

Workgroup: TEAS

Internet-Draft:

draft-ietf-teas-ietf-network-slice-nbi-yang-01

Published: 4 March 2022

Intended Status: Standards Track

Expires: 5 September 2022

Authors: B. Wu	D. Dhody	R. Rokui
Huawei Technologies	Huawei Technologies	Ciena
T. Saad	L. Han	
Juniper Networks	China Mobile	

IETF Network Slice Service YANG Model

Abstract

This document defines a YANG model for the IETF Network Slice service model. The model can be used by a IETF Network Slice customer to manage IETF Network Slice from an IETF Network Slice Controller (NSC).

Status of This Memo

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

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

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

This Internet-Draft will expire on 5 September 2022.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in

Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

- [1. Introduction](#)
- [2. Conventions used in this document](#)
 - [2.1. Acronyms](#)
- [3. IETF Network Slice Service Model Usage](#)
- [4. IETF Network Slice Service Model Overview](#)
- [5. IETF Network Slice Templates](#)
- [6. IETF Network Slice Modeling Description](#)
 - [6.1. IETF Network Slice Connectivity](#)
 - [6.2. IETF Network Slice SLO and SLE Policy](#)
 - [6.3. IETF Network Slice Endpoint \(NSE\)](#)
- [7. IETF Network Slice Monitoring](#)
- [8. IETF Network Slice Service Module](#)
- [9. Security Considerations](#)
- [10. IANA Considerations](#)
- [11. Acknowledgments](#)
- [12. Contributors](#)
- [13. References](#)
 - [13.1. Normative References](#)
 - [13.2. Informative References](#)
- [Appendix A. IETF Network Slice Service Model Usage Example](#)
- [Appendix B. Comparison with Other Possible Design choices for IETF Network Slice Service Interface](#)
 - [B.1. ACTN VN Model Augmentation](#)
 - [B.2. RFC8345 Augmentation Model](#)
- [Appendix C. Appendix B IETF Network Slice Match Criteria](#)
- [Authors' Addresses](#)

1. Introduction

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

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

The IETF Network Slice Controller (NSC) is a logical entity that allows customers to manage IETF network slices. Customers operate on abstract IETF network slices. Details related to the production of slices that fulfil the request are internal to the entity that

operates the network. Such details are deployment- and implementation-specific.

The NSC receives request from its customer-facing interface (e.g., from a management system). This interface carries data objects the IETF network slice user provides, describing the needed IETF network slices in terms of topology, target service level objectives (SLO), and also monitoring and reporting requirements. These requirements are then translated into technology-specific actions that are implemented in the underlying network using a network-facing interface. The details of how the IETF network slices are put into effect are out of scope for this document.

The YANG model discussed in this document describes the requirements of an IETF Network Slice from the point of view of the customer. It is thus classified as customer service model in [[RFC8309](#)].

Editorial Note: (To be removed by RFC Editor)

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

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

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

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

2. Conventions used in this document

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

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

- *client

- *configuration data

- *state data

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

The tree diagram used in this document follow the notation defined in [[RFC8340](#)].

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

This document defines the following terms:

*IETF Network Slice Connection (NS-Connection): Refers to connectivity construct defined in [[I-D.ietf-teas-ietf-network-slices](#)]. An IETF Network Slice can have one or multiple NS-Connections.

*IETF Network Slice Connection (NS-Connection-group): When an IETF Network Slice has multiple NS-connections. The connections with similar SLO or SLE are treated as one NS-connection group. An IETF Network Slice can have one or multiple NS-Connection-groups.

2.1. Acronyms

The following acronyms are used in the document:

CE Customer Edge
NSC Network Slice Controller
NSE Network Slice Endpoint
MTU Maximum Transmission Unit
PE Provider Edge
SLE Service Level Expectation
SLO Service Level Objective

3. IETF Network Slice Service Model Usage

The intention of the IETF Network Slice service model is to allow the customer to manage IETF Network Slices. In particular, the model allows customers to operate in an abstract and technology-agnostic manner, with details of the IETF Network Slices realization hidden.

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

As shown in [Figure 1](#), in all these use-cases, the model is used by the higher management system to communicate with NSC for life cycle manage of IETF Network Slices including both enablement and monitoring. For example, in 5G E2E (End-to-end) network slicing use-

case the E2E network slice orchestrator acts as the higher layer system to request the IETF Network Slices. The interface is used to support dynamic IETF Network Slice creation and its lifecycle management to facilitate end-to-end network slice services.

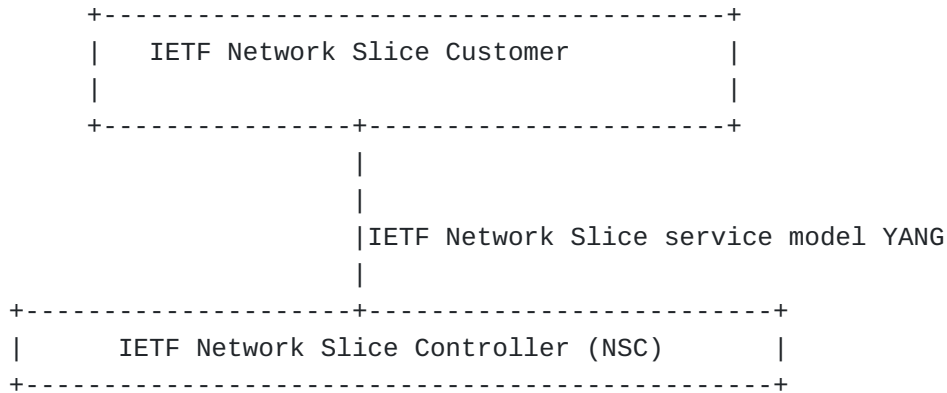
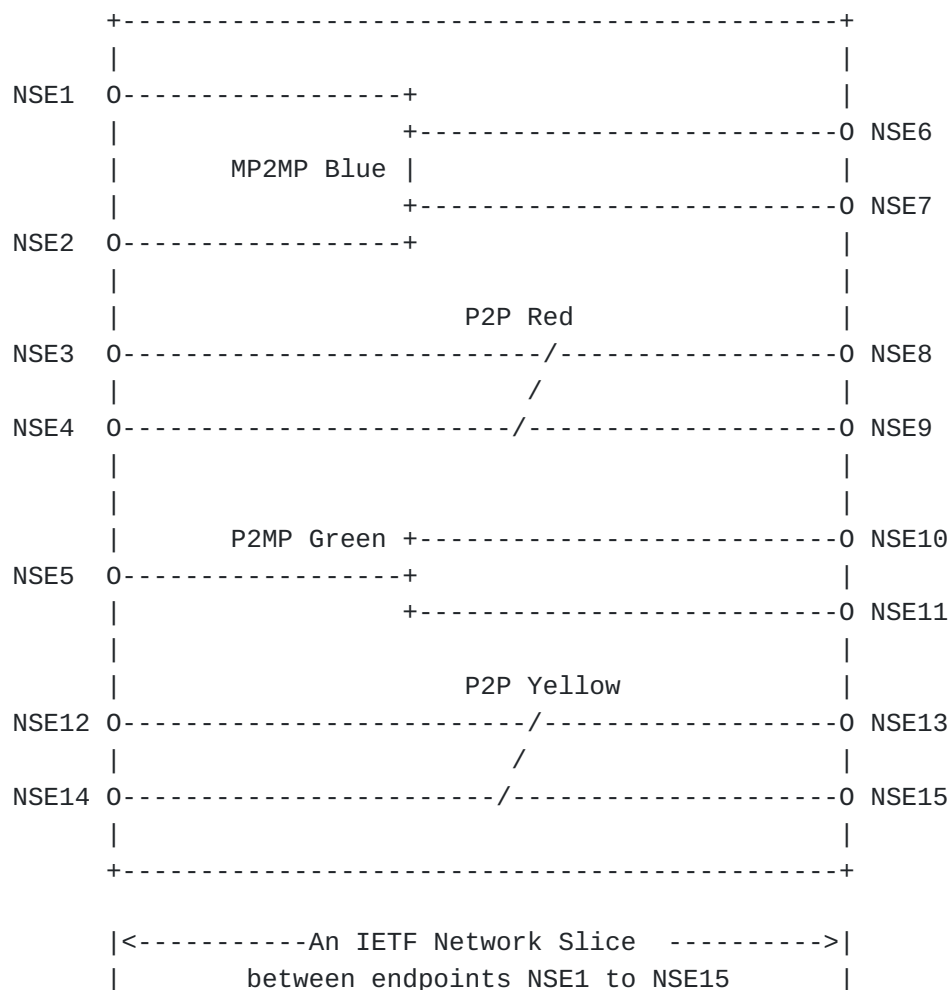


