

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 10, 2022

B. Wu
D. Dhody
Huawei Technologies
R. Rokui
Nokia
T. Saad
Juniper Networks
L. Han
China Mobile
July 9, 2021

**A Yang Data Model for IETF Network Slice NBI
draft-wd-teas-ietf-network-slice-nbi-yang-03**

Abstract

This document provides a YANG data model for the IETF Network Slice Controller (NSC) Northbound Interface (NBI). The model can be used by a IETF Network Slice customer to request configuration, and management IETF Network Slice services from the IETF NSC.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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

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

This Internet-Draft will expire on January 10, 2022.

Copyright Notice

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

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Conventions used in this document	3
2.1.	Tree Diagrams	4
3.	IETF Network Slice NBI Model Usage	4
4.	IETF Network Slice NBI Model Overview	5
5.	IETF Network Slice Templates	9
6.	IETF Network Slice Modeling Description	10
6.1.	IETF Network Slice Connectivity Type	11
6.2.	IETF Network Slice SLO and SLE Policy	11
6.3.	IETF Network Slice Endpoint (NSE)	13
7.	IETF Network Slice Monitoring	16
8.	IETF Network Slice NBI Module	17
9.	Security Considerations	35
10.	IANA Considerations	36
11.	Acknowledgments	36
12.	References	37
12.1.	Normative References	37
12.2.	Informative References	38
Appendix A.	IETF Network Slice NBI Model Usage Example	39
Appendix B.	Comparison with Other Possible Design choices for IETF Network Slice NBI	42
B.1.	ACTN VN Model Augmentation	42
B.2.	RFC8345 Augmentation Model	43
Appendix C.	Appendix B IETF Network Slice Match Criteria	43
	Authors' Addresses	45

[1.](#) Introduction

This document provides a YANG [[RFC7950](#)] data model for the IETF Network Slice NBI.

The YANG model discussed in this document is defined based on the description of the IETF Network Slice in [[I-D.ietf-teas-ietf-network-slices](#)], which is used to operate IETF Network Slice during the IETF Network Slice instantiation. This YANG model supports various operations on IETF Network Slices such as creation, modification, deletion, and monitoring of IETF Network Slices.

The IETF Network Slice Controller (NSC) provides a Northbound Interface (NBI) that allows customers of network slices to request and monitor IETF network slices.

The NBI carries information that the IETF network slice customer provides, describing generic requirements of connectivity, service level objectives (SLO), etc. and also monitoring and reporting requirements that may apply. It is an abstract interface that hides excessive technology-related information which may then be realized using some technology-specific Southbound Interface (SBI) by the NSC.

The YANG model discussed in this document describes the requirements of an IETF Network Slice from the point of view of the customer, which is classified as Customer Service Model in [\[RFC8309\]](#).

It will be up to the management system or NSC to take this model as an input and use other management system or specific configuration models to configure the different network elements to deliver an IETF Network Slice. The YANG models can be used with network management protocols such as NETCONF [\[RFC6241\]](#) or RESTCONF [\[RFC8040\]](#). The details of how the IETF network slices are realized by the NSC is out of scope for this document.

The IETF Network Slice operational state is included in the same tree as the configuration consistent with Network Management Datastore Architecture [\[RFC8342\]](#).

2. Conventions used in this document

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

The following terms are defined in [\[RFC6241\]](#) and are used in this specification:

- o client
- o configuration data
- o state data

This document makes use of the following terminology introduced in the YANG 1.1 Data Modeling Language [\[RFC7950\]](#):

- o augment

- o data model
- o data node

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

- o NBI: Northbound Interface
- o NS: IETF Network Slice
- o NSC: IETF Network Slice Controller
- o NSE: Network Slice Endpoint
- o SLO: Service Level Objective
- o SLE: Service Level Expectation

This document defines the following term:

- o IETF Network Slice Connection (NS-Connection): In the context of an IETF Network Slice, an IETF NS-Connection is an abstract entity which represents a particular connection between a pair of NSEs. An IETF Network Slice can has one or multiple NS-Connections.

[2.1.](#) Tree Diagrams

Tree diagrams used in this document follow the notation defined in [[RFC8340](#)].

[3.](#) IETF Network Slice NBI Model Usage

The intention of the IETF Network Slice NBI model is to allow the customer, e.g. a higher-level management system, to request and monitor IETF Network Slices. In particular, the model allows customers to operate on abstract and technology-agnostic manner, with details of the IETF Network Slices realization hidden.

According to the [[I-D.ietf-teas-ietf-network-slices](#)] description, the NBI model is applicable to use cases such as (but not limited to) network wholesale services, network infrastructure sharing among operators, NFV connectivity, Data Center Interconnect, and 5G E2E network slice.

As shown in Figure 1, in all these use-cases, the NBI model is used by the higher management system to communicate with IETF Network Slice controller for life cycle manage of IETF Network Slices

including both enablement and monitoring. For example, in 5G E2E network slicing use-case the E2E network slice orchestrator acts as the higher layer system to request the IETF Network Slices. The interface is used to support dynamic IETF Network Slice creation and its lifecycle management to facilitate end-to-end network slice services.

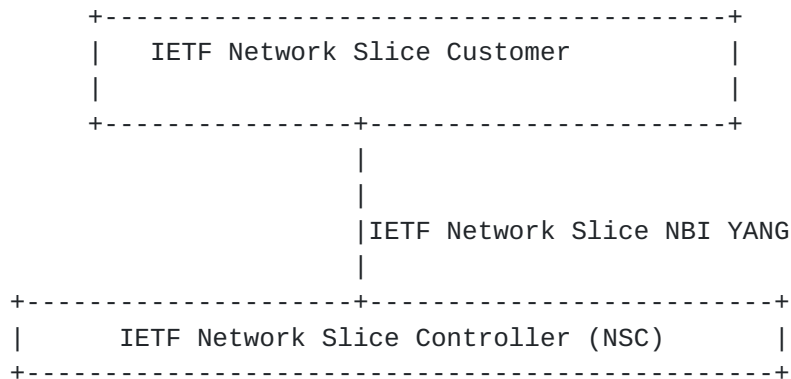
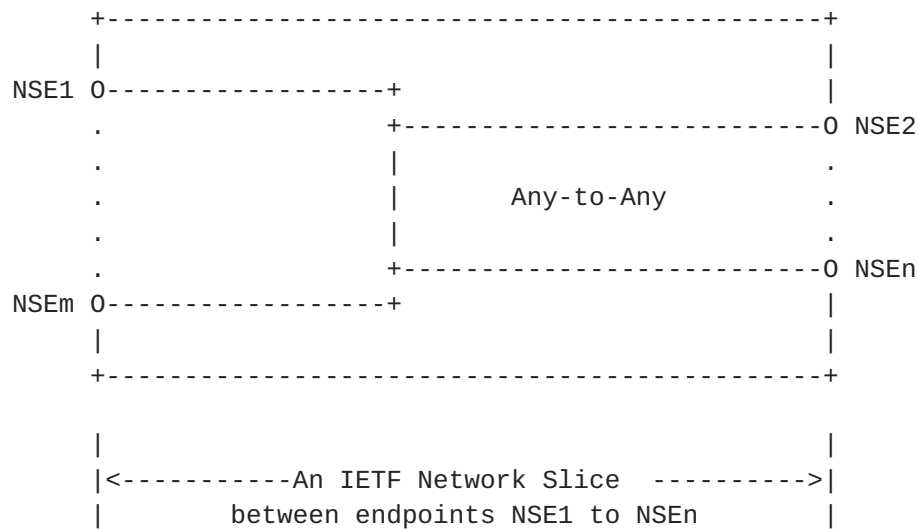


Figure 1: IETF Network Slice NBI Model Context

4. IETF Network Slice NBI Model Overview

As defined in [[I-D.ietf-teas-ietf-network-slices](#)], an IETF network slice is a logical network connecting a number of endpoints with specified SLOs. The connectivity type can be Hub-and-Spoke, any-to-any, or custom connectivity type. In addition, a minimum set of SLOs is defined, including but not limited to bandwidth, latency, and etc. An example of an IETF network slice is shown in Figure 2 .



