSMs.

Commenté [BMI15]: To align with the framework, but also to cover brokering/hierarchical cases/

Commenté [BMI16]: As many may be deployed

Commenté [BMI17]: Consider adding a leaf for « customer-name »

```
|  |  +--rw value*      string
|  +--rw tag-opaque* [tag-name]
|     +--rw tag-name    string
|     +--rw value*      string
+--rw (slo-sle-policy)?
|  +--:(standard)
|  |  +--rw slo-sle-template?         leafref
|  +--:(custom)
|     +--rw service-slo-sle-policy
|        +--rw policy-description?     string
|        +--rw metric-bounds
|        |  +--rw metric-bound* [metric-type]
|        |     +--rw metric-type          identityref
|        |     +--rw metric-unit          string
|        |     +--rw value-description?   string
|        |     +--rw bound?               uint64
|        +--rw security*               identityref
|        +--rw isolation?              identityref
|        +--rw max-occupancy-level?    uint8
|        +--rw mtu?                    uint16
|        +--rw steering-constraints
|           +--rw path-constraints
|           +--rw service-function
+--rw status
|  +--rw admin-status
|  |  +--rw status?         identityref
|  |  +--rw last-updated?   yang:date-and-time
|  +--ro oper-status
|     +--ro status?         identityref
|     +--ro last-updated?   yang:date-and-time
+--rw sdps
|  +--rw sdp* [sdp-id]
|     +--rw sdp-id                    string
|     +--rw sdp-description?          string
|     +--rw location
|     |  +--rw altitude?    int64
|     |  +--rw latitude?    decimal64
|     |  +--rw longitude?   decimal64
|     +--rw node-id?                  string
|     +--rw sdp-ip?                   inet:ip-address
|     +--rw peer-sap-id?              string
|     +--rw ltp?                      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
```

Commenté [BMI18]: What about having an explicit "requested-service-start" to schedule service placement.

Commenté [BMI19]: It would be good to have some discussion how/whether this can be used to expose the actual-slice-service start/stops.

Commenté [BMI20]: Should the node-id be added as a second key?

Commenté [BMI21]: What about multiple addresses, each per address-family?

Commenté [BMI22]: The framework says « Excess traffic is dropped by default, unless specific out-of-profile policies are agreed between the customer and the provider."

It seems this is not covered in the NSSM.

```
         |     |     +--rw cbs?   uint64
         |     |     +--rw eir?   uint64
         |     |     +--rw ebs?   uint64
         |     |     +--rw pir?   uint64
         |     |     +--rw pbs?   uint64
         |     +--rw outgoing-qos-policy
         |     |  +--rw qos-policy-name?   string
         |     |  +--rw rate-limits
         |     |     +--rw cir?   uint64
         |     |     +--rw cbs?   uint64
         |     |     +--rw eir?   uint64
         |     |     +--rw ebs?   uint64
         |     |     +--rw pir?   uint64
         |     |     +--rw pbs?   uint64
         |     +--rw sdp-peering
         |     |  +--rw protocol* [protocol-type]
         |     |  |  +--rw protocol-type    identityref
         |     |  |  +--rw attribute* [attribute-type]
         |     |  |     +--rw attribute-type    identityref
         |     |  |     +--rw value*            string
         |     |  +--rw opaque* [attribute-name]
         |     |     +--rw attribute-name    string
         |     |     +--rw value*            string
         |     +--rw attachment-circuits
         |     |  +--rw attachment-circuit* [ac-id]
         |     |     +--rw ac-id                  string
         |     |     +--rw ac-description?        string
         |     |     +--rw ac-node-id?            string
         |     |     +--rw ac-tp-id?              string
         |     |     +--rw ac-ip-address?         inet:ip-address
         |     |     +--rw ac-ip-prefix-length?   uint8
         |     |     +--rw peer-sap-id?           string
         |     |     +--rw mtu?                   uint16
         |     |     +--rw ac-tags
         |     |     |  +--rw ac-tags* [ac-tag-type]
         |     |     |  |  +--rw ac-tag-type    identityref
         |     |     |  |  +--rw value*         string
         |     |     |  +--rw ac-tag-opaque* [tag-name]
         |     |     |     +--rw tag-name    string
         |     |     |     +--rw value*      string
         |     |     +--rw incoming-qos-policy
         |     |     |  +--rw qos-policy-name?   string
         |     |     |  +--rw rate-limits
         |     |     |     +--rw cir?   uint64
         |     |     |     +--rw cbs?   uint64
         |     |     |     +--rw eir?   uint64
         |     |     |     +--rw ebs?   uint64
         |     |     |     +--rw pir?   uint64
         |     |     |     +--rw pbs?   uint64
         |     |     +--rw outgoing-qos-policy
         |     |     |  +--rw qos-policy-name?   string
         |     |     |  +--rw rate-limits
         |     |     |     +--rw cir?   uint64
         |     |     |     +--rw cbs?   uint64
         |     |     |     +--rw eir?   uint64
         |     |     |     +--rw ebs?   uint64
         |     |     |     +--rw pir?   uint64
         |     |     |     +--rw pbs?   uint64
```

Commenté [BMI23]: Shouldn't multiple CoSes be allowed for the same slice service?

If so, I would expect a structure similar to the in Figure 19 of RFC9291 be used here.

Commenté [BMI24]: Idem as for the inbound

Commenté [BMI25]: Values can be an ip address + a prefix length for a given attribute.

The order and semantic is important here. I'm afraid the current approach may lead to issues.

Commenté [BMI26]: I would reused L2/L3 connection from LxNMs

Commenté [BMI27]: This may not be available to the CE/customer

Commenté [BMI28]: Why about dual-stack?

Commenté [BMI29]: See the comment about CoSes

```
|     |     +--rw sdp-peering
|     |        +--rw protocol* [protocol-type]
|     |        |  +--rw protocol-type    identityref
|     |        |  +--rw attribute* [attribute-type]
|     |        |     +--rw attribute-type    identityref
|     |        |     +--rw value*            string
|     |        +--rw opaque* [attribute-name]
|     |           +--rw attribute-name    string
|     |           +--rw value*            string
|     +--rw status
|     |  +--rw admin-status
|     |  |  +--rw status?         identityref
|     |  |  +--rw last-updated?   yang:date-and-time
|     |  +--ro oper-status
|     |     +--ro status?         identityref
|     |     +--ro last-updated?   yang:date-and-time
|     +--ro sdp-monitoring
|        +--ro incoming-utilized-bandwidth?
|        |       te-types:te-bandwidth
|        +--ro incoming-bw-utilization        decimal64
|        +--ro outgoing-utilized-bandwidth?
|        |       te-types:te-bandwidth
|        +--ro outgoing-bw-utilization        decimal64
+--rw connection-groups
|  +--rw connection-group* [connection-group-id]
|     +--rw connection-group-id                string
|     +--rw connectivity-type?
|     |       identityref
|     +--rw (slo-sle-policy)?
|     |  +--:(standard)
|     |  |  +--rw slo-sle-template?            leafref
|     |  +--:(custom)
|     |     +--rw service-slo-sle-policy
|     |        +--rw policy-description?     string
|     |        +--rw metric-bounds
|     |        |  +--rw metric-bound* [metric-type]
|     |        |     +--rw metric-type
|     |        |     |       identityref
|     |        |     +--rw metric-unit          string
|     |        |     +--rw value-description?   string
|     |        |     +--rw bound?               uint64
|     |        +--rw security*               identityref
|     |        +--rw isolation?              identityref
|     |        +--rw max-occupancy-level?    uint8
|     |        +--rw mtu?                    uint16
|     |        +--rw steering-constraints
|     |           +--rw path-constraints
|     |           +--rw service-function
|     +--rw service-slo-sle-policy-override?
|     |       identityref
|     +--rw connectivity-construct* [cc-id]
|     |  +--rw cc-id
|     |  |       uint32
|     |  +--rw (connectivity-construct-type)?
|     |  |  +--:(p2p)
|     |  |  |  +--rw p2p-sender-sdp?
|     |  |  |  |       -> ../../../../sdps/sdp/sdp-id
|     |  |  |  +--rw p2p-receiver-sdp?
```

Commenté [BMI30]: The name may be confused with the data node right after

Commenté [BMI31]: This should be a gauge

Commenté [BMI32]: Idem

```
         |     |  |  |          -> ../../../../sdps/sdp/sdp-id
         |     |  |  +--:(p2mp)
         |     |  |  |  +--rw p2mp-sender-sdp?
         |     |  |  |  |       -> ../../../../sdps/sdp/sdp-id
         |     |  |  |  +--rw p2mp-receiver-sdp*
         |     |  |  |          -> ../../../../sdps/sdp/sdp-id
         |     |  |  +--:(a2a)
         |     |  |     +--rw a2a-sdp* [sdp-id]
         |     |  |        +--rw sdp-id                          leafref
         |     |  |        +--rw (slo-sle-policy)?
         |     |  |           +--:(standard)
         |     |  |           |  +--rw slo-sle-template?         leafref
         |     |  |           +--:(custom)
         |     |  |              +--rw service-slo-sle-policy
         |     |  |                 +--rw policy-description?
         |     |  |                 |       string
         |     |  |                 +--rw metric-bounds
         |     |  |                 |  +--rw metric-bound*
         |     |  |                 |          [metric-type]
         |     |  |                 |     +--rw metric-type
         |     |  |                 |     |       identityref
         |     |  |                 |     +--rw metric-unit
         |     |  |                 |     |       string
         |     |  |                 |     +--rw value-description?
         |     |  |                 |     |       string
         |     |  |                 |     +--rw bound?
         |     |  |                 |             uint64
         |     |  |                 +--rw security*
         |     |  |                 |       identityref
         |     |  |                 +--rw isolation?
         |     |  |                 |       identityref
         |     |  |                 +--rw max-occupancy-level?
         |     |  |                 |       uint8
         |     |  |                 +--rw mtu?
         |     |  |                 |       uint16
         |     |  |                 +--rw steering-constraints
         |     |  |                    +--rw path-constraints
         |     |  |                    +--rw service-function
         |     |  +--rw (slo-sle-policy)?
         |     |  |  +--:(standard)
         |     |  |  |  +--rw slo-sle-template?              leafref
         |     |  |  +--:(custom)
         |     |  |     +--rw service-slo-sle-policy
         |     |  |        +--rw policy-description?     string
         |     |  |        +--rw metric-bounds
         |     |  |        |  +--rw metric-bound* [metric-type]
         |     |  |        |     +--rw metric-type
         |     |  |        |     |       identityref
         |     |  |        |     +--rw metric-unit          string
         |     |  |        |     +--rw value-description?   string
         |     |  |        |     +--rw bound?               uint64
         |     |  |        +--rw security*
         |     |  |        |       identityref
         |     |  |        +--rw isolation?
         |     |  |        |       identityref
         |     |  |        +--rw max-occupancy-level?    uint8
         |     |  |        +--rw mtu?                    uint16
         |     |  |        +--rw steering-constraints
         |     |  |           +--rw path-constraints
         |     |  |           +--rw service-function
         |     |  +--rw service-slo-sle-policy-override?
         |     |  |       identityref
         |     |  +--ro connectivity-construct-monitoring
         |     |     +--ro one-way-min-delay?         uint32
         |     |     +--ro one-way-max-delay?         uint32
         |     |     +--ro one-way-delay-variation?   uint32
         |     |     +--ro one-way-packet-loss?       decimal64
         |     |     +--ro two-way-min-delay?         uint32
         |     |     +--ro two-way-max-delay?         uint32
         |     |     +--ro two-way-delay-variation?   uint32
         |     |     +--ro two-way-packet-loss?       decimal64
         |     +--ro connection-group-monitoring
         |        +--ro one-way-min-delay?         uint32
         |        +--ro one-way-max-delay?         uint32
         |        +--ro one-way-delay-variation?   uint32
         |        +--ro one-way-packet-loss?       decimal64
         |        +--ro two-way-min-delay?         uint32
         |        +--ro two-way-max-delay?         uint32
         |        +--ro two-way-delay-variation?   uint32
         |        +--ro two-way-packet-loss?       decimal64
         +--rw te-topology-identifier
            +--rw provider-id?   te-global-id
            +--rw client-id?     te-global-id
            +--rw topology-id?   te-topology-id
```

Commenté [BMI33]: Shoudln't most of these be defined as gauge ?

Figure 4

Commenté [BMI34]: Consider splitting the tree into fragments as per "If the data nodes tree is too long,
   it is also possible to split the diagram into smaller diagrams for ..." (RFC8340).

5.1. IETF Network Slice Service SLO and SLE Templates

The 'slo-sle-templates' container (Figure 4) is used by the service provider of the NSC to define and maintain a set of common IETF Network Slice templates that apply to one or several IETF Network Slice services. The exact definition of the templates is deployment specific to each network provider.

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

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

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

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

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

```
{
  "network-slice-services": {
    "slo-sle-templates": {
      "slo-sle-template": [
       {
         "id":"GOLD-template",
         "template-description": "Two-way bandwidth: 1 Gbps,\
          one-way latency 100ms "
         "service-slo-sle-policy:isolation":"service-isolation-shared"~~,~~
        },
       {
         "id":"PLATINUM-template",
         "template-description": "Two-way bandwidth: 1 Gbps,\
           one-way latency 50ms "
         "service-slo-sle-policy:isolation":"service-isolation-
dedicated"~~,~~
        }~~,~~
       ]~~,~~
      }
   }
}
```

Commenté [BMI35]:

Commenté [BMI36]: Please check the validation of JSON examples, not only this one.

