TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: 28 July 2026

X. Liu
Alef Edge
L.M. Contreras
Telefonica
S. Belotti
Nokia
A. Guo
Futurewei
I. Busi
Huawei
24 January 2026

IETF Network Slice Topology YANG Data Model
draft-ietf-teas-network-slice-topology-yang-03

Abstract

An RFC 9543 network slice customer may utilize intent-based topologies to express resource reservation intentions within the provider’s network. These customer-defined intent topologies allow customers to request shared resources for future connections that can be flexibly allocated and customized. Additionally, they provide an extensive level of control over underlay service paths within the network slice.

This document describes a YANG data model for expressing customer intent topologies which can be used to enhance the RFC 9543 Network Slice Services in specific use cases, such as Network wholesale scenarios, where both topology and connectivity intents need to be expressed.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Network service providers utilize topologies to convey controlled information about their networks, such as bandwidth availability and connectivity, with customers, to facilitates customer service requests. Customers can also define intent-based topologies to streamline their internal operations. When requesting provider support for such custom topologies, they are considered as customer intent topologies.

In the context of network slicing, customer intent topologies enables customers to express resource reservation preferences. These topologies allow flexible configuration and activation of network slices on demand. By providing full control over resource allocation timing and methods, customer intent topologies ensure that resources are consistently available. Moreover, the resources reserved via customer intent topologies can be shared across network slices created at different times or between different connectivity constructs within the same slice. Compared to network slices with dedicated full-mesh connectivity constructs between endpoints, network slices utilizing customer intent topologies can reduce overall resource requirements, offering significant economic benefits to the customer.

Consider a hub-and-spoke network slice scenario where multiple customer spoke sites dynamically connect to a central hub site, sharing available bandwidth. By designing a customer intent topology with two virtual nodes - one representing all the spoke sites and the other representing the hub site - connected via a shared link, we proactively reserve resources for the shared connection. This ensures that bandwidth is readily available whenever the customer requires it. In contrast, achieving equivalent bandwidth assurance through individual dedicated connectivity constructs would necessitate creating separate links between each spoke and the hub, which would lead to substantial bandwidth inefficiency.

Customer intent topology complements connectivity-based network slicing by providing customers a mechanism to specify additional underlay service paths to gain extensive control over specific or all connectivity constructs within the network slice, as outlined in [RFC9543].

A customer intent topology is defined within the customer’s context. It can include pure customer information or may also refer to network resources identifiable within the provider’s context. There is a minimum level of a-prior shared knowledge between the customer and the provider, and this is the same information needed to supported connectivity-based network slice services as described in [RFC9543]. The provider’s responsibility lies in understanding the customer intent topology request and translating that into suitable realization within their domain.

This document introduces a YANG data model, based on [RFC7950], for configuring customer intent topologies. The YANG model extends the existing data model from [RFC8345], allowing customers to express desired service-level objectives (SLOs) and service-level expectations (SLEs) across different elements within the customer intent topology.

The defined data model serves as an interface between customers and providers, enabling configurations and state retrievals for network slicing as a service. Customers can use this model to request or negotiate the creation of network slice instances. Additionally, they can incrementally adjust requirements for individual topology elements within the slice - for instance, adding or removing nodes or links, updating link bandwidth - and retrieve operational states. Leveraging other IETF mechanisms and data models, telemetry information can also be convey to the customer.

The YANG model encompasses constructs that are independent of specific technologies, accommodating network slicing across diverse layers (including IP/MPLS, MPLS-TP, OTN, and WDM optical). As a result, this model serves as a foundational framework upon which technology-specific network slicing models - such as [I-D.ietf-ccamp-yang-otn-slicing] - can be developed.

Section 3 of [I-D.ietf-teas-ns-controller-models] outlines that the use of customer intent topologies and resource reservation control is optional within network slicing. These features complement the data model defined in [I-D.ietf-teas-ietf-network-slice-nbi-yang].

The YANG data model in this document conforms to the Network Management Datastore Architecture (NMDA) [RFC8342].

1.1. Use Case Applicability

In Traffic Engineering (TE)-enabled networks like Layer-0/1 transport (OTN, MW, DWDM), customer intent topology is useful for routing RFC 9543 network slices across varied paths with TE constraints. Thus, most of the use cases for which this model target are transport oriented. Nonetheless, it is also relevant to non-transport networks like IP/MPLS, where customers may use intent topologies to influence the realization of network slices. These intents help build the logical view of the desired RFC 9543 Network Slice service (and its constituent parts), aiding providers in fulfilling slice requests and defining the service instantiation.