[I-D.ietf-teas-ietf-network-slices] introduces the IETF network slice endpoints (NSEs) which are conceptual points of connection to IETF network slice. As such, they are ingress/egress point where the traffic enters/exits the IETF network slice. In other words, they are the edge of the IETF network slices.

When IETF network slice controller (NSC) receives a message via its NBI for creation/modification of an IETF network slice, it uses the provided IETF network slice endpoints to map them to appropriate services/tunnels/paths endpoints in the underlay IETF network. It then uses services/tunnels/paths endpoints to realize the IETF network slice.

The IETF Network Slice ("ietf-network-slice") is defined to manage network slices in the IETF network. In particular, the 'ietf-network-slice' module can be used to create, modify, and monitor network slices of an IETF network.

The 'ietf-network-slice' module uses two main nodes: list 'ietf-network-slice' and container 'ns-templates' (see Figure 3).

The 'ietf-network-slice' list includes the set of IETF Network slices managed within IETF network. 'ietf-network-slice' is the data structure that abstracts an IETF Network Slice. Under the "ietf-network-slice", list "ns-endpoint" is used to abstract the NSEs, e.g. NSEs in the example above.

The 'ns-templates' container is used by the NSC to maintain a set of common network slice templates that apply to one or several IETF Network Slices.

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

```

module: ietf-network-slice
  +--rw network-slices
    +--rw ns-slo-sle-templates
      | +--rw ns-slo-sle-template* [id]
      |   +--rw id string
      |   +--rw template-description? string
    +--rw network-slice* [ns-id]
      +--rw ns-id string
      +--rw ns-description? string
      +--rw ns-tag* string
      +--rw ns-connectivity-type? identityref
      +--rw (ns-slo-sle-policy)?
        | +--:(standard)
        | | +--rw slo-sle-template? leafref
        | +--:(custom)
        |   +--rw slo-policy
        |     | +--rw policy-description? string
        |     | +--rw ns-metric-bounds
        |     |   +--rw ns-metric-bound* [metric-type]
        |     |     +--rw metric-type identityref
        |     |     +--rw metric-unit string
        |     |     +--rw value-description? string
        |     |     +--rw bound? uint64
        |   +--rw sle-policies
        |     +--rw security-sle* identityref
        |     +--rw isolation? identityref
        |     +--rw max-occupancy-level? uint8
      +--rw status
        | +--rw admin-enabled? boolean
        | +--ro oper-status? operational-type
      +--rw ns-endpoints
        | +--rw ns-endpoint* [ep-id]
        |   +--rw ep-id string
        |   +--rw ep-description? string
        |   +--rw ep-role? identityref
        |   +--rw location
        |     | +--rw altitude? int64
        |     | +--rw latitude? decimal64
        |     | +--rw longitude? decimal64
        |   +--rw node-id? string
        |   +--rw ep-ip? inet:host
        |   +--rw ns-match-criteria

```



```

|   |   +--rw ns-match-criterion* [match-type]
|   |   |   +--rw match-type      identityref
|   |   |   +--rw values* [index]
|   |   |       +--rw index      uint8
|   |   |       +--rw value?     string
|   +--rw ep-network-access-points
|   |   +--rw ep-network-access-point* [network-access-id]
|   |   |   +--rw network-access-id      string
|   |   |   +--rw network-access-description? string
|   |   |   +--rw network-access-node-id? string
|   |   |   +--rw network-access-tp-id?   string
|   |   |   +--rw network-access-tp-ip?   inet:host
|   |   |   +--rw ep-rate-limit
|   |   |       +--rw incoming-rate-limit?
|   |   |       |       te-types:te-bandwidth
|   |   |       +--rw outgoing-rate-limit?
|   |   |       |       te-types:te-bandwidth
|   +--rw ep-rate-limit
|   |   +--rw incoming-rate-limit? te-types:te-bandwidth
|   |   +--rw outgoing-rate-limit? te-types:te-bandwidth
|   +--rw ep-protocol
|   +--rw status
|   |   +--rw admin-enabled?   boolean
|   |   +--ro oper-status?     operational-type
|   +--ro ep-monitoring
|   |   +--ro incoming-utilized-bandwidth?
|   |   |       te-types:te-bandwidth
|   |   +--ro incoming-bw-utilization      decimal64
|   |   +--ro outgoing-utilized-bandwidth?
|   |   |       te-types:te-bandwidth
|   |   +--ro outgoing-bw-utilization      decimal64
+--rw ns-connections
  +--rw ns-connection* [ns-connection-id]
    +--rw ns-connection-id      uint32
    +--rw ns-connection-description? string
    +--rw src
    |   +--rw src-ep-id?   leafref
    +--rw dest
    |   +--rw dest-ep-id?   leafref
    +--rw (ns-slo-sle-policy)?
    |   +--:(standard)
    |   |   +--rw slo-sle-template?   leafref
    |   +--:(custom)
    |   |   +--rw slo-policy
    |   |       +--rw policy-description? string
    |   |       +--rw ns-metric-bounds
    |   |           +--rw ns-metric-bound* [metric-type]
    |   |           |       +--rw metric-type      identityref

```



```

|      |      +--rw metric-unit          string
|      |      +--rw value-description?   string
|      |      +--rw bound?               uint64
|      +--rw sle-policies
|          +--rw security-sle*           identityref
|          +--rw isolation?              identityref
|          +--rw max-occupancy-level?    uint8
+--rw monitoring-type?                  ns-monitoring-type
+--ro ns-connection-monitoring
    +--ro latency?                      yang:gauge64
    +--ro jitter?                       yang:gauge32
    +--ro loss-ratio?                   decimal64

```

Figure 3

5. IETF Network Slice Templates

The 'ns-templates' container (Figure 3) is used by service provider of the NSC to define and maintain a set of common IETF Network Slice templates that apply to one or several IETF Network Slices. The exact definition of the templates is deployment specific to each network provider.

The model includes only the identifiers of SLO and SLE templates. When creation of IETF Network slice, the SLO and SLE policies can be easily identified.