Figure 1: IETF Network Slice Service Reference Architecture

4. IETF Network Slice Service Model Overview

As defined in [[I-D.ietf-teas-ietf-network-slices](#)], an IETF Network Slice service is specified in terms of a set of endpoints, a set of one or more connectivity constructs (point-to-point (P2P), point-to-multipoint (P2MP), or multipoint-to-multipoint (MP2MP) between subsets of these endpoints, and a set of SLOs and SLEs for each endpoints sending to each connectivity construct. A connection construct is the basic connectivity unit of a network slice, and a slice service may consist of one or more connection constructs. The endpoints are conceptual points that could map to a device, application or a network function. And the specific service requirements, typically expressed as bandwidth, latency, latency variation, and other desired or required characteristics, such as security, MTU, traffic-type (e.g., IPv4, IPv6, Ethernet or unstructured) or a higher-level behavior to process traffic according to user-application (which may be realized using network function). An example of an IETF network slice containing multiple connectivity constructs is shown in [Figure 2](#) .



NSE: IETF Network Slice Endpoint
 0: Represents IETF Network Slice Endpoints

Figure 2: An IETF Network Slice Example

As shown in the example, an IETF network slice may have multiple NSEs. The NSEs are the ingress/egress points where traffic enters/exits the IETF network slice. As the edge of the IETF network slice, the NSEs also delimit a topological network portion within which the committed SLOs apply.

When an NSC receives a message via its customer-facing interface for creation/modification of an IETF network slice, it uses the provided NSEs to retrieve the corresponding service demarcation link or slice provider edge node" (e.g., PE). The NSC further maps them to the appropriate service/tunnel/path endpoints in the underlying network. It then uses services/tunnels/paths to realize the IETF network slice.

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

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

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

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

```

module: ietf-network-slice
+--rw network-slices
  +--rw ns-slo-sle-templates
  | +--rw ns-slo-sle-template* [id]
  |   +--rw id string
  |   +--rw template-description? string
  +--rw network-slice* [ns-id]
    +--rw ns-id string
    +--rw ns-description? string
    +--rw ns-tags
    | +--rw ns-tag* [index]
    |   +--rw index uint32
    |   +--rw ns-tag-type? identityref
    |   +--rw ns-tag-value? string
    +--rw (ns-slo-sle-policy)?
    | +--:(standard)
    | | +--rw slo-sle-template? leafref
    | +--:(custom)
    |   +--rw slo-sle-policy
    |     +--rw policy-description? string
    |     +--rw ns-metric-bounds
    |     | +--rw ns-metric-bound* [metric-type]
    |     |   +--rw metric-type identityref
    |     |   +--rw metric-unit string
    |     |   +--rw value-description? string
    |     |   +--rw bound? uint64
    |     +--rw security* identityref
    |     +--rw isolation? identityref
    |     +--rw max-occupancy-level? uint8
    |     +--rw mtu uint16
    |     +--rw steering-constraints
    |     +--rw path-constraints
    |     +--rw service-function
    +--rw status
    | +--rw admin-enabled? boolean
    | +--ro oper-status? operational-type
    +--rw ns-endpoints
    | +--rw ns-endpoint* [ep-id]
    |   +--rw ep-id string
    |   +--rw ep-description? string
    |   +--rw location
    |     | +--rw altitude? int64
    |     | +--rw latitude? decimal64
    |     | +--rw longitude? decimal64
    |   +--rw node-id? string
    |   +--rw ep-ip? inet:ip-address
    |   +--rw ns-match-criteria
    |     | +--rw ns-match-criterion* [index]
    |     |   +--rw index uint32

```



```

|         |--rw match-type?
|         |         identityref
|         |--rw values* [index]
|         |         |--rw index      uint8
|         |         |--rw value?    string
|         |--rw target-ns-connection-group-id?    leafref
|--rw ep-peering
|   |--rw protocol* [protocol-type]
|   |--rw protocol-type    identityref
|   |--rw attribute* [index]
|   |         |--rw index          uint8
|   |         |--rw attribute-description?    string
|   |         |--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-address?
|   |         inet:ip-address
|   |--rw network-access-tp-ip-prefix-length?    uint8
|   |--rw network-access-qos-policy-name?
|   |         string
|   |--rw mtu
|   |         uint16
|   |--rw network-access-tags
|   |         |--rw network-access-tag* [index]
|   |         |         |--rw index          uint32
|   |         |         |--rw network-access-tag-type?
|   |         |         |         identityref
|   |         |         |--rw network-access-tag-value?    string
|--rw ns-match-criteria
|   |--rw ns-match-criterion* [index]
|   |         |--rw index          uint32
|   |         |--rw match-type?
|   |         |         identityref
|   |         |--rw values* [index]
|   |         |         |--rw index      uint8
|   |         |         |--rw value?    string
|   |         |--rw target-ns-connection-group-id?    leafref
|--rw ep-peering
|   |--rw protocol* [protocol-type]
|   |--rw protocol-type    identityref

```

```

|         |         |         +--rw attribute* [index]
|         |         |         +--rw index                               uint8
|         |         |         +--rw attribute-description?         string
|         |         |         +--rw value?                         string
|         |         +--rw incoming-rate-limits
|         |         |         +--rw cir?      uint64
|         |         |         +--rw cbs?      uint64
|         |         |         +--rw eir?      uint64
|         |         |         +--rw ebs?      uint64
|         |         |         +--rw pir?      uint64
|         |         |         +--rw pbs?      uint64
|         |         +--rw outgoing-rate-limits
|         |         |         +--rw cir?      uint64
|         |         |         +--rw cbs?      uint64
|         |         |         +--rw eir?      uint64
|         |         |         +--rw ebs?      uint64
|         |         |         +--rw pir?      uint64
|         |         |         +--rw pbs?      uint64
|         +--rw incoming-rate-limits
|         |         +--rw cir?      uint64
|         |         +--rw cbs?      uint64
|         |         +--rw eir?      uint64
|         |         +--rw ebs?      uint64
|         |         +--rw pir?      uint64
|         |         +--rw pbs?      uint64
|         +--rw outgoing-rate-limits
|         |         +--rw cir?      uint64
|         |         +--rw cbs?      uint64
|         |         +--rw eir?      uint64
|         |         +--rw ebs?      uint64
|         |         +--rw pir?      uint64
|         |         +--rw pbs?      uint64
|         +--rw 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-connection-groups
+--rw ns-connection-group* [ns-connection-group-id]
+--rw ns-connection-group-id          string
+--rw (ns-slo-sle-policy)?
| +--:(standard)
| | +--rw slo-sle-template?          leafref
| +--:(custom)