5.2. IETF Network Slice Services

The 'slice-service' is the data structure that abstracts an IETF Network Slice ~~service~~Service. Each 'slice-service' is uniquely identified by an identifier: 'service-id'.

Commenté [BMI37]: Indicate the uniqueness scope

An IETF Network Slice service has the following main parameters:

- "service-id": Is an identifier that is used to uniquely identify the IETF Network Slice service within NSC.

- "service-description": Gives some description of an IETF Network Slice service.

- "status": Is used to show the operative and administrative status of the IETF Network Slice service, and can be used as indicator to detect network slice anomalies.

- "service-tags": It is a means to correlate the higher level "Customer higher level operation system" and IETF network slices. It might be used by IETF network Slice service operator to provide additional information to the ~~IETF Network Slice Controller (~~NSC~~)~~ during the automation of the IETF network slices. E.g. adding tag with "customer-name" when multiple actual customers use a same network slice service. Another use-case for "service-tag" might be for an operator to provide additional attributes to NSC which might be used during the realization of IETF Network Slice services such as type of services (e.g., L2 or L3). These additional attributes can also be used by the NSC for various use-cases such as monitoring and assurance of the IETF Network Slice services where NSC can notify the customer system by issuing the notifications. Note that all these attributes are OPTIONAL but might be useful for some use-cases.

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

- "sdp": Represents a set of ~~endpoints (~~SDPs~~) ~~ that are involved in the IETF Network Slice service with each 'sdp' belonging to a single 'slice-service'. ~~More ~~More ~~description are ~~details are provided in Section 5.2.1.

- "connection-groups": Abstracts the connections to the set of SDPs of the IETF Network Slice service.

5.2.1. IETF Network Slice Service Demarcation Points

An SDP belong to a single IETF Network Slice service. An IETF Network Slice service involves two or more SDPs. An IETF Network Slice service can be modified by adding new "sdp" or removing existing "sdp".

Section 4.2 of [I-D.ietf-teas-ietf-network-slices] describes four possible ways in which the SDP may be placed:

- Within CE

- Provider-facing ports on the CE

- Customer-facing ports on the PE
- Within PE

Commenté [BMI38]: Should there be a tag to cover cases where the SDPs are managed by the provider?

In the four options, the Attachment Circuit (AC) may be part of the IETF Network Slice service or may be external to it. Based on the definition of AC in ~~section ~~Section 2.1 of [I-D.ietf-teas-ietf-network-slices], the customer and provider may agree on a per {IETF Network Slice service, connectivity construct, and SLOs/SLEs} basis to police or shape traffic on the AC in both the ingress (CE to PE) direction and egress (PE to CE) direction, ~~This ~~this ensures that the traffic is within the capacity profile that is agreed in an IETF Network Slice service. Excess traffic is dropped by default, unless specific out-of-profile policies are agreed between the customer and the provider.

An ~~IETF Network Slice ~~SDP has several characteristics:

- "sdp-id": Uniquely identifies the SDP within the Network Slice Controller (NSC). The identifier is a string that allows any encoding for the local administration of the IETF Network Slice service.

- "location": Indicates SDP location information, which helps the NSC to identify an SDP.

- "node-id": ~~The SDP node information~~A reference to the node that hosts the SDP, which helps the NSC to identify an SDP.

Commenté [BMI39]: L3SM includes a reference to a bearer to reuse a pre-established link. I suggest we include that in the NSSM as well.

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

- "peer-sap-id": A unique identifier that identifies a given Service Attachment Point (SAP) defined in [I-D.ietf-opsawg-sap], which defines a model of an abstract view of the provider network topology that contains the points from which its services can be attached.

- "service-match-criteria": Defines matching policies for network slice service traffic to apply on a given SDP.

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

Commenté [BMI40]: I would expect at least the same details as those included in the LxSMs to be reused here.

- "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 rate-limit is applicable to the traffic from the SDP to the IETF scope Network that passes through the AC. When Bandwidth is applied to the outgoing direction, it is applied to the traffic from the IETF Network to the SDP of that particular slice service. If an SDP has multiple 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 refers to [I-D.ietf-teas-ietf-network-slices].

Commenté [BMI41]: Consider adding an explicit pointer to the section where this is discussed. Thanks.

- "sdp-peering": Specifies the protocol for an SDP for exchanging control-plane information, e.g. L1 signaling protocol or L3 routing protocols, etc.

Commenté [BMI42]: The description is not clear enough. Does this refer to the peering with the peer SAP (peer-sap-id)?

Commenté [BMI43]: Rather than having opaque or string values, why not include specific protocols that are likely to be used?

- "status": Enables the control of the operative and administrative status of the SDP, can be used as indicator to detect SDP anomalies.

The customer may choose to use an explicit "service-match-criteria" to map all the SDP's traffic or a subset of the SDP's traffic to a specific connection-group or connectivity-construct.

If an SDP is placed at the port or AC of a CE or PE, and there is only one single connectivity construct with a source at the SDP, traffic can be implicitly mapped to this connectivity construct since the port or AC can be used to identify the traffic and the SDP is the only source of the connectivity-construct. Appendix A.1 shows an example of both the implicit and explicit approaches.

If an SDP is placed within a CE or PE, or there are many connectivity constructs with a source at the SDP. Traffic needs to be explicitly mapped into the IETF Network Slice's specific connectivity construct. The policies, "service-match-criteria", are based on the values in which combination of layer 2 and layer 3 header and payload fields within a packet to identify to which {IETF Network Slice service, connectivity construct, and SLOs/SLEs} that packet is assigned.

The customer may choose to use an explicit match-type of "match-any" to map all the SDP's traffic to the appropriate connection-group or connectivity-construct.

Similarly, if a subset of traffic is matched (ie. dscp-match) and mapped to a connectivity-construct, the customer may choose to add a subsequent "match-any" to explicitly map the remaining SDP traffic to a separate connectivity-construct. If the customer chooses to implicitly map remaining traffic and if there is no additional connectivity constructs where the "sdp-id" source is specified, then that traffic will be dropped.

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

To illustrate the use of SDP options, the below are two examples. How the NSC realize the mapping is out of scope for this document.

- SDPs at customer-facing ports on the PEs: As shown in Figure 5, customer of the IETF network slice service would like to connect two SDPs to satisfy specific service, e.g., Network wholesale services. In this case, the IETF network slice SDPs are mapped to customer-facing ports of PE nodes. The IETF network slice controller (NSC) uses 'node-id' (PE device ID), 'attachment-circuit' (ACs) to map SDPs to the customer-facing ports on the PEs.

```
              SDP1                                     SDP2
       (With PE1 parameters)                       (with PE2 parameters)
               o<--------- IETF Network Slice 1 ------->o
               +     |                            |     +
               +     |<----------- S1 ----------->|     +
               +     |                            |     +
               +     |    |<------ T1 ------>|    |     +
                 +   v    v                  v    v   +
                   + +----+                  +----+ +
    +-----+    |     | PE1|==================| PE2|          +-----+
    |     |----------X    |                  |    |     |    |     |
    |     |    |     |    |                  |    X----------|     |
    |     |----------X    |                  |    |     |    |     |
    +-----+    |     |    |==================|    |     |    +-----+
               AC    +----+                  +----+     AC
    Customer         Provider                Provider        Customer
    Edge 1           Edge 1                  Edge 2           Edge 2
```

Legend:
O: Representation of the IETF network slice endpoints (SDP)
+: Mapping of SDP to customer-facing ports on the PE
X: Physical interfaces used for realization of IETF network slice service
S1: L0/L1/L2/L3 services used for realization of IETF network slice service
T1: Tunnels used for realization of IETF network slice service

Figure 5

Commenté [BMI44]: Having an example to illustrate how the model can be used with ancillary SDPs would be helpful.

- SDPs within CEs: As shown in Figure 6, customer of the IETF network slice service would like to connect two SDPs to provide connectivity between transport portion of 5G RAN to 5G Core network functions. In this scenario, the IETF network slice controller (NSC) uses 'node-id' (CE device ID) , 'sdp-ip' (CE tunnel endpoint IP), 'service-match-criteria' (VLAN tag), 'attachment-circuit' (ACs) to map SDPs to the CE tunnel endpoints. And the NSC can also retrieve the corresponding ACs, or PEs, and further map the slice service to services/tunnels/ paths.

```
           SDP3                                     SDP4
    (With CE1 parameters)                       (with CE2 parameters)
    +<----------------- IETF Network Slice 2 -------------->o
    +                                                       +
    +|<------------------------- S2 ---------------------->|+
    +|                                                     |+
    +|                 |<------ T2 ------>|                |+
    +|                 v                  v                |+
    +v            +----+                  +----+           v+
 +--+--+    |     | PE1|==================| PE2|     |    +-+---+
 |  +  X----------X    |                  |    |     |    | +   |
 |  o  |    |     |    |                  |    X----------X o   |
 |     X----------X    |                  |    |     |    |     |
 +-----+    |     |    |==================|    |     |    +-----+
            AC    +----+                  +----+     AC
 Customer         Provider                Provider         Customer
 Edge 1           Edge 1                  Edge 2           Edge 2

Legend:
O: Representation of the IETF network slice endpoints (SDP)
+: Mapping of SDP to CE
X: Physical interfaces used for realization of IETF network slice
S2: L0/L1/L2/L3 services used for realization of IETF network slice
T2: Tunnels used for realization of IETF network slice
```

Figure 6

5.2.2. IETF Network Slice Service Connectivity Constructs

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

[I-D.ietf-teas-ietf-network-slices] defines the basic connectivity construct for a network slice, and the connectivity construct may have different SLO and SLE requirements. "connectivity-construct" represents this connectivity construct, and "slo-sle-policy" under it represents the per-connectivity construct SLO and SLE requirements.

Apart from the per-connectivity construct SLO and SLE, slice service traffic is usually managed by combining similar types of traffic. For example, some connections for video services require high bandwidth, and some connections for voice over IP request low latency and reliability. "connection-group" is thus defined to treat each type as a class with per-connection-group SLO and SLE such that the connectivity construct can inherit the SLO/SLE from the group if not explicitly defined.

5.2.3. IETF Network Slice Service SLO and SLE Policy

As defined in section 4 [I-D.ietf-teas-ietf-network-slices], the SLO and SLE policy of an IETF Network Slice service defines some common attributes.

"slo-sle-policy" is used to represent specific SLO and SLE policies. During the creation of an IETF Network Slice service, the policy can be specified either by a standard SLO and SLO template or a customized SLO and SLE policy.

The policy can apply to per-network slice service, per-connection group "connection group", or per-connectivity construct "connectivity-construct". Since there are multiple mechanisms for assigning a policy to a single connectivity construct, an overridden precedence order among them is as follows:

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

The container "metric-bounds" supports all the variations and combinations of SLOs, which includes a list of "metric-bound" and each "metric-bound" could specify a particular "metric-type". "metric-type" is defined with YANG identity and supports the following options:

"service-slo-one-way-bandwidth": Indicates the guaranteed minimum bandwidth between any two SDP. And the bandwidth is unidirectional.

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

"service-slo-one-way-delay": Indicates the maximum one-way latency between two SDP.

"service-slo-two-way-delay": Indicates the maximum round-trip latency between two SDP.

"service-slo-one-way-delay-variation": Indicates the jitter constraint of the slice maximum permissible delay variation, and is measured by the difference in the one-way latency between

Commenté [BMI45]: A check based in the CC type will be needed.

sequential packets in a flow.

"service-slo-two-way-delay-variation": Indicates the jitter constraint of the slice maximum permissible delay variation, and is measured by the difference in the two-way latency between sequential packets in a flow.

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

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

"service-slo-availability": Is defined as the ratio of up-time to total_time(up-time+down-time), where up-time is the time the IETF Network Slice is available in accordance with the SLOs associated with it.

Commenté [BMI46]: What about percentile values ?

The following common SLEs are defined:

"mtu": Refers to the service MTU, which is the maximum PDU size that the customer may use.

"security": Includes the request for encryption or other security techniques to traffic flowing between the two ~~NS endpoints~~SDPs.

"isolation": Specifies the isolation level that a customer expects, including dedicated, shared, or other level.

max-occupancy-level: Specifies the number of flows to be admitted and optionally a maximum number of countable resource units (e.g., IP or MAC addresses) an IETF Network Slice service can consume.

"steering-constraints": Specifies the constraints how the provider routes traffic for the IETF Network Slice service.

The following shows an example where a network slice policy can be configured:

```
{
  "slice-services": {
    "slice-service": {
      "service-id": "exp-slice",
      "service-slo-sle-policy": {
        "policy-description":"video-service-policy",
        "metric-bounds": {
            "metric-bound": [
             {
                "metric-type": "service-slo-one-way-bandwidth",
                "metric-unit": "mbps",
                "bound": "1000"
             },
             {
                "metric-type": "service-slo-availability",
                "bound": "99.9%"
             },
```

Commenté [BMI47]: Please check that all JSONs are valid

```
                    ]~~,~~
                  }
                }
              }
            }
          }
```

For a multiple connectivity-construct slice service, when a connection-group or a connectivity-construct has some specific requirements, such as bandwidth or latency, that are different from those defined of the slice-service, the new set of SLO&SLE with full or partial override can be applied. "service-slo-sle-policy-override" is used to specify the requirements.

5.2.4. IETF Network Slice Service Monitoring

Commenté [BMI48]: I would used « Fulfillment/Assurance »

An IETF Network Slice service defines connectivity with specific SLO characteristics, including bandwidth, latency, etc. The connectivity is a combination of logical unidirectional connections, represented by 'connectivity-construct'.

This model also describes performance status of an IETF Network Slice. The statistics are described in the following granularity:

- Per SDP: specified in 'sdp-monitoring' under the "sdp".