The following shows an example where two network slice templates can be retrieved by the upper layer management system:


```
{
  "ietf-network-slices": {
    "ns-templates": {
      "slo-sle-template": [
        {
          "id": "GOLD-template",
          "template-description": "Two-way bandwidth: 1 Gbps,
            one-way latency 100ms "
          "sle-isolation": "ns-isolation-shared",
        },
        {
          "id": "PLATINUM-template",
          "template-description": "Two-way bandwidth: 1 Gbps,
            one-way latency 50ms "
          "sle-isolation": "ns-isolation-dedicated",
        },
      ],
    }
  }
}
```

6. IETF Network Slice Modeling Description

The 'ietf-network-slice' is the data structure that abstracts an IETF Network Slice of the IETF network. Each 'ietf-network-slice' is uniquely identified by an identifier: 'ns-id'.

An IETF Network Slice has the following main parameters:

- o "ns-id": Is an identifier that is used to uniquely identify the IETF Network Slice within NSC.
- o "ns-description": Gives some description of an IETF Network Slice service.
- o "ns-connectivity-type": Indicates the network connectivity type for the IETF Network Slice: Hub-and-Spoke, any-to-any, or custom type.
- o "status": Is used to show the operative and administrative status of the IETF Network Slice, and can be used as indicator to detect network slice anomalies.
- o "ns-tag": Is used to show the correlation between higher level function and the IETF network slices. If provided, this parameter may be used by IETF Network Slice Controller (NSC) during the realization. It may also be used by NSC for monitoring and assurance of the IETF network slices where NSC can notify the

higher system by issuing the notifications. It is noted that a single higher level customer might have multiple IETF Network Slices for a single application. This attribute may be used by NSC to also correlated multiple IETF network slices for a single application.

- o "ns-slo-sle-policy": Defines SLO and SLE policies for the "ietf-network-slice". More description are provided in [Section 6.2](#)

The "ns-endpoint" is an abstrac entity that represents a set of matching rules applied to an IETF network edge device or a customer network edge device involved in the IETF Network Slice and each 'ns-endpoint' belongs to a single 'ietf-network-slice'. More description are provided in [Section 6.3](#)

6.1. IETF Network Slice Connectivity Type

Based on the customer's traffic pattern requirements, an IETF Network Slice connection type could be point-to-point (P2P), point-to-multipoint (P2MP), multipoint-to-point (MP2P), or multipoint-to-multipoint (MP2MP). The "ns-connectivity-type" under the node "ietf-network-slice" is used for this.

For the connectivity requirements, the model proposes to support any-to-any, Hub-and-Spoke (where Hubs can exchange traffic), and the custom. By default, the any-to-any is used. New connectivity type could be added via augmentation or by list of 'ns-connection' specified.

In addition, "ep-role" under the node "ns-endpoint" also needs to be defined, which specifies the role of the NSE in a particular Network Slice connectivity type. In the any-to-any, all NSEs MUST have the same role, which will be "any-to-any-role". In the Hub-and-Spoke, NSEs MUST have a Hub role or a Spoke role.

6.2. IETF Network Slice SLO and SLE Policy

As defined in [[I-D.ietf-teas-ietf-network-slices](#)], the SLO policy of an IETF Network Slice defines the minimum IETF Network Slice SLO attributes, and additional attributes can be added as needed.

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

The policy could both apply one per Network Slice or per connection 'ns-connection'.

The model allows multiple SLO and SLE attributes to be combined to meet different SLO and SLE requirements. For example, some NSs are used for video services and require high bandwidth, some NSs are used for key business services and request low latency and reliability, and some NSs need to provide connections for a large number of NSEs. That is, not all SLO or SLE attributes must be specified to meet the particular requirements of a slice.

"ns-metric-bounds" contains all these variations, which includes a list of "ns-metric-bound" and each "ns-metric-bound" could specify a particular "metric-type". "metric-type" is defined with YANG identity and the YANG module supports the following options:

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

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

"network-slice-slo-one-way-latency": Indicates the maximum one-way latency between two NSE.

"network-slice-slo-two-way-latency": Indicates the maximum round-trip latency between two NSE.

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

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

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

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

"ns-slo-availability": Is defined as the ratio of up-time to total_time(up-time+down-time), where up-time is the time the IETF

Network Slice is available in accordance with the SLOs associated with it.

Some other Network Slice SLOs or SLEs could be extended when needed.

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

```
{
  "ietf-network-slices": {
    "ietf-network-slice": {
      "slo-policy": {
        "policy-description": "video-service-policy",
        "ns-metric-bounds": {
          "ns-metric-bound": [
            {
              "metric-type": "ns-slo-one-way-bandwidth",
              "metric-unit": "mbps",
              "bound": "1000"
            },
            {
              "metric-type": "ns-slo-availability",
              "bound": "99.9%"
            }
          ],
        }
      }
    }
  }
}
```

6.3. IETF Network Slice Endpoint (NSE)

An IETF Network Slice Endpoint has several characteristics:

- o "ep-id": Uniquely identifies the NSE within Network Slice Controller (NSC). The identifier is a string that allows any encoding for the local administration of the IETF Network Slice.
- o "location": Indicates NSE location information that facilitates NSC easy identification of a NSE.
- o "ep-role": Represents a connectivity type role of a NSE belonging to an IETF network slice, as described in [Section 6.1](#). The "ep-role" leaf defines the role of the endpoint in a particular NS connectivity type. In the any-to-any, all NSEs MUST have the same role, which will be "any-to-any-role".

- o "node-id": The NSE node information facilities NSC with easy identification of a NSE.
- o "ep-ip": The NSE IP information facilities NSC with easy identification of a NSE.
- o "ns-match-criteria": A matching policies to apply on a given NSE.
- o "ep-network-access-points": The list of the interfaces attached to an edge device of the IETF Network Slice by which the customer traffic is received.
- o "ep-rate-limit": Set the rate-limiting policies to apply on a given NSE, including ingress and egress traffic to ensure access security. When applied in the incoming direction, the rate-limit is applicable to the traffic from the NSE to the IETF scope Network that passes through the external interface. When Bandwidth is applied to the outgoing direction, it is applied to the traffic from the IETF Network to the NSE of that particular NS.
- o "ep-protocol": Specify the protocol for a NSE for exchanging control-plane information, e.g. L1 signaling protocol or L3 routing protocols, etc.
- o "status": Enable the control of the operative and administrative status of the NSE, can be used as indicator to detect NSE anomalies.

An NSE belong to a single IETF Network Slice. An IETF Network Slice involves two or more NSEs. An IETF Network Slice can be modified by adding new "ns-endpoint" or removing existing "ns-endpoint".

A NSE is used to define the matching rule on the customer traffic that can be injected to an IETF Network Slice. "network-slice-match-criteria" is defined to support different options. Classification can be based on many criteria, such as:

- o Physical interface: Indicates all the traffic received from the interface belongs to the IETF Network Slice.
- o Logical interface: For example, a given VLAN ID is used to identify an IETF Network Slice.
- o Encapsulation in the traffic header: For example, a source IP address is used to identify an IETF Network Slice.