```

```

|      +--rw slo-sle-policy
|      |      +--rw policy-description?      string
|      |      +--rw ns-metric-bounds
|      |      |      +--rw ns-metric-bound* [metric-type]
|      |      |      |      +--rw metric-type      identityref
|      |      |      |      +--rw metric-unit      string
|      |      |      |      +--rw value-description? string
|      |      |      |      +--rw bound?          uint64
|      |      +--rw security*                identityref
|      |      +--rw isolation?                identityref
|      |      +--rw max-occupancy-level?     uint8
|      |      +--rw mtu                      uint16
|      |      +--rw steering-constraints
|      |      +--rw path-constraints
|      |      +--rw service-function
+--rw ns-connection* [ns-connection-id]
|  +--rw ns-connection-id          uint32
|  +--rw ns-connectivity-type?     identityref
|  +--rw src-nse*                  leafref
|  +--rw dest-nse*                 leafref
|  +--rw (ns-slo-sle-policy)?
|  |  +--:(standard)
|  |  |  +--rw slo-sle-template?     leafref
|  |  +--:(custom)
|  |  |  +--rw slo-sle-policy
|  |  |  |  +--rw policy-description?      string
|  |  |  |  +--rw ns-metric-bounds
|  |  |  |  |  +--rw ns-metric-bound* [metric-type]
|  |  |  |  |  |  +--rw metric-type      identityref
|  |  |  |  |  |  +--rw metric-unit      string
|  |  |  |  |  |  +--rw value-description? string
|  |  |  |  |  |  +--rw bound?          uint64
|  |  |  |  +--rw security*                identityref
|  |  |  |  +--rw isolation?                identityref
|  |  |  |  +--rw max-occupancy-level?     uint8
|  |  |  |  +--rw mtu                      uint16
|  |  |  |  +--rw steering-constraints
|  |  |  |  +--rw path-constraints
|  |  |  |  +--rw service-function
+--ro ns-connection-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 ns-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
```

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:

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

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

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

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

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

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

*"ns-connection-groups": Abstracts the connections between NSEs.

6.1. IETF Network Slice Connectivity

Based on the customer's traffic requirements, an IETF Network Slice connectivity type could be point-to-point (P2P), point-to-multipoint (P2MP), multipoint-to-point (MP2P), multipoint-to-multipoint (MP2MP) or a combination of these types.

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

Apart from the per-connection SLO and SLE, slice traffic is usually managed by combining similar types of traffic. For example, some connections for video services require high bandwidth, and some connections for voice over IP request low latency and reliability.

"ns-connect-group" is thus defined to treat each type as a class with per-connection-group SLO and SLE.

6.2. IETF Network Slice SLO and SLE Policy

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

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

The policy can apply to per-network slice, per-connection group "ns-connection group", or per-connection "ns-connection".

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

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

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

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

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

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

"ns-slo-two-way-delay-variation": Indicates the jitter constraint of the slice maximum permissible delay variation, and is measured

by the difference in the two-way latency between sequential packets in a flow.

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

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

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

The following common SLEs are defined:

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

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

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

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

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

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


```

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

```

6.3. IETF Network Slice Endpoint (NSE)

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

An IETF Network Slice Endpoint has several characteristics:

- *"ep-id": Uniquely identifies the NSE within Network Slice Controller (NSC). The identifier is a string that allows any encoding for the local administration of the IETF Network Slice.
- *"location": Indicates NSE location information that facilitates NSC easy identification of a NSE.
- *"node-id": The NSE node information facilitates NSC with easy identification of a NSE.
- *"ep-ip": The NSE IP information facilitates NSC with easy identification of a NSE.
- *"ns-match-criteria": Defines matching policies for network slice traffic to apply on a given NSE.
- *"ep-network-access-points": Specifies the list of the interfaces attached to an edge device of the IETF Network Slice by which the customer traffic is received. This is an optional NSE attribute.

When a NSE has multiple interfaces attached and the NSC needs NSE interface-specific attributes, each "ep-network-access-point" can specify attributes such as interface specific IP address, MTU, etc.

*"incoming-rate-limits" and "outgoing-rate-limits": Set the rate-limiting policies to apply on a given NSE, including ingress and egress traffic to ensure access security. When applied in the incoming direction, the rate-limit is applicable to the traffic from the NSE to the IETF scope Network that passes through the external interface. When Bandwidth is applied to the outgoing direction, it is applied to the traffic from the IETF Network to the NSE of that particular NS. If an NSE has multiple AC, the "rate limit" of "ep-network-access-point" can be set to an AC specific value, but the rate cannot exceed the "rate limit" of the NSE. If a NSE only contains a single AC, then the "rate-limit" of "ep-network-access-point" is the same with the NSE "rate-limit". The definition refers to [[RFC7640](#)].

*"ep-peering": Specifies the protocol for a NSE for exchanging control-plane information, e.g. L1 signaling protocol or L3 routing protocols, etc.

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

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

*Physical interface: Indicates all the traffic received from the interface belongs to the IETF Network Slice.

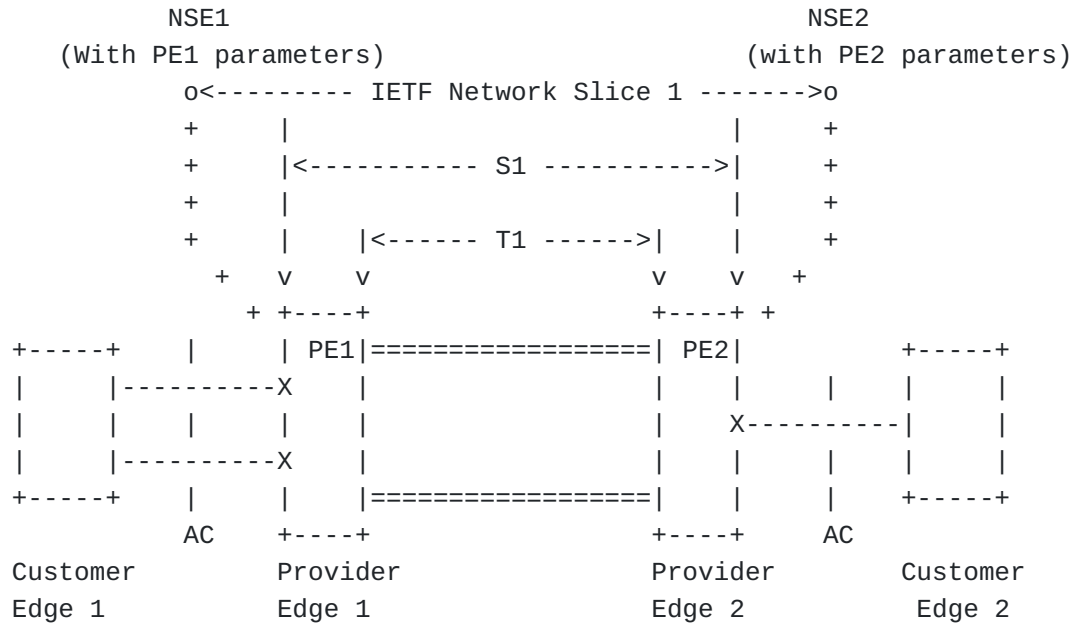
*Logical interface: For example, a given VLAN ID is used to identify an IETF Network Slice.

*Encapsulation in the traffic header: For example, a source IP address is used to identify an IETF Network Slice.