- Per connectivity construct: specified in 'connectivity-construct-monitoring' under the "connectivity-construct".

- Per connection group: specified in 'connection-group-monitoring' under the "connection-group".

This model does not define monitoring enabling methods. The mechanism defined in [RFC8640] and [RFC8641] can be used for either periodic or on-demand subscription.

By specifying subtree filters or xpath filters to "sdp", "connectivity-construct", or "connection-group", so that only interested contents will be sent. These mechanisms can be used for monitoring the IETF Network Slice performance status so that the customer management system could initiate modification based on the IETF Network Slice running status.

5.2.5. IETF Network Slice Service on Abstract TE topology

The IETF Network Slice customer might ask for some level of control of, e.g., to customize the service paths in a network slice.

Commenté [BMI49]: Does 8795 allow to indicate non-via points (e.g., exclude AS, exclude region)

Section 3.13 in [RFC8795] defines abstract TE topology concept to accommodate both the provider's resource capability and the customer's preferences. The abstract TE topology is a topology that contains abstract topological elements (nodes, links, tunnel termination points). The following nodes are the extensions for this use case

> "te-topology-identifier": The container under the list "slice-service" is defined to reference the prebuilt abstract TE topology as a customized topology constraint for a slice service.

"ltp": A reference to Link Termination Point (LTP) in the TE-topology, under the list "sdp", is used to associate an SDP with the customized topology to create point-to-point abstract TE links. These abstract links can be used as the underlying links of the connectivity-construct when a NS service is created.

The model can be extended if some implementations require path control with specific constraints.

6. IETF Network Slice Service Module

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

```
<CODE BEGINS> file "ietf-network-slice-service@2022-10-24.yang"
module ietf-network-slice-service {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-network-slice-service";
  prefix ietf-nssm;

  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Types.";
  }
  import ietf-vpn-common {
    prefix vpn-common;
    reference
      "RFC 9181: A Common YANG Data Model for Layer 2 and Layer 3
                 VPNs.";
  }
  import ietf-network {
    prefix nw;
    reference
      "RFC 8345: A YANG Data Model for Network Topologies";
  }
  import ietf-network-topology {
    prefix nt;
    reference
      "RFC 8345: A YANG Data Model for Network
       Topologies, Section 6.2";
  }

  /* Import TE Topology */

  import ietf-te-topology {
    prefix tet;
    reference
      "RFC 8795: YANG Data Model for Traffic Engineering (TE)
       Topologies";
  }
  import ietf-te-types {
    prefix te-types;
    reference
      "RFC 8776: Common YANG Data Types for Traffic Engineering.";
  }
```

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

     organization
       "IETF Traffic Engineering Architecture and Signaling (TEAS)
        Working Group";
     contact
       "WG Web:  <https://tools.ietf.org/wg/teas/>
        WG List:  <mailto:teas@ietf.org>

        Editor: Bo Wu
             <lana.wubo@huawei.com>
        Editor: Dhruv Dhody
             <dhruv.ietf@gmail.com>
        Editor: Reza Rokui
             <reza.rokui@nokia.com>
        Editor: Tarek Saad
             <tsaad@cisco.com>
        Author: Liuyan Han
             <hanliuyan@chinamobile.com>
        Editor: John Mullooly
             <jmullool@cisco.com>";
     description
       "This module defines a model for the IETF Network Slice
~~service~~Service.

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

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

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

     revision 2022-10-24 {
       description
         "initial version.";
       reference
         "RFC XXXX: A ~~Yang ~~YANG Data Model for the IETF Network Slice
~~service~~Service
          ~~operation~~";
     }
     /* Features */
     /* Identities */

     identity service-tag-type {
       description
         "Base identity for IETF Network Slice service tag type.";
     }
```

```
identity service-tag-customer {
  base service-tag-type;
  description
    "The IETF Network Slice service customer ID tag type.";
}

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

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

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

identity attachment-circuit-tag-vlan-id {
  base attachment-circuit-tag-type;
  description
    "The attachment circuit VLAN ID tag type.";
}

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

identity service-isolation-type {
  description
    "Base identity for IETF Network slice service isolation level.";
}
identity service-isolation-shared {
  base service-isolation-type;
  description
    "Shared resources (e.g. queues) are associated with the
     slice service traffic. Hence, the traffic can be impacted
     by effects of other services traffic
     sharing the same resources.";
}

identity service-isolation-dedicated {
  base service-isolation-type;
  description
    "Dedicated resources (e.g. queues) are associated with the
     Network Slice service traffic. Hence, the service traffic
     is isolated from other servce is traffic
     sharing the same resources.";
}

identity service-security-type {
```

Commenté [BMI50]: Define what is a customer tag

a mis en forme : Surlignage

Commenté [BMI51]: Why not reusing the same formalism as in L3SM?

```
    description
      "Base identity for ~~for ~~slice service security level.";
  }

  identity service-security-authenticate {
    base service-security-type;
    description
      "Indicates the slice service requires authentication.";
  }