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

- o NSE mapping to PE example: As shown in Figure 4 , customer of the IETF network slice would like to connect two NSEs to satisfy specific service, e.g., Network wholesale services. In this case, the IETF network slice endpoints are mapped to physical interfaces of PE nodes. The IETF network slice controller (NSC) uses 'node-id' (PE device ID), 'ep-network-access-points' (Two PE interfaces) to map the interfaces and corresponding services/tunnels/paths.

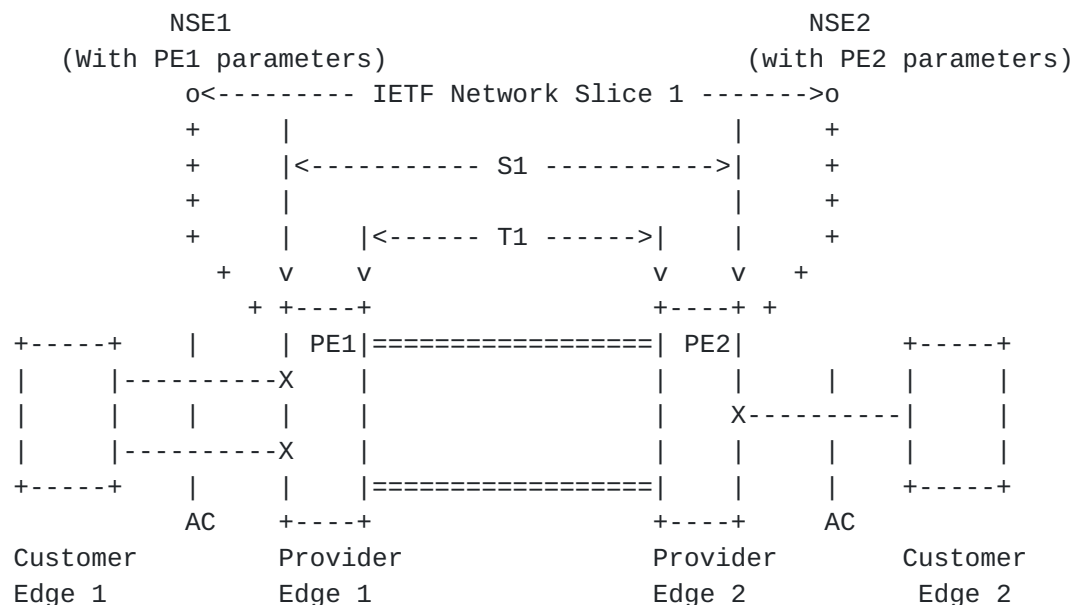
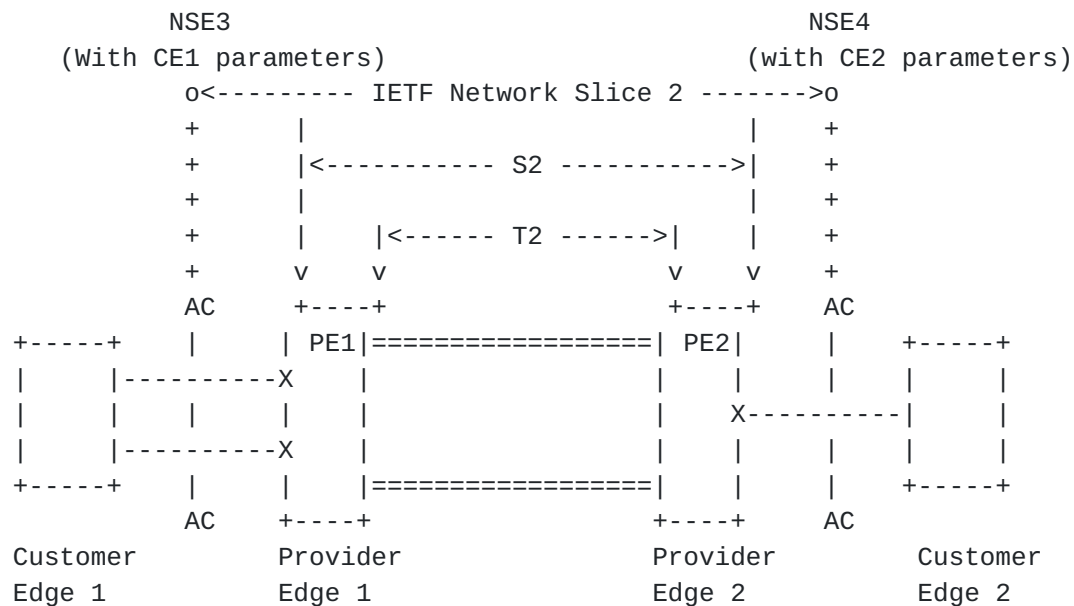


Figure 4

- o NSE mapping to CE-PE interface example: As shown in Figure 5 , customer of the IETF network slice would like to connect two NSEs to provide connectivity between transport portion of 5G RAN to 5G Core network functions. In this scenario, the IETF network slice endpoints (NSE) might be mapped to the respective PE-CE interface (see 3GPP TS 28.541 V17.1.0 [section 6.3.17](#) EP_Transport). The IETF network slice controller (NSC) uses 'node-id' (CE device ID) , 'ep-ip' (CE tunnel endpoint IP), 'network-slice-match-criteria'

(VLAN interface), 'ep-network-access-points' (Two nexthop interfaces) to map underlay services/tunnels/paths.



Legend:

- o: Representation of the IETF network slice endpoints (NSE)
- +: Mapping of NSE to PE or CE-PE interfaces on IETF network
- X: Physical interfaces used for realization of IETF network slice
- S2: L0/L1/L2/L3 services used for realization of IETF network slice
- T2: Tunnels used for realization of IETF network slice

Figure 5

7. IETF Network Slice Monitoring

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

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

- o Per NS connection: specified in 'ns-connection-monitoring' under the "ns-connection"
- o Per NS Endpoint: specified in 'ep-monitoring' under the "ns-endpoint"

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

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

8. IETF Network Slice NBI Module

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

```
<CODE BEGINS> file "ietf-network-slice@2021-07-06.yang"
module ietf-network-slice {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-network-slice";
  prefix ietf-ns;

  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Types.";
  }
  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 6991: Common YANG Types.";
  }
  import ietf-te-types {
    prefix te-types;
    reference
      "RFC 8776: Common YANG Data Types for Traffic Engineering.";
  }

  organization
    "IETF Traffic Engineering Architecture and Signaling (TEAS)
     Working Group";
  contact
    "WG Web: <https://tools.ietf.org/wg/teas/>
     WG List: <mailto:teas@ietf.org>
     Editor: Bo Wu <lane.wubo@huawei.com>
           : Dhruv Dhody <dhruv.ietf@gmail.com>
           : Reza Rokui <reza.rokui@nokia.com>
           : Tarek Saad <tsaad@juniper.net>";
  description
```


"This module contains a YANG module for the IETF Network Slice.