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

*NSE with PE parameters example: As shown in [Figure 4](#) , customer of the IETF network slice would like to connect two NSEs to satisfy specific service, e.g., Network wholesale services. In this case, the IETF network slice endpoints are mapped to physical interfaces of PE nodes. The IETF network slice controller (NSC) uses 'node-id' (PE device ID), 'ep-network-

access-points' (Two PE interfaces) to map the interfaces and corresponding services/tunnels/paths.

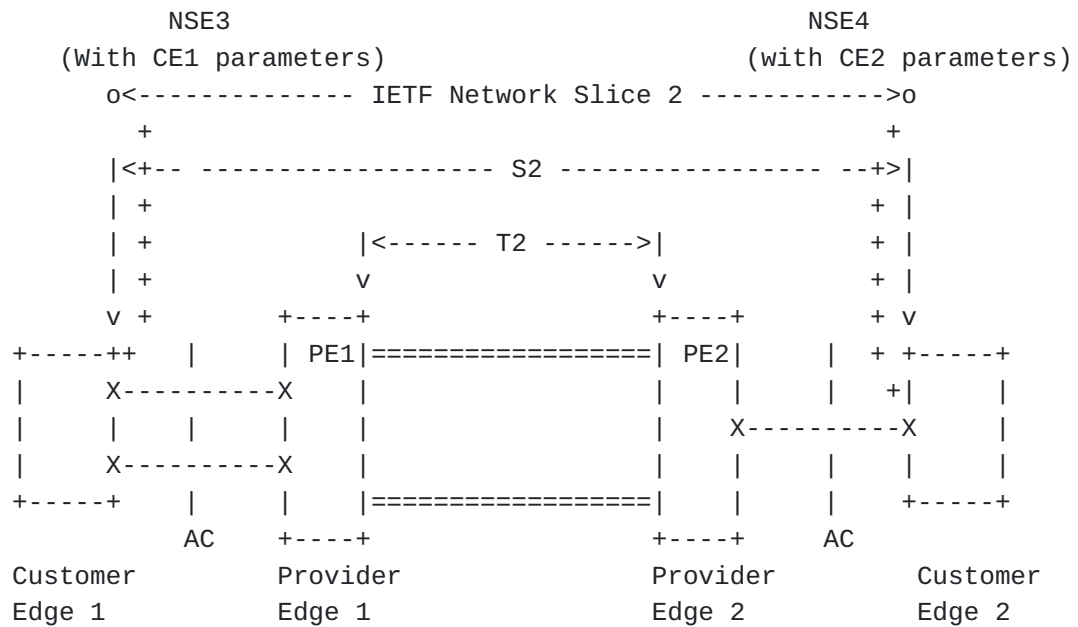


Legend:

- 0: Representation of the IETF network slice endpoints (NSE)
- +: Mapping of NES to PE or CE-PE interfaces
- X: Physical interfaces used for realization of IETF network slice
- S1: L0/L1/L2/L3 services used for realization of IETF network slice
- T1: Tunnels used for realization of IETF network slice

Figure 4

*NSE with CE parameters example: As shown in [Figure 5](#) , customer of the IETF network slice would like to connect two NSEs to provide connectivity between transport portion of 5G RAN to 5G Core network functions. In this scenario, the IETF network slice controller (NSC) uses 'node-id' (CE device ID) , 'ep-ip' (CE tunnel endpoint IP), 'network-slice-match-criteria' (VLAN interface), 'ep-network-access-points' (Two nexthop interfaces) to retrieve the corresponding CEs, ACs, or PEs, and further map to services/tunnels/paths.



Legend:

O: Representation of the IETF network slice endpoints (NSE)

+: Mapping of NSE to CE or CE-PE interfaces

X: Physical interfaces used for realization of IETF network slice

S2: L0/L1/L2/L3 services used for realization of IETF network slice

T2: Tunnels used for realization of IETF network slice

Figure 5

Note: The model needs to be optimized for better extension of other protocols or AC technologies.

7. IETF Network Slice Monitoring

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

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

*Per NS connection: specified in 'ns-connection-monitoring' under the "ns-connection".

*Per NS Endpoint: specified in 'ep-monitoring' under the "ns-endpoint".

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

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

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

Note: More critical events affecting service delivery need to be added.

8. IETF Network Slice Service Module

The "ietf-network-slice" module uses types defined in [[RFC6991](#)] and [[RFC8776](#)], and [[RFC7640](#)].

<CODE BEGINS> file "ietf-network-slice@2022-03-04.yang"

```
module ietf-network-slice {
  yang-version 1.1;
  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-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
             <lane.wubo@huawei.com>
     Editor: Dhruv Dhody
             <dhruv.ietf@gmail.com>
     Editor: Reza Rokui
             <reza.rokui@nokia.com>
     Editor: Tarek Saad
             <tasaad@juniper.net>
     Author: Liuyan Han
             <hanliuyan@chinamobile.com>";

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

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

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject
    to the license terms contained in, the Revised BSD License
```

set forth in Section 4.c of the IETF Trust's Legal Provisions
Relating to IETF Documents
(<https://trustee.ietf.org/license-info>).

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

```
revision 2022-03-04 {
  description
    "initial version.";
  reference
    "RFC XXXX: A Yang Data Model for IETF Network Slice Operation";
}

/* Features */
/* Identities */

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

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

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

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

identity network-access-tag-type {
  description
    "Base identity for the network access tag type.";
}

identity network-access-tag-vlan-id {
  base network-access-tag-type;
  description
    "The network access interface VLAN ID tag type.";
}
```

```

identity network-access-tag-ip-mask {
    base network-access-tag-type;
    description
        "The network access tag IP mask.";
}

identity network-access-tag-opaque {
    base network-access-tag-type;
    description
        "The network access opaque tag type.";
}

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

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

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

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

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

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

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

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

identity custom {
    base ns-connectivity-type;
    description
        "Identity of a custom NS topology where Hubs can act as

```

```

        Spoke for certain parts of the network or Spokes as Hubs.";
    }

identity endpoint-role {
    description
        "Base identity of a NSE role in an IETF Network Slice topology.";
}

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

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

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

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

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

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

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

```

```

        of an IETF Network Slice.";
    }

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

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

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

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

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

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

```

```

}

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

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

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

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

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

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

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

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

```

```

/*
 * Identity for availability-type
 */

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

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

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

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

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

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

/* typedef */

typedef operational-type {
    type enumeration {
        enum up {
            value 0;
            description
                "Operational status UP.";
        }
    }
}

```

```

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

typedef ns-monitoring-type {
    type enumeration {
        enum one-way {
            description
                "Represents one-way 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 "index";
            description
                "List of the IETF Network Slice traffic match criteria.";
            leaf index {
                type uint32;
                description
                    "The entry index.";
            }
            leaf match-type {
                type identityref {
                    base ns-match-type;
                }
                description
                    "Identifies an entry in the list of the IETF Network Slice
                    match criteria.";
            }
            list values {
                key "index";
                description
                    "List of match criteria values.";
                leaf index {
                    type uint8;
                    description
                        "Index of an entry in the list.";
                }
                leaf value {
                    type string;
                    description
                        "Describes the IETF Network Slice match criteria, e.g.
                        IP address, VLAN, etc.";
                }
            }
        }
    }
    leaf target-ns-connection-group-id {
        type leafref {
            path "/network-slices/network-slice"
                + "/ns-connection-groups/ns-connection-group"
                + "/ns-connection-group-id";
        }
    }
}