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

  identity service-security-encryption {
    base service-security-type;
    description
      "Indicates the slice service requires data encryption.";
  }
```

Commenté [BMI52]: What if both authentication and integrity is required?

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

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

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

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

  identity receiver-role {
    base vpn-common:role;
    description
      "An SDP is acting as a receiver.";
  }

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

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

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

  identity service-slo-shared-bandwidth {
    base service-slo-metric-type;
    description
      "The shared SLO bandwidth bound. It is the limit on the
       bandwidth that can be shared amongst a group of
       connectivity constructs of a slice service.";
  }

  identity service-slo-one-way-delay {
    base service-slo-metric-type;
    description
      "SLO one-way-delay is the upper bound of network delay when
       transmitting between two SDPs. The metric is defined in
       RFC7679.";
  }

  identity service-slo-two-way-delay {
    base service-slo-metric-type;
    description
      "SLO two-way delay is the upper bound of network delay when
       transmitting between two SDPs. The metric is defined in
       RFC2681.";
  }

  identity service-slo-one-way-delay-variation {
    base service-slo-metric-type;
    description
      "SLO one-way delay variation is defined by RFC3393, is the
       difference in the one-way delay between sequential packets
       between two SDPs.";
  }

  identity service-slo-two-way-delay-variation {
    base service-slo-metric-type;
    description
      "SLO two-way delay variation is defined by RFC5481, is the
```

```
       difference in the round-trip delay between sequential packets
       between two SDPs.";
  }

  identity service-slo-one-way-packet-loss {
    base service-slo-metric-type;
    description
      "SLO loss metric. The ratio of packets dropped to packets
       transmitted between two SDPs in one-way
       over a period of time as specified in RFC7680.";
  }

  identity service-slo-two-way-packet-loss {
    base service-slo-metric-type;
    description
      "SLO loss metric. The ratio of packets dropped to packets
       transmitted between two SDPs in two-way
       over a period of time as specified in RFC7680.";
  }

  identity service-slo-availability {
    base service-slo-metric-type;
    description
      "SLO availability level.";
  }

  identity service-match-type {
    description
      "Base identity for IETF Network Slice service traffic
       match type.";
  }

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

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

  identity service-label-match {
    base service-match-type;
    description
      "Use the MPLS label as match criteria for the slice service
       traffic.";
  }

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

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

  identity service-dscp-match {
    base service-match-type;
    description
      "Use DSCP in the IP packet header as match criteria
       for the slice service traffic.";
  }

  identity service-acl-match {
    base service-match-type;
    description
      "Use Access Control List (ACL) as match criteria
       for the slice service traffic~~, ~~.";
    reference
      "RFC 8519: YANG Data Model for
       Network Access Control Lists (ACLs)";
  }

  identity service-any-match {
    base service-match-type;
    description
      "Match all slice service traffic.";
  }

  identity peering-protocol-type {
    description
      "Base identity for SDP peering protocol type.";
  }

  identity peering-protocol-bgp {
    base peering-protocol-type;
    description
      "Use BGP as protocol for SDP peering with customer device.";
  }

  identity peering-static-routing {
    base peering-protocol-type;
    description
      "Use static routing for SDP peering with customer device.";
  }

  identity peering-attribute-type {
    description
      "Base identity for BGP peering";
  }

  identity remote-as {
    base peering-attribute-type;
    description
      "Identity for remote-as attribute of BGP peering.";
  }

  identity neighbor {
    base peering-attribute-type;
    description
      "Identity for neighbor attribute of BGP peering.";
  }

  identity local-as {
    base peering-attribute-type;
    description
      "Identity for local-as attribute of BGP peering.";
  }

  /*
   * Identity for availability-type
   */

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

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

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

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

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

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

  identity slo-sle-policy-override {
    description
```

```
         "Base identity for SLO/SLE policy override options.";
     }

     identity slo-sle-policy-full-override {
       base slo-sle-policy-override;
       description
         "The policy of SLO/SLE(s) that is defined at a
          child level override a parent SLO/SLE policy,
          which means that no SLO/SLE(s) are ~~inheritied~~inherited from
parent
          if a child SLO/SLE policy exists.";
     }

     identity slo-sle-policy-partial-override {
       base slo-sle-policy-override;
       description
         "The policy of SLO/SLE(s) that is defined at a
          child level updates the parent SLO/SLE policy.
          For example, if a specific SLO is defined
          at the child level, that specific SLO overrides the
          one inherited from a parent SLO/SLE policy, while all other
          SLOs in the parent SLO-SLE policy still apply.";
     }

     /* grouping */

     grouping service-sles {
       description
         "Indirectly Measurable Objectives of a slice service.";
       leaf-list security {
         type identityref {
           base service-security-type;
         }
         description
           "The slice service security SLE(s)";
       }
       leaf isolation {
         type identityref {
           base service-isolation-type;
         }
         default "service-isolation-shared";
         description
           "The slice service isolation SLE requirement.";
       }
       leaf max-occupancy-level {
         type uint8 {
           range "1..100";
         }
         description
           "The maximal occupancy level specifies the number of flows to
            be admitted.";
       }
       leaf mtu {
         type uint16;
         units "bytes";
         description
           "The MTU specifies the maximum length in octets of data
            packets that can be transmitted by the slice service.
            The value needs to be less than or equal to the
            minimum MTU value of all 'attachment-circuits' in the SDPs.";
       }
       container steering-constraints {
         description
           "Container for the policy of steering constraints
            applicable to the slice service.";
         container path-constraints {
           description
             "Container for the policy of path constraints
              applicable to the slice service.";
         }
         container service-function {
           description
             "Container for the policy of service function
              applicable to the slice service.";
         }
       }
     }

     grouping service-metric-bounds {
       description
         "Slice service metric bounds grouping.";
       container metric-bounds {
         description
           "Slice service metric bounds container.";
         list metric-bound {
           key "metric-type";
           description
             "List of slice service metric bounds.";
           leaf metric-type {
             type identityref {
               base service-slo-metric-type;
             }
             description
               "Identifies an entry in the list of metric type
                bounds for the slice service.";
           }
           leaf metric-unit {
             type string;
             mandatory true;
             description
               "The metric unit of the parameter. For example,
                s, ms, ns, and so on.";
           }
           leaf value-description {
             type string;
             description
               "The description of previous value.";
           }
           leaf bound {
             type uint64;
             default "0";
             description
               "The Bound on the slice service connection metric.
                A zero indicate an unbounded upper limit for the
                specific metric-type.";
           }
         }
       }
     }

     grouping sdp-peering {
       description
         "A grouping for the slice service SDP peering.";
       container sdp-peering {
         description
           "Describes SDP peering attributes.";
         list protocol {
           key "protocol-type";
           description
             "List of the SDP peering protocol.";
           leaf protocol-type {
             type identityref {
               base peering-protocol-type;
             }
             description
               "Identifies an entry in the list of SDP peering
                protocol type.";
           }
           list attribute {
             key "attribute-type";
             description
               "list of protocol attributes";
             leaf attribute-type {
               type identityref {
                 base peering-attribute-type;
               }
               description
                 "identifies the attribute type";
             }
             leaf-list value {
               type string;
               description
                 "Describes the value of protocol attribute, e.g.
                  nexthop address, peer address, etc.";
             }
           }
         }
         list opaque {
           key "attribute-name";
           description
             "List of protocol attributes.";
           leaf attribute-name {
             type string;
             description
               "The name of the attribute.";
           }
           leaf-list value {
             type string;
             description
               "The value(s) of the attribute";
           }
         }
       }
     }
```

```
grouping sdp-attachment-circuits {
  description
    "Grouping for the SDP attachment circuit definition.";
  container attachment-circuits {
    description
      "List of attachment circuit.";
    list attachment-circuit {
      key "ac-id";
      description
        "The IETF Network Slice service SDP attachment circuit
         related parameters.";
      leaf ac-id {
        type string;
        description
          "Uniquely identifier a attachment circuit.";
      }
      leaf ac-description {
        type string;
        description
          "The attachment circuit description.";
      }
      leaf ac-node-id {
        type string;
        description
          "The attachment circuit node ID in the case of
           multi-homing.";
      }
      leaf ac-tp-id {
        type string;
        description
          "The termination port ID of the attachment circuit.";
      }
      leaf ac-ip-address {
        type inet:ip-address;
        description
          "The IP address of the attachment circuit.";
      }
      leaf ac-ip-prefix-length {
        type uint8;
        description
          "The subnet prefix length expressed in bits.";
      }
      leaf peer-sap-id {
        type string;
        description
          "Indicates an identifier of the peer's termination
           identifier (e.g., Customer Edge (CE)). This
           information can be used for correlation purposes,
           such as identifying the SAP that is attached to
           an endpoint that is provided in a service request.";
      }
      reference
        " draft-ietf-opsawg-sap: A YANG Network Model for Service
         Attachment Points (SAPs)";
      leaf mtu {
        type uint16;
        units "bytes";
```

```
          description
            "Maximum size in octets of the slice service data packet
             that can traverse an SDP.";
        }
        container ac-tags {
          description
            "Container for the attachment circuit tags.";
          list ac-tags {
            key "ac-tag-type";
            description
              "The attachment circuit tags list.";
            leaf ac-tag-type {
              type identityref {
                base attachment-circuit-tag-type;
              }
              description
                "The attachment circuit tag type.";
            }
            leaf-list value {
              type string;
              description
                "The attachment circuit tag value.";
            }
          }
          list ac-tag-opaque {
            key "tag-name";
            description
              "The attachment circuit tag opaque list.";
            leaf tag-name {
              type string;
              description
                "The opaque tags name";
            }
            leaf-list value {
              type string;
              description
                "The opaque tags value";
            }
          }
        }
        /* Per ac rate limits */
        uses service-qos;
        uses sdp-peering;
      }
    }
  }

  grouping sdp-monitoring-metrics {
    description
      "Grouping for the SDP monitoring metrics.";
    container sdp-monitoring {
      config false;
      description
        "Container for SDP monitoring metrics.";
      leaf incoming-utilized-bandwidth {
        type te-types:te-bandwidth;
        description
          "Incoming bandwidth utilization at an SDP.";
      }
      leaf incoming-bw-utilization {
        type decimal64 {
          fraction-digits 5;
          range "0..100";
        }
        units "percent";
        mandatory true;
        description
          "To be used to define the bandwidth utilization
           as a percentage of the available bandwidth.";
      }
      leaf outgoing-utilized-bandwidth {
        type te-types:te-bandwidth;
        description
          "Outoing service bandwidth utilization at an SDP.";
      }
      leaf outgoing-bw-utilization {
        type decimal64 {
          fraction-digits 5;
          range "0..100";
        }
        units "percent";
        mandatory true;
        description
          "To be used to define the service bandwidth utilization
           as a percentage of the available bandwidth.";
      }
    }
  }

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

  grouping geolocation-container {
    description
      "A grouping containing a GPS location.";
    container location {
      description
        "A container containing a GPS location.";
      leaf altitude {
        type int64;
        units "millimeter";
        description
          "Distance above the sea level.";
      }
      leaf latitude {
        type decimal64 {
          fraction-digits 8;
          range "-90..90";
        }
        description
          "Relative position north or south on the Earth's surface.";
      }
      leaf longitude {
        type decimal64 {
          fraction-digits 8;
          range "-180..180";
        }
        description
          "Angular distance east or west on the Earth's surface.";
      }
    }
    // gps-location
  }

  // geolocation-container

  grouping bw-rate-limits {
    description
      "Bandwidth rate limits grouping.";
    reference
      "RFC 7640: Traffic Management Benchmarking";
    leaf cir {
      type uint64;
      units "bps";
      description
        "Committed Information Rate. The maximum number of bits
         that a port can receive or send during one-second over an
         interface.";
    }
    leaf cbs {
      type uint64;
      units "bytes";
      description
        "Committed Burst Size. CBS controls the bursty nature
         of the traffic. Traffic that does not use the configured
         CIR accumulates credits until the credits reach the
         configured CBS.";
    }
    leaf eir {
      type uint64;
      units "bps";
      description
        "Excess Information Rate, i.e., excess frame delivery
         allowed not subject to SLA. The traffic rate can be
         limited by EIR.";
    }
    leaf ebs {
      type uint64;
      units "bytes";
      description
        "Excess Burst Size. The bandwidth available for burst
         traffic from the EBS is subject to the amount of
         bandwidth that is accumulated during periods when
         traffic allocated by the EIR policy is not used.";
    }
    leaf pir {
      type uint64;
      units "bps";
      description
        "Peak Information Rate, i.e., maximum frame delivery
```

```
         allowed. It is equal to or less than sum of CIR and EIR.";
    }
    leaf pbs {
      type uint64;
      units "bytes";
      description
        "Peak Burst Size.";
    }
  }

  grouping service-qos {
    description
      "The rate limits grouping.";
    container incoming-qos-policy {
      description
        "Container for the asymmetric traffic control.";
      leaf qos-policy-name {
        type string;
        description
          "The name of the QoS policy that is applied to the
           attachment circuit. The name can reference a QoS
           profile that is pre-provisioned on the device.";
      }
      container rate-limits {
        description
          "Container for the asymmetric traffic control.";
        uses bw-rate-limits;
      }
    }
    container outgoing-qos-policy {
      description
        "The QoS policy imposed on outgoing traffic.";
      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;
      }
    }
  }

  grouping sdp {
    description
      "Slice service SDP related information";
    leaf sdp-id {
      type string;
      description
        "Unique identifier for the referred slice service SDP.";
    }
    leaf sdp-description {
      type string;
      description
        "Give more description of the SDP.";
    }
    uses geolocation-container;
    leaf node-id {
      type string;
      description
        "Uniquely identifies an edge node of the SDP.";
    }
    leaf sdp-ip {
      type inet:ip-address;
      description
        "The IP address of the SDP.";
    }
    leaf peer-sap-id {
      type string;
      description
        "Indicates an identifier of the peer's termination
         identifier (e.g., Customer Edge (CE)). This
         information can be used for correlation purposes,
         such as identifying the SAP that is attached to
         an endpoint that is provided in a service request.";
      reference
        " draft-ietf-opsawg-sap: A YANG Network Model for Service
         Attachment Points (SAPs)";
    }
    leaf ltp {
      type leafref {
        path "/nw:networks/nw:network/nw:node/"
           + "nt:termination-point/tet:te-tp-id";
      }
      description
        "A reference to Link Termination Point (LTP) in the
         abstract TE topology";
      reference
        "RFC 8795: YANG Data Model for Traffic Engineering (TE)
         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 entry index.";
        }
        leaf match-type {
          type identityref {
            base service-match-type;
          }
          mandatory true;
          description
            "Identifies an entry in the list of the slice service
             match criteria.";
        }
```

```
      leaf-list value {
        type string;
        description
          "Describes the slice service match criteria, e.g.
           IP prefix, VLAN, etc.";
      }
      leaf target-connection-group-id {
        type leafref {
          path "../../../../../ietf-nss:connection-groups"
             + "/ietf-nss:connection-group"
             + "/ietf-nss:connection-group-id";
        }
        mandatory true;
        description
          "Reference to the slice service connection group.";
      }
      leaf connection-group-sdp-role {
        type identityref {
          base vpn-common:role;
        }
        default "vpn-common:any-to-any-role";
        description
          "Indicates the role in the connection group when
           a slice service has multiple multipoint-to-multipoint
           connection groups, e.g., hub-spoke.";
      }
      leaf target-connectivity-construct-id {
        type leafref {
          path
            "/ietf-nss:network-slice-services"
          + "/ietf-nss:slice-service"
          + "/ietf-nss:connection-groups"
          + "/ietf-nss:connection-group[connection-group-id"
          + "=current()/../target-connection-group-id]"
          + "/ietf-nss:connectivity-construct/ietf-nss:cc-id";
        }
        description
          "Reference to a Network Slice connection construct.";
      }
    }
  }
  uses service-qos;
  uses sdp-peering;
  uses sdp-attachment-circuits;
  uses vpn-common:service-status;
  uses sdp-monitoring-metrics;
}

//service-sdp

grouping connectivity-construct {
  description
    "Grouping for slice service connectivity construct.";
  list connectivity-construct {
    key "cc-id";
    description
      "List of connectivity constructs.";
    leaf cc-id {
```

```
  type uint32;
  description
    "The connectivity construct identifier.";
}
choice connectivity-construct-type {
  default "p2p";
  description
    "Choice for connectivity construct type.";
  case p2p {
    description
      "P2P connectivity construct.";
    leaf p2p-sender-sdp {
      type leafref {
        path "../../../../sdps/sdp/sdp-id";
      }
      description
        "Reference to a sender SDP.";
    }
    leaf p2p-receiver-sdp {
      type leafref {
        path "../../../../sdps/sdp/sdp-id";
      }
      description
        "Reference to a receiver SDP.";
    }
  }
  case p2mp {
    description
      "P2MP connectivity construct.";
    leaf p2mp-sender-sdp {
      type leafref {
        path "../../../../sdps/sdp/sdp-id";
      }
      description
        "Reference to a sender SDP.";
    }
    leaf-list p2mp-receiver-sdp {
      type leafref {
        path "../../../../sdps/sdp/sdp-id";
      }
      description
        "Reference to a receiver SDP.";
    }
  }
  case a2a {
    description
      "A2A connectivity construct.";
    list a2a-sdp {
      key "sdp-id";
      description
        "List of included A2A SDPs.";
      leaf sdp-id {
        type leafref {
          path "../../../../../sdps/sdp/sdp-id";
        }
        description
          "Reference to an SDP.";
      }
```

```
          uses service-slo-sle-policy;
        }
      }
    }
    uses service-slo-sle-policy;
    /* Per connectivity construct service-slo-sle-policy
     * overrides the per slice service-slo-sle-policy.
     */
    uses service-slo-sle-policy-override;
    container connectivity-construct-monitoring {
      config false;
      description
        "SLO status per connectivity construct.";
      uses connectivity-construct-monitoring-metrics;
    }
  }
}

//connectivity-construct

grouping connection-group {
  description
    "Grouping for slice service connection group.";
  leaf connection-group-id {
    type string;
    description
      "The connection group identifier.";
  }
  leaf connectivity-type {
    type identityref {
      base vpn-common:vpn-topology;
    }
    default "vpn-common:any-to-any";
    description
      "Connection group connectivity type.";
  }
  uses service-slo-sle-policy;
  uses service-slo-sle-policy-override;
  uses connectivity-construct;
  /* Per connection group service-slo-sle-policy overrides
   * the per slice service-slo-sle-policy.
   */
  container connection-group-monitoring {
    config false;
    description
      "SLO status per connection group.";
    uses connectivity-construct-monitoring-metrics;
  }
}

//connection-group

grouping slice-service-template {
  description
    "Grouping for slice service templates.";
  container slo-sle-templates {
    description
      "Contains a set of slice service templates.";
    list slo-sle-template {
      key "id";
      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 template-description {
        type string;
        description
          "Description of the SLO and SLE policy template.";
      }
      leaf template-ref {
        type leafref {
          path "/ietf-nss:network-slice-services"
             + "/ietf-nss:slo-sle-templates"
             + "/ietf-nss:slo-sle-template"
             + "/ietf-nss:id";
        }
        description
          "The reference to a standard template. When set it
            indicates the base template over which further
            SLO/SLE policy changes are made.";
      }
      container service-slo-sle-policy {
        description
          "Contains the SLO and SLE policy.";
        leaf policy-description {
          type string;
          description
            "Description of the SLO and SLE policy.";
        }
        uses service-metric-bounds;
        uses service-sles;
      }
    }
  }
}

/* Configuration data nodes */

grouping service-slo-sle-policy {
  description
    "Slice service policy grouping.";
  choice slo-sle-policy {
    description
      "Choice for SLO and SLE policy template.

       Can be standard template or customized template.";
    case standard {
      description
        "Standard SLO template.";
      leaf slo-sle-template {
        type leafref {
          path "/ietf-nss:network-slice-services"
             + "/ietf-nss:slo-sle-templates"
             + "/ietf-nss:slo-sle-template"
             + "/ietf-nss:id";
        }
        description
          "Standard SLO and SLE template to be used.";
      }
    }
    case custom {
      description
        "Customized SLO and SLE template.";
      container service-slo-sle-policy {
        description
          "Contains the SLO and SLE policy.";
        leaf policy-description {
          type string;
          description
            "Description of the SLO and SLE policy.";
        }
        uses service-metric-bounds;
        uses service-sles;
      }
    }
  }
}

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

container network-slice-services {
  description
    "Containes a list of IETF network slice services";
  uses slice-service-template;
  list slice-service {
    key "service-id";
    description
      "A slice service is identified by a service-id.";
    leaf service-id {
      type string;
      description
        "A unique slice service identifier.";
    }
    leaf service-description {
      type string;
      description
        "Textual description of the slice service.";
    }
```

```
container service-tags {
  description
    "Container for the list of service tags.";
  list tag-type {
    key "tag-type";
    description
      "The service tag list.";
    leaf tag-type {
      type identityref {
        base service-tag-type;
      }
      description
        "Slice service tag type.";
    }
    leaf-list value {
      type string;
      description
        "The tag value";
    }
  }
  list tag-opaque {
    key "tag-name";
    description
      "The service tag opaquelist.";
    leaf tag-name {
      type string;
      description
        "The opaque tag name";
    }
    leaf-list value {
      type string;
      description
        "The opaque tag value";
    }
  }
}
uses service-slo-sle-policy;
uses vpn-common:service-status;
container sdps {
  description
    "Slice service SDPs.";
  list sdp {
    key "sdp-id";
    uses sdp;
    description
      "List of SDPs in this slice service.";
  }
}
container connection-groups {
  description
    "Contains connections group.";
  list connection-group {
    key "connection-group-id";
    description
      "List of connection groups.";
    uses connection-group;
  }
}
```

```
      uses te-types:te-topology-identifier;
    }
    //ietf-network-slice-service list
  }
}
<CODE ENDS>
```

7. Security Considerations

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

The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

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

o /ietf-network-slice-service/network-slice-services/slice-service

The entries in the list above include the whole network configurations corresponding with the slice service which the higher management system requests, and indirectly create or modify the PE or P device configurations. Unexpected changes to these entries could lead to service disruption and/or network misbehavior.

8. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made:

URI: urn:ietf:params:xml:ns:yang:ietf-network-slice-service
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

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

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

9. Acknowledgments

The authors wish to thank Mohamed Boucadair, Kenichi Ogaki, Sergio

Belotti, Qin Wu, Yao Zhao, Susan Hares, Eric Grey, Daniele Ceccarelli, Ryan Hoffman, Adrian Farrel, Aihua Guo, Italo Busi, and many others for their helpful comments and suggestions.

10. Contributors

The following authors contributed significantly to this document:

Luis M. Contreras
Telefonica
Spain

Email: luismiguel.contrerasmurillo@telefonica.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>.

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

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

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

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

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018,

<https://www.rfc-editor.org/info/rfc8340>.

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

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

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

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

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

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

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

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

11.2. Informative References

[I-D.ietf-opsawg-sap] Boucadair, M., de Dios, O. G., Barguil, S., Wu, Q., and V. Lopez, "A YANG Network Model for Service Attachment Points (SAPs)", Work in Progress, Internet-Draft, draft-ietf-opsawg-sap-10, 4 October 2022, <https://www.ietf.org/archive/id/draft-ietf-opsawg-sap-10.txt>.

[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-15, 11 July 2022, <https://www.ietf.org/archive/id/draft-ietf-teas-actn-vn-yang-15.txt>.

[I-D.ietf-teas-ietf-network-slices]
Farrel, A., Drake, J., Rokui, R., Homma, S., Makhijani, K., Contreras, L. M., and J. Tantsura, "Framework for IETF Network Slices", Work in Progress, Internet-Draft, draft-ietf-teas-ietf-network-slices-15, 21 October 2022, <https://www.ietf.org/archive/id/draft-ietf-teas-ietf-network-slices-15.txt>.

[I-D.liu-teas-transport-network-slice-yang]
Liu, X., Tantsura, J., Bryskin, I., Contreras, L. M., Wu, Q., Belotti, S., and R. Rokui, "IETF Network Slice YANG Data Model", Work in Progress, Internet-Draft, draft-liu-teas-transport-network-slice-yang-05, 6 March 2022, <https://www.ietf.org/archive/id/draft-liu-teas-transport-network-slice-yang-05.txt>.

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

Appendix A. Examples of Network Slice Services

A.1. Example-1: Two Any-to-any Slice Services with different match approachs

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

* IETF Network Slice 1 on SDP1, SDP11a, and SDP4, with any-to-any connectivity type. This is a L3 slice service and using the uniform low-latency slo-sle-template policy between all SDPs. These SDPs will also have AC eBGP peering sessions with unmanaged CE elements.

* IETF Network Slice 2 on SDP2, SDP11b, with any-to-any connectivity type. This is a L3 slice service and using the uniform high-BW slo-sle-template policy between all SDPs. These SDPs will also have AC eBGP peering sessions with unmanaged CE elements.

Slice 1 uses the explicit match approach for mapping SDP traffic to a connectivity-construct, while slice 2 uses the implicit approach. Both approaches are supported.

Note: These two slices both use service-tags of "L3". This service-tag is operator defined and has no specific meaning in the YANG model other to give a "hint" to the NSC on the Service Expectation being L3 forwarding. Other examples we may choose to eliminate it. The usage of this tag is arbitrary and up to the operator and the NSC on it's need and usage.

```
+--------+         192.0.2.1
|CE1     o------/  VLAN100
+--------+      |  SDP1 +------+
+--------+      +------o|  PE A+---------------+
|CE2     o--------/-----o|      |               |
+--------+         SDP2 +---+--+               |
               198.51.100.1 |                  |    192.0.4.1
                  VLAN200   |              +---+--+ VLAN100
                            |              |      | SDP4      +--------+
                            |              |PE C  o-----/-----o CE21   |
+--------+        192.0.3.1 |              +---+--+           +--------+
|        o------/ VLAN101   |                  |
|        |      | SDP11a+---+---+              |
|CE11    |      +------o|PE B   +--------------+
|        o-------/-----o|       |
+--------+        SDP11b+------ +
                  198.51.101.1
                  VLAN201
```

Commenté [BMI53]: How to cover dual-stack?

```
{
  "data": {
    "ietf-network-slice-service:network-slice-services": {
      "slo-sle-templates": {
        "slo-sle-template": [
          {
            "id": "high-BW-template",
            "template-description": "take the highest BW forwarding path"
          },
          {
            "id": "low-latency-template",
            "template-description": "lowest possible ~~latenceey~~latency
forwarding behavior"
          }
        ]
      },
      "slice-service": [
        {
          "service-id": "slice1",
          "service-description": "example slice1",
          "service-tags": {
            "tag-type": [
              {
                "tag-type": "ietf-nss:service-tag-service",
                "value": [
                  "L3"
                ]
              }
            ]
          },
          "slo-sle-template": "low-latency-template",
          "status": {},
          "sdps": {
            "sdp": [
              {
                "sdp-id": "1",
                "node-id": "PE-A",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-network-slice-service:service-
any-match",
                      "target-connection-group-id": "matrix1",
                      "target-connectivity-construct-id": 1
```

Commenté [BMI54]: Shouldn't this be CE1?

```
          }
         ]
        },
        "attachment-circuits": {
         "attachment-circuit": [
          {
           "ac-id": "ac1",
           "ac-description": "AC1 connected to device 1",
           "ac-node-id": "PE-A",
           "ac-tp-id": "GigabitEthernet5/0/0/0.100",
           "ac-ip-address": "192.0.2.1",
           "ac-ip-prefix-length": 24,
           "ac-tags": {
            "ac-tags": [
             {
              "ac-tag-type": "ietf-nss:attachment-circuit-
tag-vlan-id",
              "value": [
               "100"
              ]
             }
            ]
           },
           "sdp-peering": {
            "protocol": [
             {
              "protocol-type": "ietf-nss:peering-protocol-
bgp",
              "attribute": [
               {
                "attribute-type": "ietf-nss:neighbor",
                "value": [
                 "192.0.2.2"
                ]
               },
               {
                "attribute-type": "ietf-nss:remote-as",
                "value": [
                 "64001"
                ]
               },
               {
                "attribute-type": "ietf-nss:local-as",
                "value": [
                 "64000"
                ]
               }
              ]
             }
            ]
           }
          }
         ]
        },
        "status": {}
       },
       {
        "sdp-id": "3a",
```

a mis en forme : Surlignage

a mis en forme : Couleur de police : Rouge

```
                "node-id": "PE-B",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-network-slice-service:service-
any-match",
                      "target-connection-group-id": "matrix1",
                      "target-connectivity-construct-id": 1
                    }
                  ]
                },
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "ac3a",
                      "ac-description": "AC3a connected to device 3",
                      "ac-node-id": "PE-B",
                      "ac-tp-id": "GigabitEthernet8/0/0/4.101",
                      "ac-ip-address": "192.0.3.1",
                      "ac-ip-prefix-length": 24,
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "ac-tag-type": "ietf-nss:attachment-circuit-
tag-vlan-id",
                            "value": [
                              "101"
                            ]
                          }
                        ]
                      },
                      "sdp-peering": {
                        "protocol": [
                          {
                            "protocol-type": "ietf-nss:peering-protocol-
bgp",
                            "attribute": [
                              {
                                "attribute-type": "ietf-nss:neighbor",
                                "value": [
                                  "192.0.3.2"
                                ]
                              },
                              {
                                "attribute-type": "ietf-nss:remote-as",
                                "value": [
                                  "64002"
                                ]
                              },
                              {
                                "attribute-type": "ietf-nss:local-as",
                                "value": [
                                  "64000"
                                ]
                              }
                            ]
                          }
                        ]
                      }
                    }
                  ]
                },
                "status": {}
              },
              {
                "sdp-id": "4",
                "node-id": "PE-C",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-network-slice-service:service-
any-match",
                      "target-connection-group-id": "matrix1",
                      "target-connectivity-construct-id": 1
                    }
                  ]
                },
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "ac4",
                      "ac-description": "AC4 connected to device 4",
                      "ac-node-id": "PE-C",
                      "ac-tp-id": "GigabitEthernet4/0/0/3.100",
                      "ac-ip-address": "192.0.4.1",
                      "ac-ip-prefix-length": 24,
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "ac-tag-type": "ietf-nss:attachment-circuit-
tag-vlan-id",
                            "value": [
                              "100"
                            ]
                          }
                        ]
                      },
                      "sdp-peering": {
                        "protocol": [
                          {
                            "protocol-type": "ietf-nss:peering-protocol-
bgp",
                            "attribute": [
                              {
                                "attribute-type": "ietf-nss:neighbor",
                                "value": [
                                  "192.0.4.2"
                                ]
                              },
                              {
                                "attribute-type": "ietf-nss:remote-as",
                                "value": [
                                  "64004"
                                ]
                              },
                              {
                                "attribute-type": "ietf-nss:local-as",
                                "value": [
                                  "64000"
                                ]
                              }
                            ]
                          }
                        ]
                      }
                    }
                  ]
                },
                "status": {}
              }
            ]
          },
          "connection-groups": {
            "connection-group": [
              {
                "connection-group-id": "matrix1",
                "connectivity-type": "ietf-vpn-common:any-to-any",
                "connectivity-construct": [
                  {
                    "cc-id": 1,
                    "a2a-sdp": [
                      {
                        "sdp-id": "1"
                      },
                      {
                        "sdp-id": "3a"
                      },
                      {
                        "sdp-id": "4"
                      }
                    ]
                  }
                ]
              }
            ]
          }
        },
        {
          "service-id": "slice2",
          "service-description": "example slice2",
          "service-tags": {
            "tag-type": [
              {
                "tag-type": "ietf-nss:service-tag-service",
                "value": [
                  "L3"
                ]
              }
            ]
          },
          "slo-sle-template": "high-BW-template",
          "status": {},
          "sdps": {
            "sdp": [
              {
                "sdp-id": "2",
                "node-id": "PE-A",
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "ac2",
                      "ac-description": "AC2 connected to device 2",
                      "ac-node-id": "PE-A",
                      "ac-tp-id": "GigabitEthernet7/0/0/3.200",
                      "ac-ip-address": "198.51.100.1",
                      "ac-ip-prefix-length": 24,
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "ac-tag-type": "ietf-nss:attachment-circuit-
tag-vlan-id",

                            "value": [
                              "100"
                            ]
                          }
                        ]
                      },
                      "sdp-peering": {
                        "protocol": [
                          {
                            "protocol-type": "ietf-nss:peering-protocol-
bgp",

                            "attribute": [
                              {
                                "attribute-type": "ietf-nss:neighbor",
                                "value": [
                                  "192.51.100.2"
                                ]
                              },
                              {
                                "attribute-type": "ietf-nss:remote-as",
                                "value": [
                                  "64031"
                                ]
                              },
                              {
                                "attribute-type": "ietf-nss:local-as",
                                "value": [
                                  "64000"
                                ]
                              }
                            ]
                          }
                        ]
                      }
                    }
                  ]
                },
                "status": {}
              },
              {
                "sdp-id": "3b",
                "node-id": "PE-B",
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "ac3b",
                      "ac-description": "AC3b connected to device 3",
                      "ac-node-id": "PE-B",
                      "ac-tp-id": "GigabitEthernet8/0/0/4.201",
                      "ac-ip-address": "198.51.101.1",
                      "ac-ip-prefix-length": 24,
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "ac-tag-type": "ietf-nss:attachment-circuit-
tag-vlan-id",

                            "value": [
                              "201"
                            ]
                          }
                        ]
                      },
                      "sdp-peering": {
                        "protocol": [
                          {
                            "protocol-type": "ietf-nss:peering-protocol-
bgp",

                            "attribute": [
                              {
                                "attribute-type": "ietf-nss:neighbor",
                                "value": [
                                  "192.51.101.2"
                                ]
                              },
                              {
                                "attribute-type": "ietf-nss:remote-as",
                                "value": [
                                  "64032"
                                ]
                              },
                              {
                                "attribute-type": "ietf-nss:local-as",
                                "value": [
                                  "64000"
                                ]
                              }
                            ]
                          }
                        ]
                      }
                    }
                  ]
                },
                "status": {}
              }
            ]
          },
          "connection-groups": {
            "connection-group": [
              {
                "connection-group-id": "matrix2",
                "connectivity-type": "ietf-vpn-common:any-to-any",
                "connectivity-construct": [
                  {
                    "cc-id": 1,
                    "a2a-sdp": [
                      {
                        "sdp-id": "2"
                      },
                      {
                        "sdp-id": "3b"
                      }
                    ]
                  }
                ]
              }
            ]
          }
        }
      ]
    }
  }
}
```

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

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

* IETF Network Slice 3 on SDP5 and SDP7a with pt-to-pt connectivity type. This is a L2 slice service and using the uniform low-latency slo-sle-template policies between the SDPs.