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

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

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

```
revision 2021-07-06 {
  description
    "initial version.";
  reference
    "RFC XXXX: A Yang Data Model for IETF Network Slice Operation";
}

/* Features */
/* Identities */

identity ns-isolation-type {
  description
    "Base identity for IETF Network slice isolation level.";
}

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

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

identity ns-security-type {
```



```
    description
      "Base identity for for IETF Network security level.";
  }

  identity ns-security-authenticate {
    base ns-security-type;
    description
      "IETF Network Slice requires authentication.";
  }

  identity ns-security-integrity {
    base ns-security-type;
    description
      "IETF Network Slice requires data integrity.";
  }

  identity ns-security-encryption {
    base ns-security-type;
    description
      "IETF Network Slice requires data encryption.";
  }

  identity ns-connectivity-type {
    description
      "Base identity for IETF Network Slice topology.";
  }

  identity any-to-any {
    base ns-connectivity-type;
    description
      "Identity for any-to-any IETF Network Slice topology.";
  }

  identity hub-spoke {
    base ns-connectivity-type;
    description
      "Identity for Hub-and-Spoke IETF Network Slice topology.";
  }

  identity custom {
    base ns-connectivity-type;
    description
      "Identity of a custom NS topology where Hubs can act as
      Spoke for certain parts of the network or Spokes as Hubs.";
  }

  identity endpoint-role {
    description
```



```
    "Base identity of a NSE role in an IETF Network Slice topology.";
}

identity any-to-any-role {
    base endpoint-role;
    description
        "Identity of any-to-any NS.";
}

identity spoke-role {
    base endpoint-role;
    description
        "A NSE is acting as a Spoke.";
}

identity hub-role {
    base endpoint-role;
    description
        "A NSE is acting as a Hub.";
}

identity custom-role {
    base endpoint-role;
    description
        "A NSE is custom role in the NS.";
}

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

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

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

identity ns-slo-one-way-latency {
    base ns-slo-metric-type;
```



```
    description
      "SLO one-way latency is upper bound of network latency when
      transmitting between two endpoints. The metric is defined in
      RFC7679";
  }

  identity ns-slo-two-way-latency {
    base ns-slo-metric-type;
    description
      "SLO two-way latency is upper bound of network latency when
      transmitting between two endpoints. The metric is defined in
      RFC2681";
  }

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

  identity ns-slo-two-way-delay-variation {
    base ns-slo-metric-type;
    description
      "SLO two-way delay variation is defined by RFC5481, is the
      difference in the round-trip delay between sequential packets
      between two endpoints.";
  }

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

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

  identity ns-slo-availability {
    base ns-slo-metric-type;
```



```
    description
      "SLO availability level.";
  }

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

  identity ns-phy-interface-match {
    base ns-match-type;
    description
      "Use the physical interface as match criteria for the IETF
      Network Slice traffic.";
  }

  identity ns-vlan-match {
    base ns-match-type;
    description
      "Use the VLAN ID as match criteria for the IETF Network Slice
      traffic.";
  }

  identity ns-label-match {
    base ns-match-type;
    description
      "Use the MPLS label as match criteria for the IETF Network
      Slice traffic.";
  }

  /*
   * Identity for availability-type
   */

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

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

  identity level-2 {
    base availability-type;
```



```
    description
      "level 2: 99.999%";
  }

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

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

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

/* typedef */

typedef operational-type {
  type enumeration {
    enum up {
      value 0;
      description
        "Operational status UP.";
    }
    enum down {
      value 1;
      description
        "Operational status DOWN.";
    }
    enum unknown {
      value 2;
      description
        "Operational status UNKNOWN.";
    }
  }
  description
    "This is a read-only attribute used to determine the
    status of a particular element.";
}

typedef ns-monitoring-type {
```



```
type enumeration {
  enum one-way {
    description
      "Represents one-way measurments monitoring type.";
  }
  enum two-way {
    description
      "represents two-way measurements monitoring type.";
  }
}
description
  "An enumerated type for monitoring on a IETF Network Slice
  connection.";
}

/* Groupings */

grouping status-params {
  description
    "A grouping used to join operational and administrative status.";
  container status {
    description
      "A container for the administrative and operational state.";
    leaf admin-enabled {
      type boolean;
      description
        "The administrative status.";
    }
    leaf oper-status {
      type operational-type;
      config false;
      description
        "The operational status.";
    }
  }
}

grouping ns-match-criteria {
  description
    "A grouping for the IETF Network Slice match definition.";
  container ns-match-criteria {
    description
      "Describes the IETF Network Slice match criteria.";
    list ns-match-criterion {
      key "match-type";
      description
        "List of the IETF Network Slice traffic match criteria.";
      leaf match-type {
```



```
    type identityref {
      base ns-match-type;
    }
    description
      "Identifies an entry in the list of the IETF Network Slice
      match criteria.";
  }
  list values {
    key "index";
    description
      "List of match criteria values.";
    leaf index {
      type uint8;
      description
        "Index of an entry in the list.";
    }
    leaf value {
      type string;
      description
        "Describes the IETF Network Slice match criteria, e.g.
        IP address, VLAN, etc.";
    }
  }
}
}
}

grouping ns-connection-group-metric-bounds {
  description
    "Grouping of Network Slice metric bounds that
    are shared amongst multiple connections of a Network
    Slice.";
  leaf ns-slo-shared-bandwidth {
    type te-types:te-bandwidth;
    description
      "A limit on the bandwidth that is shared amongst
      multiple connections of an IETF Network Slice.";
  }
}

grouping ns-sles {
  description
    "Indirectly Measurable Objectives of a IETF Network
    Slice.";
  container sle-policies {
    description
      "Container for the policy of SLEs applicable to
      IETF Network Slice.";
  }
}
```



```
leaf-list security-sle {
  type identityref {
    base ns-security-type;
  }
  description
    "The IETF Network Slice security SLE(s)";
}
leaf isolation {
  type identityref {
    base ns-isolation-type;
  }
  default "ns-isolation-shared";
  description
    "The IETF Network Slice isolation SLE requirement.";
}
leaf max-occupancy-level {
  type uint8 {
    range "1..100";
  }
  description
    "The maximal occupancy level specifies the number of flows to
    be admitted.";
}
}
}

grouping ns-metric-bounds {
  description
    "IETF Network Slice metric bounds grouping.";
  container ns-metric-bounds {
    description
      "IETF Network Slice metric bounds container.";
    list ns-metric-bound {
      key "metric-type";
      description
        "List of IETF Network Slice metric bounds.";
      leaf metric-type {
        type identityref {
          base ns-slo-metric-type;
        }
        description
          "Identifies an entry in the list of metric type
          bounds for the IETF Network Slice.";
      }
      leaf metric-unit {
        type string;
        mandatory true;
        description
```



```
        "The metric unit of the parameter. For example,
          s, ms, ns, and so on.";
    }
    leaf value-description {
        type string;
        description
            "The description of previous value. ";
    }
    leaf bound {
        type uint64;
        default "0";
        description
            "The Bound on the Network Slice connection metric. A
             zero indicate an unbounded upper limit for the
             specific metric-type.";
    }
}
}
}

grouping ep-network-access-points {
    description
        "Grouping for the endpoint network access definition.";
    container ep-network-access-points {
        description
            "List of network access points.";
        list ep-network-access-point {
            key "network-access-id";
            description
                "The IETF Network Slice network access points
                 related parameters.";
            leaf network-access-id {
                type string;
                description
                    "Uniquely identifier a network access point.";
            }
            leaf network-access-description {
                type string;
                description
                    "The network access point description.";
            }
            leaf network-access-node-id {
                type string;
                description
                    "The network access point node ID in the case of
                     multi-homing.";
            }
            leaf network-access-tp-id {
```



```
        type string;
        description
            "The termination port ID of the EP network access
            point.";
    }
    leaf network-access-tp-ip {
        type inet:host;
        description
            "The IP address of the EP network access point.";
    }
    /* Per ep-network-access-point rate limits */
    uses ns-rate-limit;
}
}
```

```
grouping endpoint-monitoring-parameters {
    description
        "Grouping for the endpoint monitoring parameters.";
    container ep-monitoring {
        config false;
        description
            "Container for endpoint monitoring parameters.";
        leaf incoming-utilized-bandwidth {
            type te-types:te-bandwidth;
            description
                "Incoming bandwidth utilization at an endpoint.";
        }
        leaf incoming-bw-utilization {
            type decimal64 {
                fraction-digits 5;
                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
                "Outgoing bandwidth utilization at an endpoint.";
        }
        leaf outgoing-bw-utilization {
            type decimal64 {
                fraction-digits 5;
                range "0..100";
            }
        }
    }
}
```



```
    }
    units "percent";
    mandatory true;
    description
        "To be used to define the bandwidth utilization
        as a percentage of the available bandwidth.";
    }
}

grouping common-monitoring-parameters {
    description
        "Grouping for link-monitoring-parameters.";
    leaf latency {
        type yang:gauge64;
        units "usec";
        description
            "The latency statistics per Network Slice connection.
            RFC2681 and RFC7679 discuss round trip times and one-way
            metrics, respectively";
    }
    leaf jitter {
        type yang:gauge32;
        description
            "The jitter statistics per Network Slice member
            as defined by RFC3393.";
    }
    leaf loss-ratio {
        type decimal64 {
            fraction-digits 6;
            range "0 .. 50.331642";
        }
        description
            "Packet loss as a percentage of the total traffic
            sent over a configurable interval. The finest precision is
            0.000003%. where the maximum 50.331642%.";
        reference
            "RFC 7810, section-4.4";
    }
}

grouping geolocation-container {
    description
        "A grouping containing a GPS location.";
    container location {
        description
            "A container containing a GPS location.";
        leaf altitude {
```