1.1.1. Use Case 1 : Multi-tenancy in Network Wholesaling

A typical use case in which the customer intent topology is essential is the wholesale multi-tenant case. Here, customer C may acquire a network slice from provider P and resell sub-slices to other customers/tenants. The creation of these sub-slices within C’s slice necessitates specifying a topology intent - reflecting the topology of C’s purchased slice - as a key input parameter.

1.1.2. Use Case 2 : Scoped Connectivity Constructs in Network Slicing

The current expression of slice requests leveraging on [I-D.ietf-teas-ietf-network-slice-nbi-yang] allows the customer to request distinct connectivity constructs as part of the same Network Resource Partition (NRP). The topology provided by the customer could imply different NRPs, instead.

As another example, a slice request leveraging [I-D.ietf-teas-ietf-network-slice-nbi-yang] without topology differentiation could result in all connectivity constructs being realized in the same manner on the same NRP, e.g. implementing all of them within the same VRF in a L3VPN. Using topological views can help providers infer differentiated realizations of some of the connectivity constructs, for instance, by implementing them on different VRFs. This approach can offer operational advantages, like limiting the necessary VRF reconfiguration to only those affected connectivity constructs when adding new nodes or SDPs.

Finally, by using customer intent topology it can be easier for the slice provider to infer different technologies for sets of connectivity constructs of every topology segment (e.g., IP/MPLS, optical, microwave, etc).

1.2. Terminologies and Notations

The following terminologies for describing network slices are defined in [RFC9543] and are not redefined herein.

* Network Slice (NS)

* Network Slice Customer

* Network Slice Service Provider

* Network Slice Controller (NSC)

* Network Resource Partition (NRP)

The following terms are defined and used in this document.

* Customer Intent Topology: A topology defined by the customer and provided as input to the network slice service provider (specifically, the Network Slice Controller or NSC). It represents the customer's desired network topology.

* Abstract Topology: A topology exposed to the customer by the network slice service provider prior to the creation of network slices. The provider may optionally uses an abstract topology to expose useful information, such as available resources to the customer, which can facilitate the build-up of customer intent topologies by the customer.

* NRP Topology: A topology internal to the NSC to facilitate the mapping of network slices to underlying network resources.

1.3. Tree Diagram

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

1.4. Prefixes in Data Node Names

In this document, names of data nodes and other data model objects are prefixed using the standard prefix associated with the corresponding YANG imported modules, as shown in Table 1.

Prefix	YANG Module	Reference
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]
nt	ietf-network-topology	[RFC8345]
nw	ietf-network-topology	[RFC8345]
tet	ietf-te-topology	[RFC8795]
ns-path	ietf-ns-underlay-path	RFC XXXX
ns-topo	ietf-ns-topo	RFC XXXX
ietf-nss	ietf-network-slice-service	RFC YYYY

Table 1: Prefixes and Corresponding YANG Modules

RFC Editor Note: Please replace XXXX with the RFC number assigned to this document. Please replace YYYY with the RFC number assigned to [I-D.ietf-teas-ietf-network-slice-nbi-yang]. Please remove this note.

2. Modeling Considerations

A network slice topology is a customer intent topology modeled as network topology defined in [RFC8345], with augmentations. A new network type "network-slice" is defined in this document.
When a network topology data instance contains the network-slice network type, it represents an instance of a network slice topology.

This data model augments the network topology model by incorporating intent-based Service-Level Objectives (SLOs) and Service-Level Expectations (SLEs). These apply to various components within the customer intent topology, including nodes, links, and termination points (TPs).

2.1. Relationship with Traffic Engineering (TE)-based Topology

The model defined in this document can be combined through multi-inheritance with other topology data models, such as Traffic Engineering (TE) topologies described in [RFC8795] or Optical Transport Network (OTN) topologies described in [I-D.ietf-ccamp-otn-topo-yang]. This flexibility allows the creation of technology-specific customer intent topologies tailored to specific network requirements.

2.2. Relationship with ACTN Virtual Network (VN)

The ACTN VN model, defined in [RFC9731], provides a self-consistent set of methods for expressing connectivity intents (Type 1 VN), optional path constraints and topology intents (Type 2 VN), using TE metrics and TE objective functions defined in [RFC8795]. Type 2 VN path constraints rely on Type 1 VN for expressing connectivity intents. See Appendix A for more details.

On the other hand, RFC9543 network slice services provide connectivity intents equivalent to Type 1 VN, using SLO and SLE attributes in a technology-agnostic manner not tied to TE technologies. This distinction is detailed in Appendix D of [I-D.ietf-teas-ietf-network-slice-nbi-yang].