```

```

    }
    description
        "reference to a Network Slice connection group.";
    }
}
}
}

grouping ns-sles {
    description
        "Indirectly Measurable Objectives of a IETF Network
        Slice.";
    leaf-list security {
        type identityref {
            base ns-security-type;
        }
        description
            "The IETF Network Slice security SLE(s)";
    }
    leaf isolation {
        type identityref {
            base ns-isolation-type;
        }
        default "ns-isolation-shared";
        description
            "The IETF Network Slice isolation SLE requirement.";
    }
    leaf max-occupancy-level {
        type uint8 {
            range "1..100";
        }
        description
            "The maximal occupancy level specifies the number of flows to
            be admitted.";
    }
    leaf mtu {
        type uint16;
        units "bytes";
        mandatory true;
        description
            "The MTU specifies the maximum length in octets of data
            packets that can be transmitted by the NS. The value needs
            to be less than or equal to the minimum MTU value of
            all 'ep-network-access-points' in the NSEs of the NS.";
    }
    container steering-constraints {
        description
            "Container for the policy of steering constraints
            applicable to IETF Network Slice.";
    }
}

```



```

    container path-constraints {
        description
            "Container for the policy of path constraints
            applicable to IETF Network Slice.";
    }
    container service-function {
        description
            "Container for the policy of service function
            applicable to IETF Network Slice.";
    }
}

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

```

```

        specific metric-type.";
    }
}
}
}

grouping ep-peering {
    description
        "A grouping for the IETF Network Slice Endpoint peering.";
    container ep-peering {
        description
            "Describes NSE peering attributes.";
        list protocol {
            key "protocol-type";
            description
                "List of the NSE peering protocol.";
            leaf protocol-type {
                type identityref {
                    base peering-protocol-type;
                }
                description
                    "Identifies an entry in the list of NSE peering
                    protocol type.";
            }
            list attribute {
                key "index";
                description
                    "List of protocol attribute.";
                leaf index {
                    type uint8;
                    description
                        "Index of an entry in the list.";
                }
                leaf attribute-description {
                    type string;
                    description
                        "The description of the attribute.";
                }
                leaf value {
                    type string;
                    description
                        "Describes the value of protocol attribute, e.g.
                        nexthop address, peer address, etc.";
                }
            }
        }
    }
}
}
}
}

```

```

grouping ep-network-access-points {
  description
    "Grouping for the endpoint network access definition.";
  container ep-network-access-points {
    description
      "List of network access points.";
    list ep-network-access-point {
      key "network-access-id";
      description
        "The IETF Network Slice network access points
          related parameters.";
      leaf network-access-id {
        type string;
        description
          "Uniquely identifier a network access point.";
      }
      leaf network-access-description {
        type string;
        description
          "The network access point description.";
      }
      leaf network-access-node-id {
        type string;
        description
          "The network access point node ID in the case of
            multi-homing.";
      }
      leaf network-access-tp-id {
        type string;
        description
          "The termination port ID of the EP network access
            point.";
      }
      leaf network-access-tp-ip-address {
        type inet:ip-address;
        description
          "The IP address of the EP network access point.";
      }
      leaf network-access-tp-ip-prefix-length {
        type uint8;
        description
          "The subnet prefix length expressed in bits.";
      }
      leaf network-access-qos-policy-name {
        type string;
        description
          "The name of the QoS policy that is applied to the
            network access point. The name can reference a QoS
            profile that is pre-provisioned on the device.";
      }
    }
  }
}

```

```

    }
    leaf mtu {
        type uint16;
        units "bytes";
        mandatory true;
        description
            "Maximum size in octets of a data packet that
             can traverse a NSE network access point.";
    }
    container network-access-tags {
        description
            "Container for the network access tags.";
        list network-access-tag {
            key "index";
            description
                "The network access point tags list.";
            leaf index {
                type uint32;
                description
                    "The entry index.";
            }
            leaf network-access-tag-type {
                type identityref {
                    base network-access-tag-type;
                }
                description
                    "The network access point tag type.";
            }
            leaf network-access-tag-value {
                type string;
                description
                    "The network access point tag value.";
            }
        }
    }
}
/* Per ep-network-access-point rate limits */
uses ns-match-criteria;
uses ep-peering;
uses ns-rate-limit;
}
}

grouping ep-monitoring-metrics {
    description
        "Grouping for the NS endpoint monitoring metrics.";
    container ep-monitoring {
        config false;
        description

```

```

    "Container for NS endpoint monitoring metrics.";
  leaf incoming-utilized-bandwidth {
    type te-types:te-bandwidth;
    description
      "Incoming bandwidth utilization at an endpoint.";
  }
  leaf incoming-bw-utilization {
    type decimal64 {
      fraction-digits 5;
      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 ns-connection-monitoring-metrics {
  description
    "Grouping for NS connection monitoring metrics.";
  uses te-packet-types:one-way-performance-metrics-packet;
  uses te-packet-types:two-way-performance-metrics-packet;
}

grouping geolocation-container {
  description
    "A grouping containing a GPS location.";
  container location {
    description
      "A container containing a GPS location.";
  }
}

```

```

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

// geolocation-container

grouping bw-rate-limits {
    description
        "Bandwidth rate limits grouping.";
    reference
        "RFC 7640: Traffic Management Benchmarking";
    leaf cir {
        type uint64;
        units "bps";
        description
            "Committed Information Rate. The maximum number of bits
            that a port can receive or send during one-second over an
            interface.";
    }
    leaf cbs {
        type uint64;
        units "bytes";
        description
            "Committed Burst Size. CBS controls the bursty nature
            of the traffic. Traffic that does not use the configured
            CIR accumulates credits until the credits reach the
            configured CBS.";
    }
}

```

```

}
leaf eir {
    type uint64;
    units "bps";
    description
        "Excess Information Rate, i.e., excess frame delivery
        allowed not subject to SLA. The traffic rate can be
        limited by EIR.";
}
leaf ebs {
    type uint64;
    units "bytes";
    description
        "Excess Burst Size. The bandwidth available for burst
        traffic from the EBS is subject to the amount of
        bandwidth that is accumulated during periods when
        traffic allocated by the EIR policy is not used.";
}
leaf pir {
    type uint64;
    units "bps";
    description
        "Peak Information Rate, i.e., maximum frame delivery
        allowed. It is equal to or less than sum of CIR and EIR.";
}
leaf pbs {
    type uint64;
    units "bytes";
    description
        "Peak Burst Size.";
}
}

grouping ns-rate-limit {
    description
        "The rate limits grouping.";
    container incoming-rate-limits {
        description
            "Container for the asymmetric traffic control.";
        uses bw-rate-limits;
    }
    container outgoing-rate-limits {
        description
            "The rate-limit imposed on outgoing traffic.";
        uses bw-rate-limits;
    }
}

grouping endpoint {

```

```

description
    "IETF Network Slice endpoint related information";
leaf ep-id {
    type string;
    description
        "Unique identifier for the referred IETF Network
        Slice endpoint.";
}
leaf ep-description {
    type string;
    description
        "Give more description of the Network Slice endpoint.";
}
uses geolocation-container;
leaf node-id {
    type string;
    description
        "Uniquely identifies an edge node within the IETF slice
        network.";
}
leaf ep-ip {
    type inet:ip-address;
    description
        "The IP address of the endpoint.";
}
uses ns-match-criteria;
uses ep-peering;
uses ep-network-access-points;
uses ns-rate-limit;
/* Per NSE rate limits */
uses status-params;
uses ep-monitoring-metrics;
}

//ns-endpoint

grouping ns-connection {
    description
        "The network slice connection grouping.";
    list ns-connection {
        key "ns-connection-id";
        description
            "List of Network Slice connections.";
        leaf ns-connection-id {
            type uint32;
            description
                "The Network Slice connection identifier.";
        }
        leaf ns-connectivity-type {