* IETF Network Slice 4 on SDP6 and SDP7b, with pt-to-pt connectivity type. This is a L2 slice service and using the a high-BW slo-sle-template policies between the SDPs. Traffic from SDP6 and SDP7b is requesting a BW of 1000Mbps, while in the reverse direction from SDO7b 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.

```
+--------+
|  CE5   o------/  VLAN100
+--------+      |  SDP5 +------+
+--------+      +------o| PE A +---------------+
|  CE6   o-------/-----o|      |               |
+--------+         SDP6 +---+--+               |
                 VLAN200    |                  |
                            |              +---+--+
                            |              |      |
                            |              | PE C o
+--------+                  |              +---+--+
|        o------/ VLAN101   |                  |
|        |      | SDP7a +---+--+               |
| CE7    |      +------o| PE B +---------------+
|        o-------/-----o|      |
+--------+        SDP7b +------+
                  VLAN201
```

```
{
  "data": {
    "ietf-network-slice-service:network-slice-services": {
      "slo-sle-templates": {
        "slo-sle-template": [
          {
            "id": "high-BW-template",
            "template-description": "take the highest BW forwarding path"
          },
          {
            "id": "low-latency-template",
            "template-description": "lowest possible latencey forwarding
behavior"
          }
        ]
      },
      "slice-service": [
        {
          "service-id": "slice3",
          "service-description": "example slice3",
          "slo-sle-template": "low-latency-template",
          "status": {},
          "sdps": {
            "sdp": [
              {
                "sdp-id": "5",
                "node-id": "PE-A",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-network-slice-service:service-
any-match",
                      "target-connection-group-id": "matrix3"
                    }
                  ]
                },
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "ac5",
                      "ac-description": "AC5 connected to device 5",
                      "ac-node-id": "PE-A",
                      "ac-tp-id": "GigabitEthernet5/0/0/1",
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "ac-tag-type": "ietf-nss:attachment-circuit-
tag-vlan-id",
                            "value": [
                              "100"
                            ]
                          }
                        ]
                      }
                    }
                  ]
                },
                "status": {}
              },
              {
                "sdp-id": "7a",
                "node-id": "PE-B",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-network-slice-service:service-
any-match",
                      "target-connection-group-id": "matrix3"
                    }
                  ]
                },
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "ac7a",
                      "ac-description": "AC7a connected to device 7",
                      "ac-node-id": "PE-B",
                      "ac-tp-id": "GigabitEthernet8/0/0/5",
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "ac-tag-type": "ietf-nss:attachment-circuit-
tag-vlan-id",
                            "value": [
                              "200"
                            ]
                          }
                        ]
                      }
                    }
                  ]
                },
                "status": {}
              }
            ]
          },
          "connection-groups": {
            "connection-group": [
              {
                "connection-group-id": "matrix3",
                "connectivity-type": "ietf-network-slice-service:point-
to-point",
                "connectivity-construct": [
                  {
                    "cc-id": 1,
                    "p2p-sender-sdp": "5",
                    "p2p-receiver-sdp": "7a"
                  },
                  {
                    "cc-id": 2,
                    "p2p-sender-sdp": "7a",
                    "p2p-receiver-sdp": "5"
                  }
                ]
              }
            ]
          }
        },
        {
          "service-id": "slice4",
          "service-description": "example slice4",
          "slo-sle-template": "high-BW-template",
          "status": {},
          "sdps": {
            "sdp": [
              {
                "sdp-id": "6",
                "node-id": "PE-A",
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "ac6",
                      "ac-description": "AC6 connected to device 6",
                      "ac-node-id": "PE-A",
                      "ac-tp-id": "GigabitEthernet7/0/0/4",
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "ac-tag-type": "ietf-nss:attachment-circuit-
tag-vlan-id",
                            "value": [
                              "101"
                            ]
                          }
                        ]
                      }
                    }
                  ]
                },
                "status": {}
              },
              {
                "sdp-id": "7b",
                "node-id": "PE-B",
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "ac7b",
                      "ac-description": "AC7b connected to device 7",
                      "ac-node-id": "PE-B",
                      "ac-tp-id": "GigabitEthernet8/0/0/5",
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "ac-tag-type": "ietf-nss:attachment-circuit-
tag-vlan-id",
                            "value": [
                              "201"
                            ]
                          }
                        ]
                      }
                    }
                  ]
                },
                "status": {}
              }
            ]
          },
          "connection-groups": {
            "connection-group": [
              {
                "connection-group-id": "matrix4",
                "connectivity-type": "ietf-network-slice-service:point-
to-point",
                "connectivity-construct": [
                  {
                    "cc-id": 1,
                    "p2p-sender-sdp": "6",
                    "p2p-receiver-sdp": "7b",
                    "service-slo-sle-policy": {
                      "metric-bounds": {
                        "metric-bound": [
                          {
                            "metric-type": "ietf-nss:service-slo-one-way-
bandwidth",
                            "metric-unit": "Mbps",
                            "value-description": "1000"
                          }
                        ]
                      }
                    }
                  },
                  {
                    "cc-id": 2,
                    "p2p-sender-sdp": "7b",
                    "p2p-receiver-sdp": "6",
                    "service-slo-sle-policy": {
                      "metric-bounds": {
                        "metric-bound": [
                          {
                            "metric-type": "ietf-nss:service-slo-one-way-
bandwidth",
                            "metric-unit": "Mbps",
                            "value-description": "5000"
                          }
                        ]
                      }
                    }
                  }
                ]
              }
            ]
          }
        }
      ]
    }
  }
}
```

A.3. Example-3: A Hub-and-spoke Slice Service with a P2MP connectivity construct

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

> IETF Network Slice 5 is a hub-spoke slice with SDP14 as the hub and SDP11, SDP12, SDP13a, SDP13b as spokes. This is a L3 slice service and using the uniform low-latency slo-sle-template policies between all spokes and the hub SDP, but using an explicit set of slo policies with a latency metric of 10ms for hub to spoke traffic. There is no peering protocol's configured in this example.

```
+--------+         196.0.2.1
|Device11o------/  VLAN100
+--------+      |  SDP11+------+
+--------+      +------o|  A   +---------------+
|Device12o-------/-----o|      |               |
+--------+         SDP12+---+--+               |
               196.51.100.1 |                  |    196.0.4.1
                  VLAN200   |              +---+--+ VLAN100
                            |              |      | SDP14     +--------+
                            |              |   C  o-----/-----oDevice14|
+--------+        196.0.3.1 |              +---+--+           +--------+
|        o------/ VLAN101   |                  |
|        |      | SDP13a+---+--+               |
|Device13|      +------o|  B   +---------------+
|        o-------/-----o|      |
+--------+        SDP13b+------+
                  196.51.101.1
                  VLAN201
```

```
{
  "data": {
    "ietf-network-slice-service:network-slice-services": {
      "slo-sle-templates": {
        "slo-sle-template": [
          {
            "id": "high-BW-template",
            "template-description": "take the highest BW forwarding path"
          },
          {
            "id": "low-latency-template",
            "template-description": "lowest possible latencey forwarding
behavior"
          }
        ]
      },
      "slice-service": [
        {
          "service-id": "slice5",
          "service-description": "example slice5",
          "service-tags": {
            "tag-type": [
              {
                "tag-type": "ietf-nss:service-tag-service",
                "value": [
                  "L3"
                ]
              }
            ]
          },
          "slo-sle-template": "low-latency-template",
          "status": {},
          "sdps": {
            "sdp": [
              {
                "sdp-id": "11",
                "node-id": "PE-A",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-network-slice-service:service-
any-match",
                      "target-connection-group-id": "matrix5",
                      "connection-group-sdp-role": "ietf-vpn-
common:spoke-role"
                    }
                  ]
                },
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "ac11",
                      "ac-description": "AC11 connected to device 11",
                      "ac-node-id": "PE-A",
                      "ac-tp-id": "GigabitEthernet5/0/0/2",
                      "ac-ip-address": "196.0.2.1",
                      "ac-ip-prefix-length": 24,
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "ac-tag-type": "ietf-nss:attachment-circuit-
tag-vlan-id",
                            "value": [
                              "100"
                            ]
                          }
                        ]
                      }
                    }
                  ]
                },
                "status": {}
              },
              {
                "sdp-id": "12",
                "node-id": "PE-A",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-network-slice-service:service-
any-match",
                      "target-connection-group-id": "matrix5",
                      "connection-group-sdp-role": "ietf-vpn-
common:spoke-role"
                    }
                  ]
                },
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "ac12",
                      "ac-description": "AC12 connected to device 12",
                      "ac-node-id": "PE-A",
                      "ac-tp-id": "GigabitEthernet7/0/0/5",
                      "ac-ip-address": "196.51.100.1",
                      "ac-ip-prefix-length": 24,
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "ac-tag-type": "ietf-nss:attachment-circuit-
tag-vlan-id",
                            "value": [
                              "200"
                            ]
                          }
                        ]
                      }
                    }
                  ]
                },
                "status": {}
              },
              {
                "sdp-id": "13a",
                "node-id": "PE-B",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-network-slice-service:service-
any-match",
                      "target-connection-group-id": "matrix5",
                      "connection-group-sdp-role": "ietf-vpn-
common:spoke-role"
```

```
                    }
                  ]
                },
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "ac13a",
                      "ac-description": "AC13a connected to device 13",
                      "ac-node-id": "PE-B",
                      "ac-tp-id": "GigabitEthernet8/0/0/6",
                      "ac-ip-address": "196.0.3.1",
                      "ac-ip-prefix-length": 24,
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "ac-tag-type": "ietf-nss:attachment-circuit-
tag-vlan-id",
                            "value": [
                              "101"
                            ]
                          }
                        ]
                      }
                    }
                  ]
                },
                "status": {}
              },
              {
                "sdp-id": "13b",
                "node-id": "PE-B",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-network-slice-service:service-
any-match",
                      "target-connection-group-id": "matrix5",
                      "connection-group-sdp-role": "ietf-vpn-
common:spoke-role"
                    }
                  ]
                },
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "ac13b",
                      "ac-description": "AC3b connected to device 13",
                      "ac-node-id": "PE-B",
                      "ac-tp-id": "GigabitEthernet8/0/0/4",
                      "ac-ip-address": "196.51.101.1",
                      "ac-ip-prefix-length": 24,
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "ac-tag-type": "ietf-nss:attachment-circuit-
tag-vlan-id",
                            "value": [
                              "201"
                            ]
                          }
                        ]
                      }
                    }
                  ]
                },
                "status": {}
              },
              {
                "sdp-id": "14",
                "node-id": "PE-C",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-network-slice-service:service-
any-match",
                      "target-connection-group-id": "matrix5",
                      "connection-group-sdp-role": "ietf-vpn-common:hub-
role"
                    }
                  ]
                },
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "ac14",
                      "ac-description": "AC14 connected to device 14",
                      "ac-node-id": "PE-C",
                      "ac-tp-id": "GigabitEthernet4/0/0/3",
                      "ac-ip-address": "196.0.4.1",
                      "ac-ip-prefix-length": 24,
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "ac-tag-type": "ietf-nss:attachment-circuit-
tag-vlan-id",
                            "value": [
                              "100"
                            ]
                          }
                        ]
                      }
                    }
                  ]
                },
                "status": {}
              }
            ]
          },
          "connection-groups": {
            "connection-group": [
              {
                "connection-group-id": "matrix5",
                "connectivity-type": "ietf-vpn-common:hub-spoke",
                "connectivity-construct": [
                  {
                    "cc-id": 1,
                    "p2mp-sender-sdp": "14",
                    "p2mp-receiver-sdp": [
                      "11",
                      "12",
                      "13a",
                      "13b"
                    ],
                    "service-slo-sle-policy": {
                      "metric-bounds": {
                        "metric-bound": [
                          {
                            "metric-type": "ietf-nss:service-slo-one-way-
delay",
                            "metric-unit": "miliseconds",
                            "value-description": "10"
                          }
                        ]
                      }
                    }
                  },
                  {
                    "cc-id": 2,
                    "p2p-sender-sdp": "11",
                    "p2p-receiver-sdp": "14"
                  },
                  {
                    "cc-id": 3,
                    "p2p-sender-sdp": "12",
                    "p2p-receiver-sdp": "14"
                  },
                  {
                    "cc-id": 4,
                    "p2p-sender-sdp": "13a",
                    "p2p-receiver-sdp": "14"
                  },
                  {
                    "cc-id": 5,
                    "p2p-sender-sdp": "13b",
                    "p2p-receiver-sdp": "14"
                  }
                ]
              }
            ]
          }
        }
      ]
    }
  }
}
```