The proposed models in this draft aim to deliver a solution equivalent to Type 2 VN to provide optional path constraints and topology intent within the context of RFC 9543 network slicing. These models complement the existing solution outlined in [I-D.ietf-teas-ietf-network-slice-nbi-yang], while ensuring consistent use of SLO and SLE attributes in a technology-agnostic manner to express customer intent.

In a nutshell:

* the data models, defined in this draft, are intended to be used when there is a need to extend, with more control over network resources allocation by the customer, the connectivity service intent, expressed using the Network Slice Service data model, defined in [I-D.ietf-teas-ietf-network-slice-nbi-yang];

* the VN type 2 data models, defined in [RFC9731], are intended to be used when there is a need to extend, with more control over network resources allocation by the customer, the connectivity service intent expressed using the VN type 1 data models, defined in [RFC9731].

Appendix D of [I-D.ietf-teas-ietf-network-slice-nbi-yang] provides guidance to decide when to use the Network Slice Service data model, defined in [I-D.ietf-teas-ietf-network-slice-nbi-yang], or the VN type 1 data models, defined in [RFC9731], to express the connectivity intent.

2.3. Relationship with Service Attachment Point (SAP) Topology

[RFC9408] introduces a YANG data model that represents an abstract view of the provider network topology. This model includes a list of Service Attachment Points (SAPs), where customer services can be connected. The SAP topology is made visible to customers by the provider before configuring network slice services. In contrast, the customer intent topology described in this document captures a customer’s intentions, while the provider acts as the recipient of these intents. As a result, these two models serve distinct purposes.

In certain scenarios, customers can leverage the SAP topology to construct customer intent topologies to aid in the realization of their intended network configurations. For instance, within a node of a customer intent topology, the Link Termination Point (LTP) identifiers may explicitly reference their supporting Termination Points (TPs), which correspond to the SAPs exposed in the provider’s SAP model. However, the specifics of this mechanism fall beyond the scope of this document.

2.4. Data Model Relationship

The data model presented in this document builds upon the generic network topology model defined in [RFC8345]. Other data models, including OTN Slicing (as defined in [I-D.ietf-ccamp-yang-otn-slicing]), can leverage this extended model.

The relationship of the related data models is illustrated in Figure 1. Within this diagram, the box outlined with dotted lines specifically represents the data model defined in this document.

```
        +----------+                 +----------+
        | Network  |                 | Network  |
        | Slice    |                 | Topology +
        | NBI YANG +------+          | Model    |
        | Model    |      |          | RFC 8345 |
        +----+-----+      |          +-----+----+
             |            |                |
             |augments    |augments        |augments
             |            |                |
        +----^-----+      |          ......^.....
        | OTN      |      +----------< Network  :
        | Slicing  | augments        : Slice    :
        | Model    >-----------------: Topology :
        |          |                 : Model    :
        +----------+                 :..........:
```

Figure 1: Model Relationship

3. Model Applicability

Network slicing can be achieved through various technologies. The data model defined in this document serves as a means for configuring resource reservation-based network slices. In this approach, resources for network slices are reserved and represented using a customer intent topology. This topology can then be mapped to a network resource partition (NRP) and realized based on the scenarios outlined in [RFC9543].

Network slices can be abstracted in various ways, depending on the specific requirements of the network slice customer. For instance, a customer might request a network slice with direct connectivity between pairs of Service Demarcation Points (SDPs). Within this network slice, each connectivity construct could be further supported by an end-to-end tunnel that follows a specific path defined in a customer intent topology, which the customer provides. The resources associated with each link are immediately commissioned during the network slice configuration process.

Alternatively, a customer can request resources to be reserved for potential network slices through a customer intent topology. These reserved resources are not immediately commissioned at the time of the request. Instead, they serve as a pool of allocated resources that the customer can utilize to build network slices in the future. By adopting this approach, customers gain the flexibility to share resources across multiple endpoints and activate them on demand.

In the example shown in Figure 2, two topology intents named as Network Slice Blue and Network Slice Red, are created by separate customers and delivered to the network slice service provider. The provider maps the two intents to corresponding network resource partitions (NRPs) internally. In realizing the network resource partitions, node virtualization is used to separate and allocate resources in physical devices. Two virtual routers VR1 and VR2 are created over physical router R1, and two virtual routers VR3 and VR4 are created over physical router R2, respectively. Each of the virtual routers,as a partition of the physical router, takes a portion of the resources such as ports and memory in the physical router.
Depending on the requirements and the implementations, they may share certain resources such as processors, ASICs, and switch fabric.