```



```

    type identityref {
        base ns-connectivity-type;
    }
    default "point-to-point";
    description
        "Network Slice connection construct type.";
}
leaf-list src-nse {
    type leafref {
        path "/network-slices/network-slice"
            + "/ns-endpoints/ns-endpoint/ep-id";
    }
    description
        "reference to source Network Slice endpoint.";
}
leaf-list dest-nse {
    type leafref {
        path "/network-slices/network-slice"
            + "/ns-endpoints/ns-endpoint/ep-id";
    }
    description
        "reference to source Network Slice endpoint.";
}
uses ns-slo-sle-policy;
/* Per connection ns-slo-sle-policy overrides
 * the per network slice ns-slo-sle-policy.
 */
container ns-connection-monitoring {
    config false;
    description
        "SLO status Per NS connection.";
    uses ns-connection-monitoring-metrics;
}
}
}

//ns-connection

grouping ns-connection-group {
    description
        "The Network Slice connection group is described in this
        container.";
    leaf ns-connection-group-id {
        type string;
        description
            "The Network Slice connection group identifier.";
    }
}
uses ns-slo-sle-policy;
uses ns-connection;

```

```

/* Per connection ns-slo-sle-policy overrides
 * the per network slice ns-slo-sle-policy.
 */
container ns-connection-group-monitoring {
    config false;
    description
        "SLO status Per NS connection.";
    uses ns-connection-monitoring-metrics;
}
}

//ns-connection-group

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

/* Configuration data nodes */

grouping ns-slo-sle-policy {
    description
        "Network Slice policy grouping.";
    choice ns-slo-sle-policy {
        description
            "Choice for SLO and SLE policy template.
            Can be standard template or customized template.";
    }
}

```

```

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

container network-slices {
  description
    "Contains a list of IETF network slice";
  uses slice-template;
  list network-slice {
    key "ns-id";
    description
      "A network-slice is identified by a ns-id.";
    leaf ns-id {
      type string;
      description
        "A unique network-slice identifier across an IETF NSC.";
    }
    leaf ns-description {
      type string;
      description
        "Give more description of the network slice.";
    }
    container ns-tags {

```

```

description
  "Container for the list of IETF Network Slice tags.";
list ns-tag {
  key "index";
  description
    "IETF Network Slice tag list.";
  leaf index {
    type uint32;
    description
      "The entry index.";
  }
  leaf ns-tag-type {
    type identityref {
      base ns-tag-type;
    }
    description
      "The IETF Network Slice tag type.";
  }
  leaf ns-tag-value {
    type string;
    description
      "The IETF Network Slice tag value.";
  }
}
}
uses ns-slo-sle-policy;
uses status-params;
container ns-endpoints {
  description
    "NS Endpoints.";
  list ns-endpoint {
    key "ep-id";
    uses endpoint;
    description
      "List of endpoints in this slice.";
  }
}
container ns-connection-groups {
  description
    "Contains NS connections group.";
  list ns-connection-group {
    key "ns-connection-group-id";
    description
      "List of Network Slice connections.";
    uses ns-connection-group;
  }
}
}
//ietf-network-slice list

```

```
}  
}
```

<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 Mohamed Boucadair, John Mullooly, Kenichi Ogaki, Sergio Belotti, Qin Wu, Susan Hares, Eric Grey, and many others for their helpful comments and suggestions.

12. Contributors

The following authors contributed significantly to this document:

Luis M. Contreras
Telefonica
Spain

Email: luismiguel.contrerasmurillo@telefonica.com

13. References

13.1. Normative References

- [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>>.

13.2. Informative References

[I-D.geng-teas-network-slice-mapping]

Geng, X., Dong, J., Pang, R., Han, L., Rokui, R., Niwa, T., Jin, J., Liu, C., and N. Nageshar, "5G End-to-end Network Slice Mapping from the view of Transport Network", Work in Progress, Internet-Draft, draft-geng-teas-network-slice-mapping-04, 25 October 2021, <<https://www.ietf.org/archive/id/draft-geng-teas-network-slice-mapping-04.txt>>.

[I-D.ietf-opsawg-vpn-common] Barguil, S., Dios, O. G. D., Boucadair, M., and Q. Wu, "A Common YANG Data Model for Layer 2 and Layer 3 VPNs", Work in Progress, Internet-Draft, draft-ietf-opsawg-vpn-common-12, 29 September 2021, <<https://www.ietf.org/archive/id/draft-ietf-opsawg-vpn-common-12.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 VN Operation", Work in Progress, Internet-Draft, draft-ietf-teas-actn-vn-yang-13, 23 October 2021, <<https://www.ietf.org/archive/id/draft-ietf-teas-actn-vn-yang-13.txt>>.

[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", Work in Progress, Internet-Draft, draft-ietf-teas-ietf-network-slices-05, 25 October 2021, <<https://www.ietf.org/archive/id/draft-ietf-teas-ietf-network-slices-05.txt>>.

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

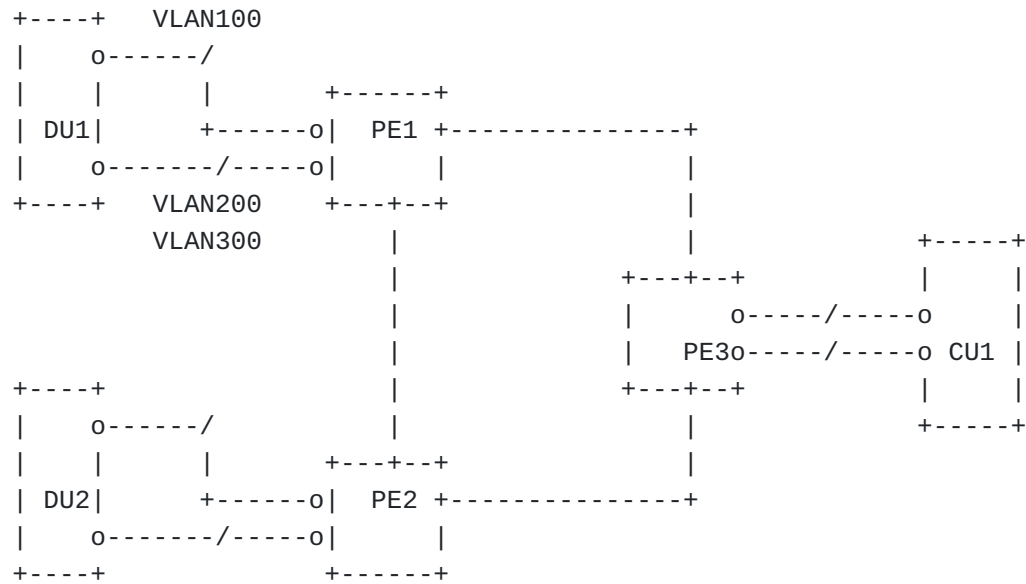
Liu, X., Tantsura, J., Bryskin, I., Contreras, L. M., Wu, Q., Belotti, S., and R. Rokui, "IETF Network Slice YANG Data Model", Work in Progress, Internet-Draft, draft-liu-teas-transport-network-slice-yang-04, 9 July 2021, <<https://www.ietf.org/archive/id/draft-liu-teas-transport-network-slice-yang-04.txt>>.