A.4. Example-4: An Any-to-any Slice service with multiple SLOs and DSCP Matching

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

IETF Network Slice 6 on SDP21, SDP23a, and SDP24, with any-to-any connectivity type. This is a L3 slice service and using the uniform standard slo-sle-template policies between all SDPs. For traffic matching the dscp of EF, a slo-sle-template policy of low-latency will be used. The slice uses the explicit match approach for mapping SDP traffic to a connectivity-construct.

```
+--------+         194.0.2.1
| CE21   o------/  VLAN100
+--------+      |  SDP21+------+
                +------o| PE A +---------------+
                        |      |               |
                        +---+--+               |
                            |                  |    194.0.4.1
                            |              +---+--+ VLAN100
                            |              |      | SDP24     +--------+
                            |              | PE C o-----/-----o CE24   |
+--------+        194.0.3.1 |              +---+--+           +--------+
|        o------/ VLAN101   |                  |
|        |      | SDP23a+---+--+               |
|CE23    |      +------o| PE B +---------------+
|        o              |      |
+--------+              +------+
```

```
{
  "data": {
    "ietf-network-slice-service:network-slice-services": {
      "slo-sle-templates": {
        "slo-sle-template": [
          {
            "id": "high-BW-template",
            "template-description": "take the highest BW forwarding path"
          },
          {
            "id": "low-latency-template",
            "template-description": "lowest possible latencey forwarding
behavior"
          },
          {
            "id": "standard-template",
            "template-description": "take the standard forwarding path"
          }
        ]
      },
      "slice-service": [
        {
          "service-id": "slice6",
          "service-description": "example slice6",
          "service-tags": {
            "tag-type": [
              {
                "tag-type": "ietf-nss:service-tag-service",
                "value": "L3"
              }
            ]
          },
          "slo-sle-template": "standard-template",
          "status": {},
          "sdps": {
            "sdp": [
              {
                "sdp-id": "21",
                "node-id": "PE-A",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-network-slice-service:service-
dscp-match",
                      "value": "EF",
                      "target-connection-group-id": "matrix6",
                      "target-connectivity-construct-id": 2
                    },
                    {
                      "index": 2,
                      "match-type": "ietf-network-slice-service:service-
any-match",
                      "target-connection-group-id": "matrix6",
                      "target-connectivity-construct-id": 1
                    }
                  ]
                },
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "ac21",
                      "ac-description": "AC21 connected to device 21",
                      "ac-node-id": "PE-A",
                      "ac-tp-id": "GigabitEthernet5/0/0/0",
                      "ac-ip-address": "194.0.2.1",
                      "ac-ip-prefix-length": 24,
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "ac-tag-type": "ietf-nss:attachment-circuit-
tag-vlan-id",
                            "value": [
                              "100"
                            ]
                          }
                        ]
                      }
                    }
                  ]
                },
                "status": {}
              },
              {
                "sdp-id": "23a",
                "node-id": "PE-B",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-network-slice-service:service-
dscp-match",
                      "value": "EF",
                      "target-connection-group-id": "matrix6",
                      "target-connectivity-construct-id": 2
                    },
                    {
                      "index": 2,
                      "match-type": "ietf-network-slice-service:service-
any-match",
                      "target-connection-group-id": "matrix6",
                      "target-connectivity-construct-id": 1
                    }
                  ]
                },
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "ac23a",
                      "ac-description": "AC23a connected to device 23",
                      "ac-node-id": "PE-B",
                      "ac-tp-id": "GigabitEthernet8/0/0/4",
                      "ac-ip-address": "194.0.3.1",
                      "ac-ip-prefix-length": 24,
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "ac-tag-type": "ietf-nss:attachment-circuit-
tag-vlan-id",
                            "value": [
                              "101"
                            ]
                          }
                        ]
                      }
                    }
                  ]
                },
                "status": {}
              },
              {
                "sdp-id": "24",
                "node-id": "PE-C",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-network-slice-service:service-
dscp-match",
                      "value": "EF",
                      "target-connection-group-id": "matrix6",
                      "target-connectivity-construct-id": 2
                    },
                    {
                      "index": 2,
                      "match-type": "ietf-network-slice-service:service-
any-match",
                      "target-connection-group-id": "matrix6",
                      "target-connectivity-construct-id": 1
                    }
                  ]
                },
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "ac24",
                      "ac-description": "AC24 connected to device 24",
                      "ac-node-id": "PE-C",
                      "ac-tp-id": "GigabitEthernet4/0/0/3",
                      "ac-ip-address": "194.0.4.1",
                      "ac-ip-prefix-length": 24,
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "ac-tag-type": "ietf-nss:attachment-circuit-
tag-vlan-id",

                            "value": [
                              "100"
                            ]
                          }
                        ]
                      }
                    }
                  ]
                },
                "status": {}
              }
            ]
          },
          "connection-groups": {
            "connection-group": [
              {
                "connection-group-id": "matrix6",
                "connectivity-type": "ietf-vpn-common:any-to-any",
                "connectivity-construct": [
                  {
                    "cc-id": 1,
                    "a2a-sdp": [
                      {
                        "sdp-id": "21"
                      },
                      {
                        "sdp-id": "23a"
                      },
                      {
                        "sdp-id": "24"
                      }
                    ]
                  },
                  {
                    "cc-id": 2,
                    "a2a-sdp": [
                      {
                        "sdp-id": "21"
                      },
                      {
                        "sdp-id": "23a"
                      },
                      {
                        "sdp-id": "24"
                      }
                    ],
                    "slo-sle-template": "low-latency-template"
                  }
                ]
              }
            ]
          }
        }
      ]
    }
  }
}
```

A.5. Example-5: An any-to-any Network Slice Service with SLO Precedence Policies

The following examples describes a simplified service configuration of an IETF Network slice instance 'NS1' with four SDPs: SDP1, SDP2, SDP3 and SDP4 with any-to-any connectivity type. All SDPs are designated as customer-facing ports on the PE.

The service is realized using a single any-to-any connectivity construct, and a uniform low-bandwidth slo-sle-template policy applied to SDP4, SDP3, while a high-bandwidth slo-sle-template policy applied to SDP1 and SDP2.

```
+---------+        192.0.1.1                        192.0.3.1
|CE1     o------/  VLAN100                          VLAN100
+---------+     |  SDP1 +------+           +------+ SDP3
                +------o| PE A +-----------| PE C |           +---------+
                        |      |           |      |-----/-----o CE3     |
                        +---+--+           +------+           +---------+
                            |                  |
                            |                  |
                            |                  |
                            |                  |
+---------+       192.0.2.1 |                  |
|CE2     o------/ VLAN100   |                  |    192.0.4.1
+---------+     | SDP2  +---+--+           +---+--+ VLAN100
                +------o| PE B +-----------|PE D  | SDP4      +---------+
                        |      |           |      o-----/-----o     CE4 |
                        +------+           +---+--+           +---------+
```

```
{
  "data": {
    "ietf-network-slice-service:network-slice-services": {
      "slo-sle-templates": {
        "slo-sle-template": [
          {
            "id": "high-BW-template",
            "template-description": "take the highest BW forwarding path"
          },
          {
            "id": "low-BW-template",
            "template-description": "lowest BW forwarding behavior"
          }
        ]
      },
      "slice-service": [
        {
          "service-id": "NS1",
          "service-description": "URLLC",
          "service-tags": {
            "tag-type": [
              {
                "tag-type": "ietf-nss:service-tag-customer",
                "value": [
                  "Customer-FOO"
                ]
              },
              {
                "tag-type": "ietf-nss:service-tag-service",
                "value": [
                  "L3"
                ]
              }
            ]
          },
          "status": {},
          "sdps": {
            "sdp": [
              {
                "sdp-id": "SDP1",
                "sdp-description": "Central Office 1 at location PE-A",
                "node-id": "PE-A",
                "sdp-ip": "192.0.1.1",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-network-slice-service:service-
vlan-match",
                      "value": [
                        "100"
                      ],
                      "target-connection-group-id": "matrix1"
                    }
                  ]
                },
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "AC-SDP1",
                      "ac-description": "Device 1 to PE-A",
                      "ac-node-id": "PE-A",
                      "ac-tp-id": "GigabitEthernet1/0/0/0",
                      "ac-ip-address": "192.0.1.1",
                      "ac-ip-prefix-length": 24,
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "ac-tag-type": "ietf-nss:attachment-circuit-
tag-vlan-id",
                            "value": [
                              "100"
                            ]
                          }
                        ],
                        "ac-tag-opaque": [
                          {
                            "tag-name": "VRF",
                            "value": [
                              "FOO"
                            ]
                          }
                        ]
                      },
                      "sdp-peering": {
                        "protocol": [
                          {
                            "protocol-type": "ietf-nss:peering-protocol-
bgp",
                            "attribute": [
                              {
                                "attribute-type": "ietf-nss:neighbor",
                                "value": [
                                  "192.0.1.2"
                                ]
                              }
                            ]
                          }
                        ],
                        "opaque": [
                          {
                            "attribute-name": "color",
                            "value": [
                              "10"
                            ]
                          }
                        ]
                      },
                      "incoming-qos-policy": {
                        "qos-policy-name": "QoS-Gold",
                        "rate-limits": {
                          "cir": "1000000",
                          "cbs": "1000",
                          "pir": "5000000",
                          "pbs": "1000"
                        }
                      }
                    }
                  ]
                },
                "status": {}
              },
              {
                "sdp-id": "SDP2",
                "sdp-description": "Central Office 2 at location PE-B",
                "node-id": "PE-B",
                "sdp-ip": "192.0.2.1",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-network-slice-service:service-
vlan-match",
                      "value": [
                        "100"
                      ],
                      "target-connection-group-id": "matrix1"
                    }
                  ]
                },
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "AC-SDP2",
                      "ac-description": "Device 2 to PE-B",
                      "ac-node-id": "PE-B",
                      "ac-tp-id": "GigabitEthernet2/0/0/0",
                      "ac-ip-address": "192.0.2.1",
                      "ac-ip-prefix-length": 24,
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "ac-tag-type": "ietf-nss:attachment-circuit-
tag-vlan-id",
                            "value": [
                              "100"
                            ]
                          }
                        ],
                        "ac-tag-opaque": [
                          {
                            "tag-name": "VRF",
                            "value": [
                              "FOO"
                            ]
                          }
                        ]
                      },
                      "sdp-peering": {
                        "protocol": [
                          {
                            "protocol-type": "ietf-nss:peering-protocol-
bgp",
                            "attribute": [
                              {
                                "attribute-type": "ietf-nss:neighbor",
                                "value": [
                                  "192.0.2.2"
                                ]
                              }
                            ]
                          }
                        ],
                        "opaque": [
                          {
                            "attribute-name": "color",
                            "value": [
                              "10"
                            ]
                          }
                        ]
                      },
                      "incoming-qos-policy": {
                        "qos-policy-name": "QoS-Gold",
                        "rate-limits": {
                          "cir": "1000000",
                          "cbs": "1000",
                          "pir": "5000000",
                          "pbs": "1000"
                        }
                      }
                    }
                  ]
                },
                "status": {}
              },
              {
                "sdp-id": "SDP3",
                "sdp-description": "Remote Office 1 at location PE-C",
                "node-id": "PE-C",
                "sdp-ip": "192.0.3.1",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-network-slice-service:service-
vlan-match",
                      "value": [
                        "100"
                      ],
                      "target-connection-group-id": "matrix1"
                    }
                  ]
                },
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "AC-SDP3",
                      "ac-description": "Device 3 to PE-C",
                      "ac-node-id": "PE-C",
                      "ac-tp-id": "GigabitEthernet3/0/0/0",
                      "ac-ip-address": "192.0.3.1",
                      "ac-ip-prefix-length": 24,
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "ac-tag-type": "ietf-nss:attachment-circuit-
tag-vlan-id",
                            "value": [
                              "100"
                            ]
                          }
                        ],
                        "ac-tag-opaque": [
                          {
                            "tag-name": "VRF",
                            "value": [
                              "FOO"
                            ]
                          }
                        ]
                      },
                      "sdp-peering": {
                        "protocol": [
                          {
                            "protocol-type": "ietf-nss:peering-protocol-
bgp",
                            "attribute": [
                              {
                                "attribute-type": "ietf-nss:neighbor",
                                "value": [
                                  "192.0.3.2"
                                ]
                              }
                            ]
                          }
                        ],
                        "opaque": [
                          {
                            "attribute-name": "color",
                            "value": [
                              "10"
                            ]
                          }
                        ]
                      },
                      "incoming-qos-policy": {
                        "qos-policy-name": "QoS-Gold",
                        "rate-limits": {
                          "cir": "1000000",
                          "cbs": "1000",
                          "pir": "5000000",
                          "pbs": "1000"
                        }
                      }
                    }
                  ]
                },
                "status": {}
              },
              {
                "sdp-id": "SDP4",
                "sdp-description": "Remote Office 2 at location PE-D",
                "node-id": "PE-D",
                "sdp-ip": "192.0.4.1",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-network-slice-service:service-
vlan-match",
                      "value": [
                        "100"
                      ],
                      "target-connection-group-id": "matrix1"
                    }
                  ]
                },
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "AC-SDP4",
                      "ac-description": "Device 4 to PE-D",
                      "ac-node-id": "PE-A",
                      "ac-tp-id": "GigabitEthernet4/0/0/0",
                      "ac-ip-address": "192.0.4.1",
                      "ac-ip-prefix-length": 24,
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "ac-tag-type": "ietf-nss:attachment-circuit-
tag-vlan-id",
                            "value": [
                              "100"
                            ]
                          }
                        ],
                        "ac-tag-opaque": [
                          {
                            "tag-name": "VRF",
                            "value": [
                              "FOO"
                            ]
                          }
                        ]
                      },
                      "sdp-peering": {
                        "protocol": [
                          {
                            "protocol-type": "ietf-nss:peering-protocol-
bgp",
                            "attribute": [
                              {
                                "attribute-type": "ietf-nss:neighbor",
                                "value": [
                                  "192.0.4.2"
                                ]
                              }
                            ]
                          }
                        ],
                        "opaque": [
                          {
                            "attribute-name": "color",
                            "value": [
                              "10"
                            ]
                          }
                        ]
                      },
                      "incoming-qos-policy": {
                        "qos-policy-name": "QoS-Gold",
                        "rate-limits": {
                          "cir": "1000000",
                          "cbs": "1000",
                          "pir": "5000000",
                          "pbs": "1000"
                        }
                      }
                    }
                  ]
                },
                "status": {}
              }
            ]
          },
          "connection-groups": {
            "connection-group": [
              {
                "connection-group-id": "matrix1",
                "slo-sle-template": "low-BW-template",
                "connectivity-construct": [
                  {
                    "cc-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"
                      }
                    ]
                  }
                ]
              }
            ]
          }
        }
      ]
    }
  }
}
```

A.6. Example-6: SDP at CE, L3 any-to-any Slice Service

The following example describes a simplified service configuration of one IETF Network slice instances where the SDPs are located at the

PE-facing ports on the CE:

* IETF Network Slice 7 with SDP31 on CE Device1, SDP33 (with two ACs) on Device 3 and SDP34 on Device 4, with an any-to-any connectivity type. This is a L3 slice service and using the uniform low-latency slo-sle-template policy between all SDPs. These SDPs will also have AC eBGP peering sessions with the corresponding PE elements.

* 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.