A network slice customer has the capability to configure customer intent topologies without needing any prior knowledge of the provider's network or resource availability. However, this approach could potentially create challenges for the provider in understanding and realizing the intended topology.

Alternatively, the provider can choose to describe the available resources and capabilities in the form of an abstract topology, which is then exposed to the customer before network slice requests. By doing so, the provider empowers the customer to build their customized intent topologies based on this pre-exposed information. This approach streamlines the process, minimizing unnecessary negotiations between the customer and the provider. The process and the data models for the provider to expose abstract topologies are outside the scope of this document.

The provider communicates the operational state of the customer intent topology, reflecting the allocated resources that result from negotiations between the customer and the provider. Subsequently, customers can process the requested customer intent topology and seamlessly integrate it into their own network topology. Importantly, this relationship between the customer and provider can be recursive. For instance, a customer who requests network slices can also serve as a provider, offering network slice services to its own customers further up the hierarchy.

As an example, Appendix B. shows the JSON encoded data instances of the customer topology intent for Network Slice Blue.

```
       Customer Topology (Merged)          Customer Topology (Merged)
       Network Slice Blue                  Network Slice Red
                            +---+         +---+      +---+
                       -----|R3 |---   ---|R1 |------|R2 |
                      /     +---+         +---+      +---+
      +---+      +---+        ^             ^          ^  \     +---+
   ---|R1 |------|R2 |        |             |          |   -----|R4 |---
      +---+      +---+        |             |          |        +---+
        ^          ^          v             v          v          ^
        |          |        +---+         +---+      +---+        |
        |          |   -----|VR5|---   ---|VR2|------|VR4|        |
        v          v  /     +---+         +---+      +---+        v
      +---+      +---+                                    \     +---+
   ---|VR1|------|VR3|                                     -----|VR6|---
      +---+      +---+                                          +---+
       Customer Topology (Intended)        Customer Topology (Intended)
       Network Slice Blue                  Network Slice Red

                                 Customers
   ---------------------------------------------------------------------
                                 Provider

        Customized Topology (Network Resouce Partition)
        Provider Network with Virtual Devices

        Network Slice Blue: VR1, VR3, VR5         +---+
                                        ----------|VR5|------
                                       /          +---+
                    +---+         +---+
              ------|VR1|---------|VR3|
                    +---+         +---+
              ------|VR2|---------|VR4|
                    +---+         +---+
                                       \          +---+
                                        ----------|VR6|------
        Network Slice Red: VR2, VR4, VR6          +---+

                                 Virtual Devices
   ---------------------------------------------------------------------
                                 Physical Devices

        Native Topology
        Provider Network with Physical Devices
                                                  +---+
                                        ----------|R3 |------
                                       /          +---+
                    +---+         +---+
              ======|R1 |=========|R2 |
```

```
                    +---+         +---+
                                       \          +---+
                                        ----------|R4 |------
                                                  +---+
```

Figure 2: Network Slicing Topologies for Virtualization

4. YANG Model Overview

The YANG data model in this draft consists of two modules for flexible use and augmentation: - The first YANG module defines a customer intent topology, with SLO and SLE associated with the topological constructs. - The second YANG module extends the YANG model defined in [I-D.ietf-teas-ietf-network-slice-nbi-yang] by adding underlay paths to the connectivity constructs.

Within the YANG model, the following constructs and attributes are defined: - Network Topology: This represents a set of shared and reserved resources, organized as a virtual topology connecting all endpoints. Customers can utilize this network topology to define detailed connectivity paths traversing the topology. Additionally, it enables resource sharing between different endpoints.

* Service-Level Objectives (SLOs): These objectives are associated with various objects within the topology, including nodes, links, and termination points. SLOs provide guidelines for achieving specific performance or quality targets.