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

// geolocation-container

grouping ns-rate-limit {
    description
        "The Network Slice rate limit grouping.";
    container ep-rate-limit {
        description
            "Container for the asymmetric traffic control";
        leaf incoming-rate-limit {
            type te-types:te-bandwidth;
            description
                "The rate-limit imposed on incoming traffic.";
        }
        leaf outgoing-rate-limit {
            type te-types:te-bandwidth;
            description
                "The rate-limit imposed on outgoing traffic.";
        }
    }
}

grouping endpoint {
    description
```



```
    "IETF Network Slice endpoint related information";
  leaf ep-id {
    type string;
    description
      "unique identifier for the referred IETF Network
       Slice endpoint";
  }
  leaf ep-description {
    type string;
    description
      "endpoint name";
  }
  leaf ep-role {
    type identityref {
      base endpoint-role;
    }
    default "any-to-any-role";
    description
      "Role of the endpoint in the IETF Network Slice.";
  }
  uses geolocation-container;
  leaf node-id {
    type string;
    description
      "Uniquely identifies an edge node within the IETF slice
       network.";
  }
  leaf ep-ip {
    type inet:host;
    description
      "The address of the endpoint IP address.";
  }
  uses ns-match-criteria;
  uses ep-network-access-points;
  uses ns-rate-limit;
  /* Per NSE rate limits */
  container ep-protocol {
    description
      "Describes protocol for the Network Slice Endpoint.";
  }
  uses status-params;
  uses endpoint-monitoring-parameters;
}

//ns-endpoint

grouping ns-connection {
  description
```



```
    "The Network Slice connection is described in this container.";
  leaf ns-connection-id {
    type uint32;
    description
      "The Network Slice connection identifier";
  }
  leaf ns-connection-description {
    type string;
    description
      "The Network Slice connection description";
  }
  container src {
    description
      "the source of Network Slice link";
    leaf src-ep-id {
      type leafref {
        path "/network-slices/network-slice"
          + "/ns-endpoints/ns-endpoint/ep-id";
      }
      description
        "reference to source Network Slice endpoint";
    }
  }
  container dest {
    description
      "the destination of Network Slice link ";
    leaf dest-ep-id {
      type leafref {
        path "/network-slices/network-slice"
          + "/ns-endpoints/ns-endpoint/ep-id";
      }
      description
        "reference to dest Network Slice endpoint";
    }
  }
  uses ns-slo-sle-policy;
  /* Per connection ns-slo-sle-policy overrides
   * the per network slice ns-slo-sle-policy.
   */
  leaf monitoring-type {
    type ns-monitoring-type;
    description
      "One way or two way monitoring type.";
  }
  container ns-connection-monitoring {
    config false;
    description
      "SLO status Per network-slice endpoint to endpoint ";
  }
```



```
        uses common-monitoring-parameters;
    }
}

//ns-connection

grouping slice-template {
    description
        "Grouping for slice-templates.";
    container ns-slo-sle-templates {
        description
            "Contains a set of network slice templates to
            reference in the IETF network slice.";
        list ns-slo-sle-template {
            key "id";
            leaf id {
                type string;
                description
                    "Identification of the Service Level Objective (SLO)
                    and Service Level Expectation (SLE) template to be used.
                    Local administration meaning.";
            }
            leaf template-description {
                type string;
                description
                    "Description of the SLO & SLE policy template.";
            }
        }
        description
            "List for SLO and SLE template identifiers.";
    }
}

/* Configuration data nodes */

grouping ns-slo-sle-policy {
    description
        "Network Slice policy grouping.";
    choice ns-slo-sle-policy {
        description
            "Choice for SLO and SLE policy template.
            Can be standard template or customized template.";
        case standard {
            description
                "Standard SLO template.";
            leaf slo-sle-template {
                type leafref {
                    path "/network-slices"
```



```
        + "/ns-slo-sle-templates/ns-slo-sle-template/id";
    }
    description
        "Standard SLO and SLE template to be used.";
}
}
case custom {
    description
        "Customized SLO template.";
    container slo-policy {
        description
            "Contains the SLO policy.";
        leaf policy-description {
            type string;
            description
                "Description of the SLO policy.";
        }
        uses ns-metric-bounds;
    }
    uses ns-sles;
}
}
}

container network-slices {
    description
        "IETF network-slice configurations";
    uses slice-template;
    list network-slice {
        key "ns-id";
        description
            "a network-slice is identified by a ns-id";
        leaf ns-id {
            type string;
            description
                "A unique network-slice identifier across an IETF NSC ";
        }
        leaf ns-description {
            type string;
            description
                "Give more description of the network slice";
        }
        leaf-list ns-tag {
            type string;
            description
                "Network Slice tag for operational management";
        }
        leaf ns-connectivity-type {
```