[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. IETF Network Slice Service Model Usage Example

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

*IETF Network Slice 1 on PE1, PE2, and PE3, with two NS-connection-groups



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

Host: example.com

Content-Type: application/yang-data+json

```
{
  "ietf-network-slice:network-slices": {
    "network-slice": [
      {
        "ns-id": "NS1",
        "ns-description": "URLLC",
        "ns-tags": {
          "ns-tag": [
            {
              "index": 1,
              "ns-tag-type": "ns-tag-customer",
              "ns-tag-value": "F00"
            },
            {
              "index": 2,
              "ns-tag-type": "ns-tag-customer",
              "ns-tag-value": "BAR"
            },
            {
              "index": 3,
              "ns-tag-type": "ns-tag-service",
              "ns-tag-value": "L2"
            }
          ]
        },
        "status": {
          "admin-enabled": true,
          "oper-status": "up"
        },
        "ns-endpoints": {
          "ns-endpoint": [
            {
              "ep-id": "DU1",
              "ep-description": "DU1 at location X",
              "ep-ip": "1.1.1.1",
              "ns-match-criteria": {
                "ns-match-criterion": [
                  {
                    "index": 0,
                    "match-type": "ns-vlan-match",
                    "values": [
                      {
                        "index": 1,
                        "value": "VLAN-100"
                      }
                    ]
                  }
                ]
              }
            }
          ]
        }
      }
    ]
  }
}
```

```

        "target-ns-connection-group-id": "Matrix1"
    },
    {
        "index": 1,
        "match-type": "ns-vlan-match",
        "values": [
            {
                "index": 1,
                "value": "VLAN-200"
            },
            {
                "index": 2,
                "value": "VLAN-300"
            }
        ],
        "target-ns-connection-group-id": "Matrix2"
    }
]
},
"ep-network-access-points": {
    "ep-network-access-point": [
        {
            "network-access-id": "AC1-VRF100",
            "network-access-description": "VRF100 to PE1",
            "network-access-node-id": "PE1",
            "network-access-tp-id": "1",
            "network-access-tp-ip-address": "192.0.1.2",
            "network-access-tp-ip-prefix-length": 24,
            "network-access-qos-policy-name": "QoS-Gold",
            "network-access-tags": {
                "network-access-tag": [
                    {
                        "index": 1,
                        "network-access-tag-type": "network-access-tag",
                        "network-access-tag-value": "100"
                    },
                    {
                        "index": 2,
                        "network-access-tag-type": "network-access-tag",
                        "network-access-tag-value": "F00"
                    }
                ]
            }
        }
    ],
    "ep-peering": {
        "protocol": [
            {
                "protocol-type": "peering-protocol-bgp",
                "attribute": [
                    {

```

```

        "index": 1,
        "value": "COLOR:10"
    },
    {
        "index": 2,
        "value": "RT:20"
    },
    {
        "index": 3,
        "value": "RT:30"
    }
]
}
]
},
"incoming-rate-limits": {
    "cir": "1000000",
    "cbs": "1000",
    "pir": "5000000",
    "pbs": "1000"
}
},
{
    "network-access-id": "AC2-VRF200",
    "network-access-description": "VRF200 to PE1",
    "network-access-node-id": "PE1",
    "network-access-tp-id": "2",
    "network-access-tp-ip-address": "192.0.2.2",
    "network-access-tp-ip-prefix-length": 24,
    "network-access-qos-policy-name": "QoS-Gold",
    "network-access-tags": {
        "network-access-tag": [
            {
                "index": 1,
                "network-access-tag-type": "network-access-tag",
                "network-access-tag-value": "100"
            },
            {
                "index": 2,
                "network-access-tag-type": "network-access-tag",
                "network-access-tag-value": "FOO"
            }
        ]
    }
},
"ep-peering": {
    "protocol": [
        {
            "protocol-type": "peering-protocol-bgp",
            "attribute": [

```

```

        {
            "index": 1,
            "value": "COLOR:10"
        },
        {
            "index": 2,
            "value": "RT:20"
        },
        {
            "index": 3,
            "value": "RT:30"
        }
    ]
}
]
},
"incoming-rate-limits": {
    "cir": "1000000",
    "cbs": "1000",
    "pir": "5000000",
    "pbs": "1000"
}
}
]
},
{
    "ep-id": "DU2",
    "ep-description": "DU2 at location Y",
    "ep-ip": "2.2.2.2",
    "ep-network-access-points": {
        "ep-network-access-point": [
            {
                "network-access-id": "AC1-VRF100",
                "network-access-description": "VRF100 to PE2",
                "network-access-node-id": "PE2",
                "network-access-tp-id": "1",
                "network-access-tp-ip-address": "192.1.1.2",
                "network-access-tp-ip-prefix-length": 24,
                "network-access-qos-policy-name": "QoS-Gold",
                "ep-peering": {
                    "protocol": [
                        {
                            "protocol-type": "peering-protocol-bgp",
                            "attribute": [
                                {
                                    "index": 1,
                                    "value": "COLOR:10"
                                },

```

```

        {
            "index": 2,
            "value": "RT:20"
        },
        {
            "index": 3,
            "value": "RT:30"
        }
    ]
}
]
},
"incoming-rate-limits": {
    "cir": "1000000",
    "cbs": "1000",
    "pir": "5000000",
    "pbs": "1000"
}
},
{
    "network-access-id": "AC2-VRF200",
    "network-access-description": "VRF200 to PE1",
    "network-access-node-id": "PE2",
    "network-access-tp-id": "2",
    "network-access-tp-ip-address": "192.1.2.2",
    "network-access-tp-ip-prefix-length": 24,
    "network-access-qos-policy-name": "QoS-Gold",
    "ep-peering": {
        "protocol": [
            {
                "protocol-type": "peering-protocol-bgp",
                "attribute": [
                    {
                        "index": 1,
                        "value": "COLOR:10"
                    },
                    {
                        "index": 2,
                        "value": "RT:20"
                    },
                    {
                        "index": 3,
                        "value": "RT:30"
                    }
                ]
            }
        ]
    }
}
],
},
"incoming-rate-limits": {

```

```

        "cir": "1000000",
        "cbs": "1000",
        "pir": "5000000",
        "pbs": "1000"
    }
}
]
}
},
{
    "ep-id": "CU1",
    "ep-description": "CU1 at location Z",
    "ep-ip": "3.3.3.3",
    "ep-network-access-points": {
        "ep-network-access-point": [
            {
                "network-access-id": "AC1-VRF100",
                "network-access-description": "VRF100 to PE2",
                "network-access-node-id": "PE3",
                "network-access-tp-id": "1",
                "network-access-tp-ip-address": "192.2.1.2",
                "network-access-tp-ip-prefix-length": 24,
                "network-access-qos-policy-name": "QoS-Gold",
                "ep-peering": {
                    "protocol": [
                        {
                            "protocol-type": "peering-protocol-bgp",
                            "attribute": [
                                {
                                    "index": 1,
                                    "value": "COLOR:10"
                                },
                                {
                                    "index": 2,
                                    "value": "RT:20"
                                },
                                {
                                    "index": 3,
                                    "value": "RT:30"
                                }
                            ]
                        }
                    ]
                }
            }
        ]
    },
    "incoming-rate-limits": {
        "cir": "1000000",
        "cbs": "1000",
        "pir": "5000000",
        "pbs": "1000"
    }
}

```

```

    }
  },
  {
    "network-access-id": "AC2-VRF200",
    "network-access-description": "VRF200 to PE1",
    "network-access-node-id": "PE3",
    "network-access-tp-id": "2",
    "network-access-tp-ip-address": "192.2.2.2",
    "network-access-tp-ip-prefix-length": 24,
    "network-access-qos-policy-name": "QoS-Gold",
    "ep-peering": {
      "protocol": [
        {
          "protocol-type": "peering-protocol-bgp",
          "attribute": [
            {
              "index": 1,
              "value": "COLOR:10"
            },
            {
              "index": 2,
              "value": "RT:20"
            },
            {
              "index": 3,
              "value": "RT:30"
            }
          ]
        }
      ]
    },
    "incoming-rate-limits": {
      "cir": "1000000",
      "cbs": "1000",
      "pir": "5000000",
      "pbs": "1000"
    }
  }
]
},
"ns-connection-groups": {
  "ns-connection-group": [
    {
      "ns-connection-group-id": "Matrix1",
      "slo-sle-policy": {
        "policy-description": "URLLC-SLAs-Template1",