5. Model Tree Structure

5.1. Network Slice Topology Model Tree Structure

```
module: ietf-ns-topo

  augment /nw:networks/nw:network/nw:network-types:
    +--rw network-slice!
  augment /nw:networks/nw:network:
    +--rw (slo-sle-policy)?
       +--:(standard)
       |  +--rw slo-sle-template?         slice-template-ref
       +--:(custom)
          +--rw service-slo-sle-policy
             +--rw description?   string
             +--rw slo-policy
             |  +--rw metric-bound* [metric-type]
             |  |  +--rw metric-type          identityref
             |  |  +--rw metric-unit          string
             |  |  +--rw value-description?   string
```

```
                |  |  +--rw percentile-value?    percentile
                |  |  +--rw bound?               uint64
                |  +--rw availability?   identityref
                |  +--rw mtu?            uint32
                +--rw sle-policy
                   +--rw security*              identityref
                   +--rw isolation*             identityref
                   +--rw max-occupancy-level?   uint8
                   +--rw path-constraints
                      +--rw service-functions
                      +--rw diversity
                         +--rw diversity-type?
                                 te-types:te-path-disjointness
     augment /nw:networks/nw:network/nw:node:
       +--rw (slo-sle-policy)?
          +--:(standard)
          |  +--rw slo-sle-template?         slice-template-ref
          +--:(custom)
             +--rw service-slo-sle-policy
                +--rw description?   string
                +--rw slo-policy
                |  +--rw metric-bound* [metric-type]
                |  |  +--rw metric-type          identityref
                |  |  +--rw metric-unit          string
                |  |  +--rw value-description?   string
                |  |  +--rw percentile-value?    percentile
                |  |  +--rw bound?               uint64
                |  +--rw availability?   identityref
                |  +--rw mtu?            uint32
                +--rw sle-policy
                   +--rw security*              identityref
                   +--rw isolation*             identityref
                   +--rw max-occupancy-level?   uint8
                   +--rw path-constraints
                      +--rw service-functions
                      +--rw diversity
                         +--rw diversity-type?
                                 te-types:te-path-disjointness
     augment /nw:networks/nw:network/nw:node/nt:termination-point:
       +--rw (slo-sle-policy)?
          +--:(standard)
          |  +--rw slo-sle-template?         slice-template-ref
          +--:(custom)
             +--rw service-slo-sle-policy
                +--rw description?   string
                +--rw slo-policy
                |  +--rw metric-bound* [metric-type]
                |  |  +--rw metric-type          identityref
```

```
                |  |  +--rw metric-unit          string
                |  |  +--rw value-description?   string
                |  |  +--rw percentile-value?    percentile
                |  |  +--rw bound?               uint64
                |  +--rw availability?   identityref
                |  +--rw mtu?            uint32
                +--rw sle-policy
                   +--rw security*              identityref
                   +--rw isolation*             identityref
                   +--rw max-occupancy-level?   uint8
                   +--rw path-constraints
                      +--rw service-functions
                      +--rw diversity
                         +--rw diversity-type?
                                 te-types:te-path-disjointness
     augment /nw:networks/nw:network/nt:link:
       +--rw (slo-sle-policy)?
          +--:(standard)
          |  +--rw slo-sle-template?         slice-template-ref
          +--:(custom)
             +--rw service-slo-sle-policy
                +--rw description?   string
                +--rw slo-policy
                |  +--rw metric-bound* [metric-type]
                |  |  +--rw metric-type          identityref
                |  |  +--rw metric-unit          string
                |  |  +--rw value-description?   string
                |  |  +--rw percentile-value?    percentile
                |  |  +--rw bound?               uint64
                |  +--rw availability?   identityref
                |  +--rw mtu?            uint32
                +--rw sle-policy
                   +--rw security*              identityref
                   +--rw isolation*             identityref
                   +--rw max-occupancy-level?   uint8
                   +--rw path-constraints
                      +--rw service-functions
                      +--rw diversity
                         +--rw diversity-type?
                                 te-types:te-path-disjointness
```

Figure 3: Tree diagram for network slice topology

5.2. Network Slice Underlay Path Model Tree Structure

```
module: ietf-ns-underlay-path

  augment /ietf-nss:network-slice-services/ietf-nss:slice-service
            /ietf-nss:connection-groups/ietf-nss:connection-group
            /ietf-nss:slo-sle-policy/ietf-nss:custom
            /ietf-nss:service-slo-sle-policy/ietf-nss:sle-policy
            /ietf-nss:path-constraints:
    +--rw underlay-path
       +--rw network-ref?    -> /nw:networks/network/network-id
       +--rw path-element* [index]
          +--rw index            uint32
          +--rw is-strict-hop?   boolean
          +--rw (type)?
             +--:(node-hop)
             |  +--rw node-id?   nw:node-id
             +--:(link-hop)
             |  +--rw link-id?   nt:link-id
             +--:(tp-hop)
                +--rw tp-id?     nt:tp-id
  augment /ietf-nss:network-slice-services/ietf-nss:slice-service
            /ietf-nss:connection-groups/ietf-nss:connection-group
            /ietf-nss:connectivity-construct/ietf-nss:slo-sle-policy
            /ietf-nss:custom/ietf-nss:service-slo-sle-policy
            /ietf-nss:sle-policy/ietf-nss:path-constraints:
    +--rw underlay-path
       +--rw network-ref?    -> /nw:networks/network/network-id
       +--rw path-element* [index]
          +--rw index            uint32
          +--rw is-strict-hop?   boolean
          +--rw (type)?
             +--:(node-hop)
             |  +--rw node-id?   nw:node-id
             +--:(link-hop)
             |  +--rw link-id?   nt:link-id
             +--:(tp-hop)
                +--rw tp-id?     nt:tp-id
  augment /ietf-nss:network-slice-services/ietf-nss:slice-service
            /ietf-nss:connection-groups/ietf-nss:connection-group
            /ietf-nss:connectivity-construct/ietf-nss:type
            /ietf-nss:a2a/ietf-nss:a2a-sdp/ietf-nss:slo-sle-policy
            /ietf-nss:custom/ietf-nss:service-slo-sle-policy
            /ietf-nss:sle-policy/ietf-nss:path-constraints:
    +--rw underlay-path
       +--rw network-ref?    -> /nw:networks/network/network-id
       +--rw path-element* [index]
          +--rw index            uint32
          +--rw is-strict-hop?   boolean
          +--rw (type)?
```

```
                +--:(node-hop)
                |  +--rw node-id?   nw:node-id
                +--:(link-hop)
                |  +--rw link-id?   nt:link-id
                +--:(tp-hop)
                   +--rw tp-id?     nt:tp-id
```