```
    type identityref {
      base ns-connectivity-type;
    }
    default "any-to-any";
    description
      "Network Slice topology.";
  }
  uses ns-slo-sle-policy;
  uses status-params;
  container ns-endpoints {
    description
      "Endpoints";
    list ns-endpoint {
      key "ep-id";
      uses endpoint;
      description
        "list of endpoints in this slice";
    }
  }
  container ns-connections {
    description
      "Connections container";
    list ns-connection {
      key "ns-connection-id";
      description
        "List of Network Slice connections.";
      uses ns-connection;
    }
  }
}
//ietf-network-slice list
}
```

<CODE ENDS>

9. Security Considerations

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

The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a

preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

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

o /ietf-network-slice/network-slices/network-slice

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

10. IANA Considerations

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

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

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

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

11. Acknowledgments

The authors wish to thank Sergio Belotti, Qin Wu, Susan Hares, Eric Grey, and many other NS DT members for their helpful comments and suggestions.

12. References

12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", [BCP 81](#), [RFC 3688](#), DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", [RFC 6241](#), DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", [RFC 6242](#), DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", [RFC 6991](#), DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", [RFC 7950](#), DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", [RFC 8040](#), DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", [BCP 215](#), [RFC 8340](#), DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, [RFC 8341](#), DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.

- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", [RFC 8342](#), DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", [RFC 8446](#), DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8640] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Dynamic Subscription to YANG Events and Datastores over NETCONF", [RFC 8640](#), DOI 10.17487/RFC8640, September 2019, <<https://www.rfc-editor.org/info/rfc8640>>.
- [RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", [RFC 8641](#), DOI 10.17487/RFC8641, September 2019, <<https://www.rfc-editor.org/info/rfc8641>>.
- [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>>.

12.2. Informative References

- [I-D.geng-teas-network-slice-mapping]
Geng, X., Dong, J., Pang, R., Han, L., Niwa, T., Jin, J., Liu, C., and N. Nageshar, "5G End-to-end Network Slice Mapping from the view of Transport Network", [draft-geng-teas-network-slice-mapping-03](#) (work in progress), February 2021.
- [I-D.ietf-teas-actn-vn-yang]
Lee, Y., Dhody, D., Ceccarelli, D., Bryskin, I., and B. Y. Yoon, "A YANG Data Model for VN Operation", [draft-ietf-teas-actn-vn-yang-11](#) (work in progress), February 2021.
- [I-D.ietf-teas-ietf-network-slices]
Farrel, A., Gray, E., Drake, J., Rokui, R., Homma, S., Makhijani, K., Contreras, L. M., and J. Tantsura, "Framework for IETF Network Slices", [draft-ietf-teas-ietf-network-slices-00](#) (work in progress), April 2021.

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

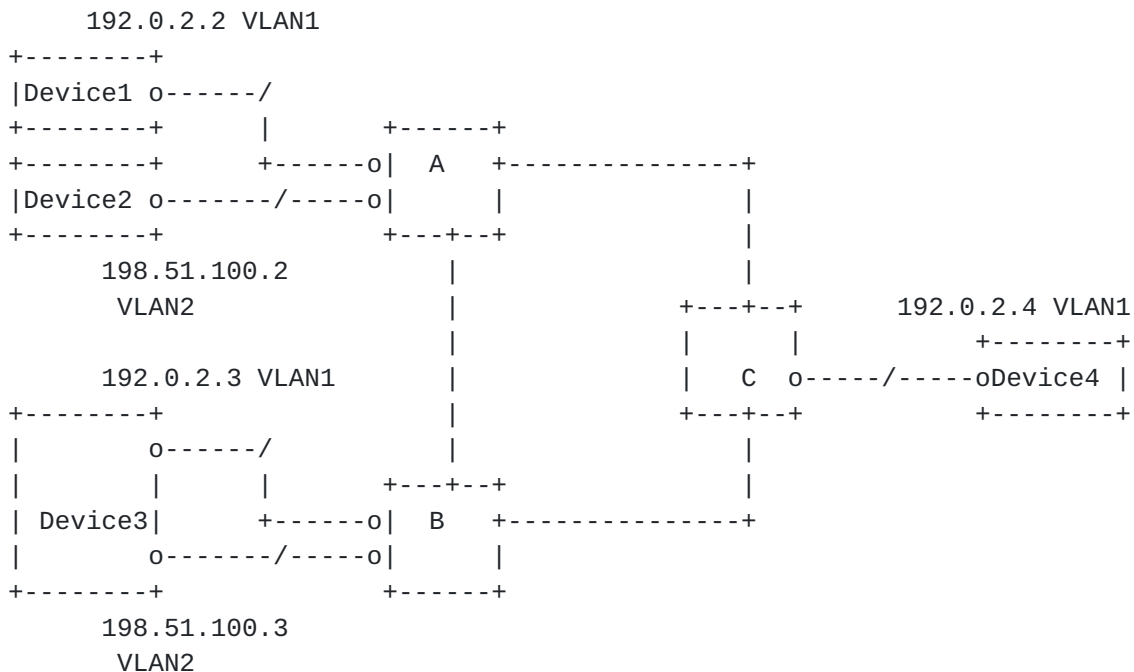
Liu, X., Tantsura, J., Bryskin, I., Contreras, L. M., Wu, Q., Belotti, S., and R. Rokui, "IETF Network Slice YANG Data Model", [draft-liu-teas-transport-network-slice-yang-02](#) (work in progress), November 2020.

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

[Appendix A](#). IETF Network Slice NBI Model Usage Example

The following example describes a simplified service configuration of two IETF Network slice instances:

- o IETF Network Slice 1 on Device1, Device3, and Device4, with any-to-any connectivity type
- o IETF Network Slice 2 on Device2, Device3, with any-to-any connectivity type



POST: /restconf/data/ietf-network-slice:ietf-network-slices

Host: example.com

Content-Type: application/yang-data+json

```

{
  "network-slices":{
    "network-slice":[
      {

```



```
"ns-id":"1",
"ns-description":"slice1",
"ns-connectivity-type":"any-to-any",
"ns-endpoints":{
  "ns-endpoint":[
    {
      "ep-id":"11",
      "ep-description":"slice1 ep1 connected to device 1",
      "ep-role":"any-to-any-role",
      "ns-match-criteria":[
        {
          "match-type":"ns-vlan-match",
          "value":[
            {
              "index":"1",
              "value":"1"
            }
          ]
        }
      ]
    }
  ],
  {
    "ep-id":"12",
    "ep-description":"slice1 ep2 connected to device 3",
    "ep-role":"any-to-any-role",
    "ns-match-criteria":[
      {
        "match-type":"ns-vlan-match",
        "value":[
          {
            "index":"1",
            "value":"20"
          }
        ]
      }
    ]
  }
],
  {
    "ep-id":"13",
    "ep-description":"slice1 ep3 connected to device 4",
    "ep-role":"any-to-any-role",
    "ns-match-criteria":[
      {
        "match-type":"ns-vlan-match",
        "value":[
          {
            "index":"1",
            "value":"1"
          }
        ]
      }
    ]
  }
]
```