  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

```
SDP31
SDP-ip 10.1.1.1
(Loopback)
    |
    |     192.0.2.2
    v      VLAN200      +------+
+--------+ ac31         | PE A +---------------+
|  CE1   o-------/-----o|      |               |                SDP34
+--------+              +---+--+               |       SDP-ip 10.1.1.4
                            |                  |                 |
SDP33                                                            |
SDP-ip 101.1.3              |              +---+--+              v
    |      192.0.3.2        |              |      |           +--------+
    v      VLAN101          |              | PE C o-----/-----o CE2    |
+--------+ ac33a            |              +---+--+    ac34   +--------+
|        o------/           |                  |       VLAN201
|        |      |       +---+---+              |       198.51.101.2
|  CE3   |      +------o| PE B  +--------------+
|        o-------/-----o|       |
+--------+ ac33b        +-------+
           VLAN201
           198.51.101.2
```

```
{
  "data": {
    "ietf-network-slice-service:network-slice-services": {
      "slo-sle-templates": {
        "slo-sle-template": [
          {
            "id": "high-BW-template",
            "template-description": "take the highest BW forwarding path"
          },
          {
            "id": "low-latency-template",
            "template-description": "lowest possible ~~latencey~~latency
forwarding behavior"
          }
        ]
      },
      "slice-service": [
        {
          "service-id": "slice7",
          "service-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": [
              {
                "sdp-id": "31",
                "node-id": "Device-1",
                "sdp-ip": "10.1.1.1",
                "peer-sap-id": "PEA-SAP-NS-1",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-network-slice-service:service-
any-match",
                      "target-connection-group-id": "matrix1",
                      "target-connectivity-construct-id": 1
                    }
                  ]
                },
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "ac31",
                      "ac-description": "AC1 connected to PE-A",
                      "ac-node-id": "Device-1",
                      "ac-tp-id": "GigabitEthernet0",
                      "ac-ip-address": "192.0.2.2",
                      "ac-ip-prefix-length": 24,
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "ac-tag-type": "ietf-nss:attachment-circuit-
tag-vlan-id",
                            "value": [
                              "100"
                            ]
                          }
                        ]
                      },
                      "sdp-peering": {
                        "protocol": [
                          {
                            "protocol-type": "ietf-nss:peering-protocol-
bgp",
                            "attribute": [
                              {
                                "attribute-type": "ietf-nss:neighbor",
                                "value": [
                                  "192.0.2.1"
                                ]
                              },
                              {
                                "attribute-type": "ietf-nss:remote-as",
                                "value": [
                                  "64000"
                                ]
                              },
                              {
                                "attribute-type": "ietf-nss:local-as",
                                "value": [
                                  "64001"
                                ]
                              }
                            ]
                          }
                        ]
                      }
                    }
                  ]
                },
                "status": {}
              },
              {
                "sdp-id": "33",
                "node-id": "Device-3",
                "sdp-ip": "10.1.1.3",
                "peer-sap-id": "PEB-SAP-NS-1",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-network-slice-service:service-
any-match",
                      "target-connection-group-id": "matrix1",
                      "target-connectivity-construct-id": 1
                    }
                  ]
                },
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "ac33a",
                      "ac-description": "AC33a connected to PE-B",
                      "ac-node-id": "Device-3",
                      "ac-tp-id": "GigabitEthernet0",
                      "ac-ip-address": "192.0.3.2",
                      "ac-ip-prefix-length": 24,
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "ac-tag-type": "ietf-nss:attachment-circuit-
tag-vlan-id",
                            "value": [
                              "101"
                            ]
                          }
                        ]
                      },
                      "sdp-peering": {
                        "protocol": [
                          {
                            "protocol-type": "ietf-nss:peering-protocol-
bgp",
                            "attribute": [
                              {
                                "attribute-type": "ietf-nss:neighbor",
                                "value": [
                                  "192.0.3.1"
                                ]
                              },
                              {
                                "attribute-type": "ietf-nss:remote-as",
                                "value": [
                                  "64000"
                                ]
                              },
                              {
                                "attribute-type": "ietf-nss:local-as",
                                "value": [
                                  "64002"
                                ]
                              }
                            ]
                          }
                        ]
                      }
                    },
                    {
                      "ac-id": "ac33b",
                      "ac-description": "AC33b connected to PE-B",
                      "ac-node-id": "Device-3",
                      "ac-tp-id": "GigabitEthernet1",
                      "ac-ip-address": "198.51.101.2",
                      "ac-ip-prefix-length": 24,
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "ac-tag-type": "ietf-nss:attachment-circuit-
tag-vlan-id",
                            "value": [
                              "201"
                            ]
                          }
                        ]
                      },
                      "sdp-peering": {
                        "protocol": [
                          {
                            "protocol-type": "ietf-nss:peering-protocol-
bgp",
                            "attribute": [
                              {
                                "attribute-type": "ietf-nss:neighbor",
                                "value": [
                                  "192.51.101.1"
                                ]
                              },
                              {
                                "attribute-type": "ietf-nss:remote-as",
                                "value": [
                                  "64000"
                                ]
                              },
                              {
                                "attribute-type": "ietf-nss:local-as",
                                "value": [
                                  "64002"
                                ]
                              }
                            ]
                          }
                        ]
                      }
                    }
                  ]
                },
                "status": {}
              },
              {
                "sdp-id": "34",
                "node-id": "CE2",
                "sdp-ip": "10.1.1.4",
                "peer-sap-id": "PEC-SAP-NS-1",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-network-slice-service:service-
any-match",
                      "target-connection-group-id": "matrix1",
                      "target-connectivity-construct-id": 1
                    }
                  ]
                },
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "ac34",
                      "ac-description": "AC34 connected to PE-C",
                      "ac-node-id": "Device-4",
                      "ac-tp-id": "GigabitEthernet3",
                      "ac-ip-address": "192.0.4.2",
                      "ac-ip-prefix-length": 24,
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "ac-tag-type": "ietf-nss:attachment-circuit-
tag-vlan-id",
                            "value": [
                              "100"
                            ]
                          }
                        ]
                      },
                      "sdp-peering": {
                        "protocol": [
                          {
                            "protocol-type": "ietf-nss:peering-protocol-
bgp",
                            "attribute": [
                              {
                                "attribute-type": "ietf-nss:neighbor",
                                "value": [
                                  "192.0.4.1"
                                ]
                              },
                              {
                                "attribute-type": "ietf-nss:remote-as",
                                "value": [
                                  "64000"
                                ]
                              },
                              {
                                "attribute-type": "ietf-nss:local-as",
                                "value": [
                                  "64004"
                                ]
                              }
                            ]
                          }
                        ]
                      }
                    }
                  ]
                },
                "status": {}
              }
            ]
          },
          "connection-groups": {
            "connection-group": [
              {
                "connection-group-id": "matrix1",
                "connectivity-type": "ietf-vpn-common:any-to-any",
                "connectivity-construct": [
                  {
                    "cc-id": 1,
                    "a2a-sdp": [
                      {
                        "sdp-id": "31"
                      },
                      {
                        "sdp-id": "33"
                      },
                      {
                        "sdp-id": "34"
                      }
                    ]
                  }
                ]
              }
            ]
          }
        }
      ]
    }
  }
}
```

A.7. Example-7: SDP at CE, L3 any-to-any Slice Service with Network abstraction

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

In this example it is assumed that the NSC has either internal circuit binding details between the CE and PE already configured or has mechanisms to determine this. The NSC can use various mapping mechanisms depending on the capabilities of the NSC including which parameters the NSC considers as unique (this mapping is out-of-scope of this document) such as using the CE device name, sdp-id, sdp-ip, ac-id or the peer-sap-id to complete this circuit binding.

* IETF Network Slice 9 with SDP31 on CPE Device1, SDP33 (with two ACs) on Device 3 and SDP34 on Device 4, with an any-to-any connectivity type. This is a L3 slice service and using the uniform low-latency slo-sle-template policy between all SDPs. These SDPs will also have AC eBGP peering sessions with the corresponding PE elements.

* This example also includes 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.

  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

SDP31
SDP-ip 10.1.1.1
(Loopback)
    |
    |
    v                   +------+
+--------+ ac31         | PE A +---------------+
|  CE1   o-------/-----o|      |               |                SDP34
+--------+              +---+--+               |       SDP-ip 10.1.1.4
                            |                  |                 |
SDP32                                                            |
SDP-ip 101.1.3              |              +---+--+              v
    |                       |              |      |           +--------+
    v                       |              | PE C o-----/-----o CE2    |
+--------+ ac33a            |              +---+--+    ac34   +--------+
|        o------/           |                  |
|        |      |       +---+---+              |
|  CE3   |      +------o| PE B  +--------------+
|        o-------/-----o|       |
+--------+ ac33b        +-------+
```

```
{
  "data": {
    "ietf-network-slice-service:network-slice-services": {
      "slo-sle-templates": {
        "slo-sle-template": [
          {
            "id": "high-BW-template",
            "template-description": "take the highest BW forwarding path"
          },
          {
            "id": "low-BW-template",
            "template-description": "lowest BW forwarding behavior"
          },
          {
            "id": "low-latency-template",
            "template-description": "lowest possible latencey forwarding
behavior"
          }
        ]
      },
      "slice-service": [
        {
          "service-id": "slice9",
          "service-description": "example slice9",
          "service-tags": {
            "tag-type": [
              {
                "tag-type": "ietf-nss:service-tag-service",
                "value": ["L3"]
              }
            ]
          },
          "slo-sle-template": "low-latency-template",
          "status": {
          },
          "sdps": {
            "sdp": [
              {
                "sdp-id": "31",
                "node-id": "Device-1",
                "sdp-ip": "10.1.1.1",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-network-slice-service:service-
any-match",
                      "target-connection-group-id": "matrix1"
                    }
                  ]
                },
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "ac31",
                      "peer-sap-id": "foo.com-circuitID-12345"
                    }
                  ]
                },
                "status": {
                }
              },
              {
                "sdp-id": "33",
                "node-id": "Device-3",
                "sdp-ip": "10.1.1.3",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-network-slice-service:service-
any-match",
                      "target-connection-group-id": "matrix1",
                      "target-connectivity-construct-id": 1
                    }
                  ]
                },
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "ac33a",
                      "peer-sap-id": "foo.com-circuitID-67890"
                    },
                    {
                      "ac-id": "ac33b",
                      "peer-sap-id": "foo.com-circuitID-54321ABC"
                    }
                  ]
                },
                "status": {
                }
              },
              {
                "sdp-id": "34",
                "node-id": "Device-4",
                "sdp-ip": "10.1.1.4",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-network-slice-service:service-
any-match",
                      "target-connection-group-id": "matrix1"
                    }
                  ]
                },
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "ac-id": "ac34",
                      "peer-sap-id": "foo.com-circuitID-9876"
                    }
                  ]
                },
                "status": {
                }
              }
            ]
          },
          "connection-groups": {
            "connection-group": [
              {
                "connection-group-id": "matrix1",
                "connectivity-type": "ietf-vpn-common:any-to-any",
                "connectivity-construct": [
                  {
                    "cc-id": 1,
                    "a2a-sdp": [
                      {
                        "sdp-id": "31"
                      },
                      {
                        "sdp-id": "33"
                      },
                      {
                        "sdp-id": "34"
                      }
                    ]
                  }
                ]
              }
            ]
          }
        }
      ]
    }
  }
}
```

Appendix B. Comparison with Other Possible Design choices for IETF Network Slice Service Interface

According to the 5.3.1 IETF Network Slice Service Interface [I-D.ietf-teas-ietf-network-slices], the Network Slice service Interface is a technology-agnostic interface, which is used for a customer to express requirements for a particular IETF Network Slice. Customers operate on abstract IETF Network Slices, with details related to their realization hidden. As classified by [RFC8309], the Network Slice service Interface is classified as Customer Service Model.

This draft analyzes the following existing IETF models to identify the gap between the IETF Network Slice service Interface requirements.

B.1. ACTN VN Model Augmentation

The difference between the ACTN VN model and the IETF Network Slice service requirements is that the IETF Network Slice service interface is a technology-agnostic interface, whereas the VN model is bound to the IETF TE Topologies. The realization of the IETF Network Slice does not necessarily require the slice network to support the TE technology.

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

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

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

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

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

The Type 1 VN can be used to describe IETF Network Slice connection requirements. However, the Network Slice SLO and Network Slice SDP are not clearly defined and there's no direct equivalent. For example, the SLO requirement of the VN is defined through the IETF TE Topologies YANG model, but the TE Topologies model is related to a specific implementation technology. Also, VN-AP does not define "service-match-criteria" to specify a specific SDP belonging to an IETF Network Slice service.

B.2. RFC8345 Augmentation Model

The difference between the IETF Network Slice service requirements and the IETF basic network model is that the IETF Network Slice service requests abstract customer IETF Network Slices, with details related to the slice Network hidden. But the IETF network model is used to describe the interconnection details of a Network. The customer service model does not need to provide details on the Network.

For example, IETF Network Topologies YANG data model extension introduced in Transport Network Slice YANG Data Model [I-D.liu-teas-transport-network-slice-yang] includes three major parts:

* Network: a transport network list and an list of nodes contained in the network

* Link: "links" list and "termination points" list describe how nodes in a network are connected to each other

* Support network: vertical layering relationships between IETF Network Slice networks and underlay networks

Based on this structure, the IETF Network Slice-specific SLO attributes nodes are augmented on the Network Topologies model,, e.g. isolation etc. However, this modeling design requires the slice network to expose a lot of details of the network, such as the actual topology including nodes interconnection and different network layers interconnection.

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

Liuyan Han
China Mobile
Email: hanliuyan@chinamobile.com

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