Figure 4: Tree diagram for underlay path

6. YANG Modules

6.1. YANG Module for Network Slice Topology

```
<CODE BEGINS> file "ietf-ns-topo@2025-07-03.yang"
   module ietf-ns-topo {
     yang-version 1.1;
     namespace
       "urn:ietf:params:xml:ns:yang:ietf-ns-topo";
     prefix "ns-topo";

     import ietf-network {
       prefix "nw";
       reference
        "RFC 8345: A YANG Data Model for Network Topologies";
     }
     import ietf-network-topology {
       prefix "nt";
       reference
        "RFC 8345: A YANG Data Model for Network Topologies";
     }

     import ietf-network-slice-service {
       prefix "ietf-nss";
       reference
         "draft-ietf-teas-ietf-network-slice-nbi-yang-25:
          A YANG Data Model for the RFC 9543 Network Slice Service";
     }

     organization
       "IETF TEAS Working Group";
     contact
       "WG Web: <http://tools.ietf.org/wg/teas/>
        WG List: <mailto:teas@ietf.org>

        Editor: Xufeng Liu
                <mailto:xufeng.liu.ietf@gmail.com>

        Editor: Italo Busi
```

```
                 <mailto:italo.busi@huawei.com>

         Editor: Aihua Guo
                 <mailto:aihuaguo.ietf@gmail.com>

         Editor: Sergio Belotti
                 <mailto:sergio.belotti@nokia.com>

         Editor: Luis M. Contreras
                 <mailto:luismiguel.contrerasmurillo@telefonica.com>";

       description
         "This module defines a base YANG data model for configuring
          customer intent topologies for RFC9543 network slices.

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

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

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

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


       revision 2025-07-03 {
         description "Initial revision";
         reference
           "RFC XXXX: IETF Network Slice Topology YANG Data Model";
       }

       /*
        * Augmented data nodes
        */
       /* network type augments */
       augment "/nw:networks/nw:network/nw:network-types" {
         description
           "Defines the Network Slice topology type.";
         container network-slice {
           presence "Indicates a Network Slice topology";
           description
```

```
             "Its presence identifies the Network Slice type.";
         }
       }

       /* network topology augments */
       augment "/nw:networks/nw:network" {
         when "./nw:network-types/ns-topo:network-slice" {
           description
             "Augmentation parameters apply only for networks
              of type Network Slice topology.";
         }
         description
           "SLO and SLE for topology.";

         uses ietf-nss:service-slo-sle-policy;
       }

       /* network node augments */
       augment "/nw:networks/nw:network/nw:node" {
         when "../nw:network-types/ns-topo:network-slice" {
           description
             "Augmentation parameters apply only for networks
              of type Network Slice topology.";
         }
         description
           "SLO and SLE for nodes.";

         uses ietf-nss:service-slo-sle-policy;
       }

       /* network node’s termination point augments */
       augment "/nw:networks/nw:network/nw:node" +
               "/nt:termination-point" {
         when "../../nw:network-types/ns-topo:network-slice" {
           description
             "Augmentation parameters apply only for networks
              of type Network Slice topology.";
         }
         description
           "SLO and SLE for termination points.";

       uses ietf-nss:service-slo-sle-policy;
       }

       /* network link augments */
       augment "/nw:networks/nw:network/nt:link" {
         when "../nw:network-types/ns-topo:network-slice" {
           description
```

```
              "Augmentation parameters apply only for networks
               of type Network Slice topology.";
          }
          description
            "SLO and SLE for links.";

          uses ietf-nss:service-slo-sle-policy;
        }
      }
   <CODE ENDS>
```