```



```

"ns-metric-bounds": {
  "ns-metric-bound": [
    {
      "metric-type": "ns-slo-shared-bandwidth",
      "metric-unit": "Gbps",
      "value-description": "Shared bandwidth for Matrix1",
      "bound": "15"
    },
    {
      "metric-type": "ns-slo-one-way-bandwidth",
      "metric-unit": "Gbps",
      "value-description": "One-way bandwidth for Matrix3",
      "bound": "10"
    },
    {
      "metric-type": "ns-slo-one-way-delay",
      "metric-unit": "msec",
      "value-description": "One-way delay for Matrix3 co
    },
    {
      "metric-type": "ns-slo-one-way-delay-variation",
      "metric-unit": "msec",
      "value-description": "One-way delay variation for
    }
  ]
}
},
"ns-connection": [
  {
    "ns-connection-id": 1,
    "src-nse": [
      "DU1"
    ],
    "dest-nse": [
      "CU1"
    ],
    "slo-sle-policy": {
      "ns-metric-bounds": {
        "ns-metric-bound": [
          {
            "metric-type": "ns-slo-one-way-delay",
            "metric-unit": "msec",
            "bound": "20"
          }
        ]
      }
    }
  }
],
{

```

```

        "ns-connection-id": 2,
        "src-nse": [
            "DU2"
        ],
        "dest-nse": [
            "CU1"
        ]
    }
]
},
{
    "ns-connection-group-id": "Matrix2",
    "slo-sle-template": "URLLC-SLAs-Template2",
    "ns-connection": [
        {
            "ns-connection-id": 1,
            "src-nse": [
                "DU1"
            ],
            "dest-nse": [
                "CU1"
            ]
        },
        {
            "ns-connection-id": 2,
            "src-nse": [
                "DU2"
            ],
            "dest-nse": [
                "CU1"
            ]
        }
    ]
}
]
},
{
    "ns-id": "NS2",
    "status": {
        "admin-enabled": true,
        "oper-status": "up"
    }
}
]
}
}

```

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

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

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

B.1. ACTN VN Model Augmentation

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

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

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

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

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

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

The Type 1 VN can be used to describe IETF Network Slice connection requirements. However, the Network Slice SLO and Network Slice Endpoint are not clearly defined and there's no direct equivalent. For example, the SLO requirement of the VN is defined through the

IETF TE Topologies YANG model, but the TE Topologies model is related to a specific implementation technology. Also, VN-AP does not define "network-slice-match-criteria" to specify a specific NSE belonging to an IETF Network Slice.

B.2. RFC8345 Augmentation Model

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

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

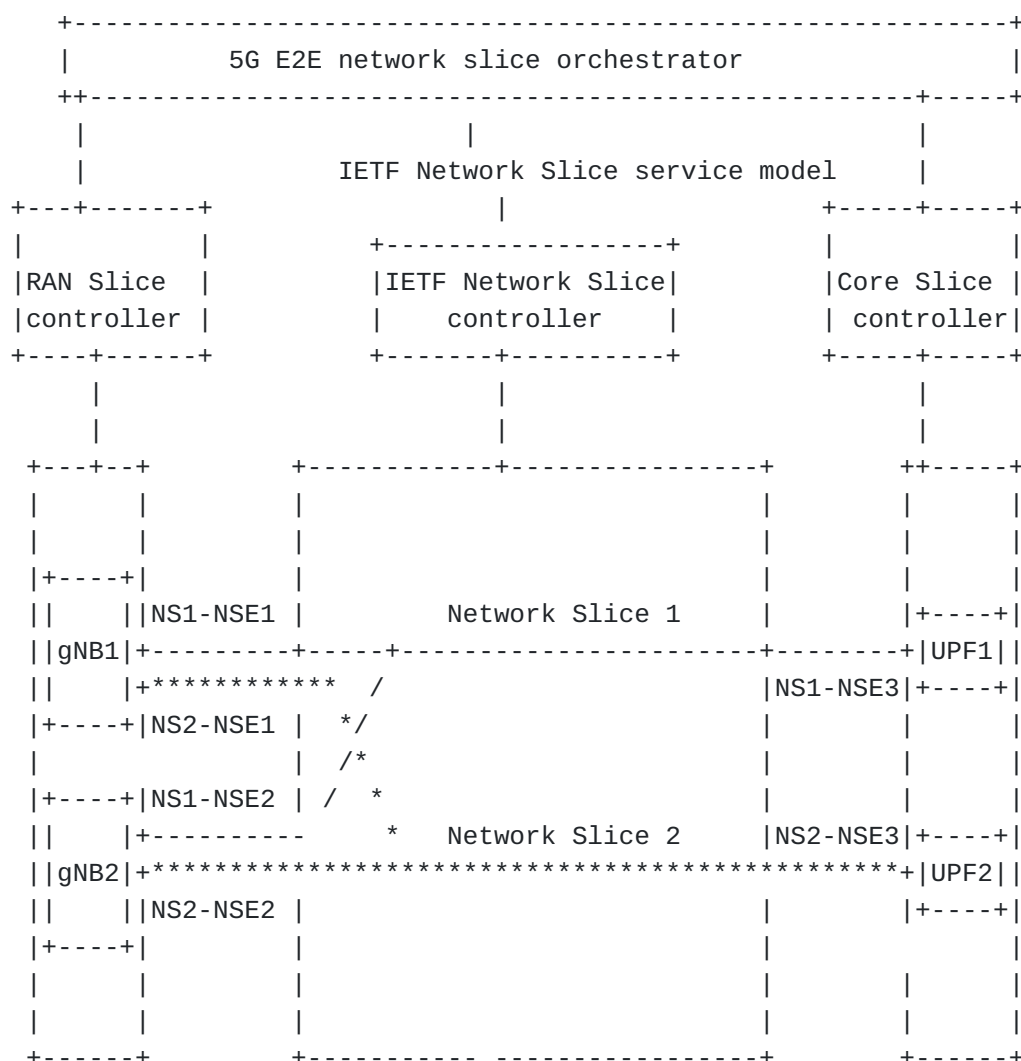
- *Network: a transport network list and an list of nodes contained in the network
- *Link: "links" list and "termination points" list describe how nodes in a network are connected to each other
- *Support network: vertical layering relationships between IETF Network Slice networks and underlay networks

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

Appendix C. Appendix B IETF Network Slice Match Criteria

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

defines two types of Network Slice interconnection and differentiation methods: by physical interface or by TNSII (Transport Network Slice Interworking Identifier). TNSII is a field in the packet header when different 5G wireless network slices are transported through a single physical interfaces of the IETF scoped Network. In the 5G scenario, "network-slice-match-criteria" refers to TNSII.



As shown in the figure, gNodeB 1 and gNodeB 2 use IP gNB1 and IP gNB2 to communicate with the IETF network, respectively. In addition, the traffic of NS1 and NS2 on gNodeB 1 and gNodeB 2 is transmitted through the same access links to the IETF slice network. The IETF slice network 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: lane.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
Juniper Networks

Email: tsaad@juniper.net

Liuyan Han
China Mobile

Email: hanliuyan@chinamobile.com