```

    }
  ]
}
},
{
  "ns-id":"ns2",
  "ns-description":"slice2",
  "ns-connectivity-type":"any-to-any",
  "ns-endpoints":{
    "ns-endpoint":[
      {
        "ep-id":"21",
        "ep-description":"slice2 ep1 connected to device 2",
        "ep-role":"any-to-any-role",
        "ns-match-criteria":[
          {
            "match-type":"ns-vlan-match",
            "value":[
              {
                "index":"1",
                "value":"2"
              }
            ]
          }
        ]
      }
    ]
  },
  {
    "ep-id":"22",
    "ep-description":"slice2 ep2 connected to device 3",
    "ep-role":"any-to-any-role",
    "ns-match-criteria":[
      {
        "match-type":"ns-vlan-match",
        "value":[
          {
            "index":"1",
            "value":"2"
          }
        ]
      }
    ]
  }
]
}

```



```

    }
  ]
}

```

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

According to the 3.3.1. Northbound Interface (NBI) [[I-D.ietf-teas-ietf-network-slices](#)], the IETF Network Slice NBI 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 IETF Network Slice NBI is classified as Customer Service Model.

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

B.1. ACTN VN Model Augmentation

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

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

- o 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.
- o AP: An AP is a logical identifier used to identify the access link which is shared between the customer and the IETF scoped Network.
- o VN-AP: A VN-AP is a logical binding between an AP and a given VN.
- o VN-member: A VN-member is an abstract edge-to-edge link between any two APs or VN-APs. Each link is formed as an E2E tunnel across the underlying networks.

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

B.2. [RFC8345](#) Augmentation Model

The difference between the IETF Network Slice NBI requirements and the IETF basic network model is that the IETF Network Slice NBI requests abstract 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:

- o Network: a transport network list and an list of nodes contained in the network
- o Link: "links" list and "termination points" list describe how nodes in a network are connected to each other
- o Support network: vertical layering relationships between IETF Network Slice networks and underlay networks

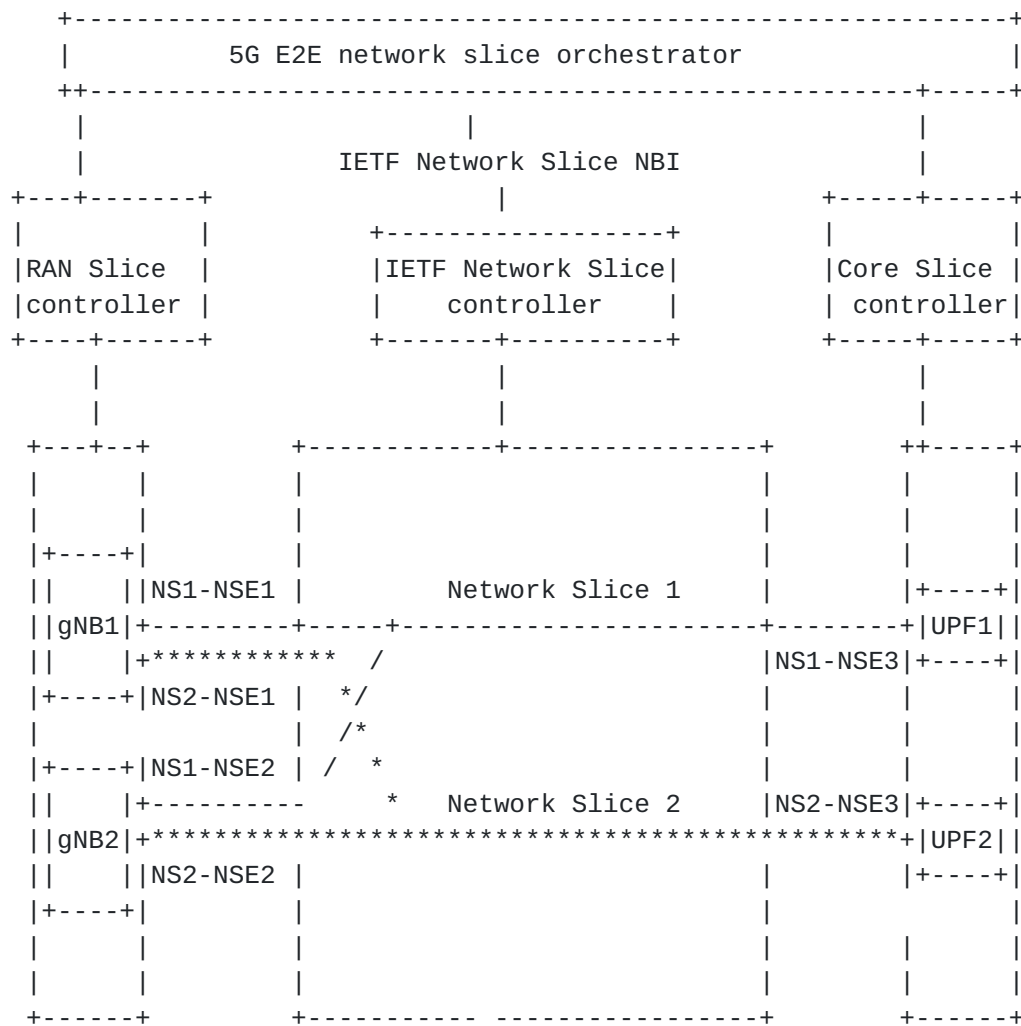
Based on this structure, the IETF Network Slice-specific SLO attributes nodes are augmented on the Network Topologies model,, e.g. isolation etc. However, this modeling design requires the slice network to expose a lot of details of the network, such as the actual topology including nodes interconnection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

Appendix C. [Appendix B](#) IETF Network Slice Match Criteria

5G is a use case of the IETF Network Slice and 5G End-to-end Network Slice Mapping from the view of IETF Network [[I-D.geng-teas-network-slice-mapping](#)]

defines two types of Network Slice interconnection and differentiation methods: by physical interface or by TNSII (Transport Network Slice Interworking Identifier). TNSII is a field in the

packet header when different 5G wireless network slices are transported through a single physical interfaces of the IETF scoped Network. In the 5G scenario, "network-slice-match-criteria" refers to TNSII.



As shown in the figure, gNodeB 1 and gNodeB 2 use IP gNB1 and IP gNB2 to communicate with the IETF network, respectively. In addition, the traffic of NS1 and NS2 on gNodeB 1 and gNodeB 2 is transmitted through the same access links to the IETF slice network. The IETF slice network need to to distinguish different IETF Network Slice traffic of same gNB. Therefore, in addition to using "node-id" and "ep-ip" to identify a Network Slice Endpoint, other information is needed along with these parameters to uniquely distinguish a NSE. For example, VLAN IDs in the user traffic can be used to distinguish the NSEs of gNBs and UPFs.

Authors' Addresses

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

Email: lana.wubo@huawei.com

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

Email: dhruv.ietf@gmail.com

Reza Rokui
Nokia

Email: reza.rokui@nokia.com

Tarek Saad
Juniper Networks

Email: tsaad@juniper.net

Liuyan Han
China Mobile

Email: hanliuyan@chinamobile.com