Figure 5: YANG model for network slice topology

6.2. YANG Module for Network Slice Underlay Path

```
   <CODE BEGINS> file "ietf-ns-underlay-path@2025-07-03.yang"
      module ietf-ns-underlay-path {
        yang-version 1.1;
        namespace
          "urn:ietf:params:xml:ns:yang:ietf-ns-underlay-path";
        prefix "ns-path";

        import ietf-network {
          prefix "nw";
          reference
           "RFC 8345: A YANG Data Model for Network Topologies";
        }
        import ietf-network-topology {
          prefix "nt";
          reference
           "RFC 8345: A YANG Data Model for Network Topologies";
        }

        import ietf-network-slice-service {
          prefix "ietf-nss";
          reference
            "draft-ietf-teas-ietf-network-slice-nbi-yang-25:
             A YANG Data Model for the RFC 9543 Network Slice Service";
        }

        organization
          "IETF TEAS Working Group";
        contact
          "WG Web: <http://tools.ietf.org/wg/teas/>
           WG List: <mailto:teas@ietf.org>

           Editor: Xufeng Liu
                   <mailto:xufeng.liu.ietf@gmail.com>
```

```
           Editor: Italo Busi
                   <mailto:italo.busi@huawei.com>

           Editor: Aihua Guo
                   <mailto:aihuaguo.ietf@gmail.com>

           Editor: Sergio Belotti
                   <mailto:sergio.belotti@nokia.com>

           Editor: Luis M. Contreras
                   <mailto:luismiguel.contrerasmurillo@telefonica.com>";

        description
          "This module defines a base YANG data model for configuring
           the underlay path of connectivity intent over a customer
           intent topology for RFC9543 network slices.

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

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

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

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


        revision 2025-07-03 {
          description "Initial revision";
          reference
            "RFC XXXX: IETF Network Slice Topology YANG Data Model";
        }

        /*
         * Groupings
         */
        grouping underlay-path {
          description
            "Underlay explicit path within a customer intent
             topology.";
```

```
          container underlay-path {
            description
              "Defines an underlay explicit path within specific
               customer intent topology.";

            uses nw:network-ref;

            list path-element {
             key "index";
             description
               "List of path elements.";
             leaf index {
               type uint32;
               description
                 "Index of the hop within the underlay path.";
             }
             leaf is-strict-hop {
               type boolean;
               description
                 "Indicate whether the hop is strict or loose";
             }
             choice type {
               description
                 "Type of the hop.";
               case node-hop {
                 leaf node-id {
                   type nw:node-id;
                   description
                     "Node identifier.";
                 }
               }
               case link-hop {
                 leaf link-id {
                   type nt:link-id;
                   description
                     "Link identifier.";
                 }
               }
               case tp-hop {
                 leaf tp-id {
                   type nt:tp-id;
                   description
                     "Termination Point (TP) identifier.";
                 }
               }
             }
            }
          }
```

```
        }

        /*
         * Augmented data nodes
         */
        augment "/ietf-nss:network-slice-services" +
                "/ietf-nss:slice-service" +
                "/ietf-nss:connection-groups" +
                "/ietf-nss:connection-group" +
                "/ietf-nss:slo-sle-policy" +
                "/ietf-nss:custom" +
                "/ietf-nss:service-slo-sle-policy" +
                "/ietf-nss:sle-policy" +
                "/ietf-nss:path-constraints" {
          description
            "Underlay path for connection group.";

          uses underlay-path;
        }

        augment "/ietf-nss:network-slice-services" +
                "/ietf-nss:slice-service" +
                "/ietf-nss:connection-groups" +
                "/ietf-nss:connection-group" +
                "/ietf-nss:connectivity-construct" +
                "/ietf-nss:slo-sle-policy" +
                "/ietf-nss:custom" +
                "/ietf-nss:service-slo-sle-policy" +
                "/ietf-nss:sle-policy" +
                "/ietf-nss:path-constraints" {
          description
            "Underlay path for connectivity construct.";

          uses underlay-path;
        }

        augment "/ietf-nss:network-slice-services" +
                "/ietf-nss:slice-service" +
                "/ietf-nss:connection-groups" +
                "/ietf-nss:connection-group" +
                "/ietf-nss:connectivity-construct" +
                "/ietf-nss:type" +
                "/ietf-nss:a2a" +
                "/ietf-nss:a2a-sdp" +
                "/ietf-nss:slo-sle-policy" +
                "/ietf-nss:custom" +
                "/ietf-nss:service-slo-sle-policy" +
                "/ietf-nss:sle-policy" +
```

```
              "/ietf-nss:path-constraints" {
        description
          "Underlay path for a2a connectivity constructs.";

        uses underlay-path;
      }
    }
  <CODE ENDS>
```

Figure 6: YANG model for underlay path

7. Manageability Considerations

To ensure the security and controllability of physical resource isolation, slice-based independent operation and management are required to achieve management isolation. Each network slice typically requires dedicated accounts, permissions, and resources for independent access and O&M. This mechanism is to guarantee the information isolation among slice tenants and to avoid resource conflicts. The access to slice management functions will only be permitted after successful security checks.

8. Security Considerations

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

The 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. Considerations in Section 8 of [RFC8795] are also applicable to their subtrees in the module defined in this document.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. Considerations in Section 8 of [RFC8795] are also applicable to their subtrees in the module defined in this document.

9. IANA Considerations

It is proposed to IANA to assign new URIs from the "IETF XML Registry" [RFC3688] as follows:

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

URI: urn:ietf:params:xml:ns:yang:ietf-ns-underlay-path
Registrant Contact: The IESG
XML: N/A; the requested URI is an XML namespace.

This document registers two YANG modules in the YANG Module Names registry [RFC6020].

name: ietf-ns-topo
namespace: urn:ietf:params:xml:ns:yang:ietf-ns-topo
prefix: ns-topo
reference: RFC XXXX

name: ietf-ns-underlay-path
namespace: urn:ietf:params:xml:ns:yang:ietf-ns-underlay-path
prefix: ns-path
reference: RFC XXXX

10. Acknowledgments

The authors would like to thank Danielle Ceccarelli, Bo Wu, Mohamed Boucadair, Vishnu Beeram, Dhruv Dhody, Oscar G. De Dios, Adrian Farrel, and many others for their valuable comments and suggestions.

11. References

11.1. Normative References

[I-D.ietf-ccamp-otn-topo-yang]
Zheng, H., Busi, I., Liu, X., Belotti, S., and O. G. de Dios, "A YANG Data Model for Optical Transport Network Topology", Work in Progress, Internet-Draft, draft-ietf-ccamp-otn-topo-yang-20, 7 November 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-ccamp-otn-topo-yang-20>.

[I-D.ietf-ccamp-yang-otn-slicing]
Guo, A., Contreras, L. M., Belotti, S., Rokui, R., Xu, Y., Zhao, Y., and X. Liu, "Framework and Data Model for OTN Network Slicing", Work in Progress, Internet-Draft, draft-ietf-ccamp-yang-otn-slicing-10, 5 January 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-ccamp-yang-otn-slicing-10>.

[I-D.ietf-teas-ietf-network-slice-nbi-yang]
Wu, B., Dhody, D., Rokui, R., Saad, T., and J. Mullooly, "A YANG Data Model for the RFC 9543 Network Slice Service", Work in Progress, Internet-Draft, draft-ietf-teas-ietf-network-slice-nbi-yang-25, 9 May 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-teas-ietf-network-slice-nbi-yang-25>.

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

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[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>.

[RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <https://www.rfc-editor.org/info/rfc7951>.

[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>.

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

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

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

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

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

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

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

[RFC9731] Lee, Y., Ed., Dhody, D., Ed., Ceccarelli, D., Bryskin, I., and B. Yoon, "A YANG Data Model for Virtual Network (VN) Operations", RFC 9731, DOI 10.17487/RFC9731, March 2025, <https://www.rfc-editor.org/info/rfc9731>.

11.2. Informative References

[I-D.ietf-teas-ns-controller-models]
Contreras, L. M., Rokui, R., Tantsura, J., Wu, B., and X. Liu, "IETF Network Slice Controller and its Associated Data Models", Work in Progress, Internet-Draft, draft-ietf-teas-ns-controller-models-06, 20 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-teas-ns-controller-models-06>.

[RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC 8453, DOI 10.17487/RFC8453, August 2018, <https://www.rfc-editor.org/info/rfc8453>.

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

Appendix A. Relationship with ACTN Virtual Network (VN)

[RFC8453] and [RFC9731] introduce the concept of a Virtual Network (VN), which can be presented to customers. These VNs are constructed from abstractions of the underlying networks, specifically those that are traffic-engineering (TE) capable. While VNs share similarities with RFC 9543 network slicing, they operate under the assumption of TE-capable networks.

Two distinct types of VNs are defined:

* Type 1 VN: Modeled as a single abstract node with edge-to-edge connectivity between customer endpoints.

* Type 2 VN: Modeled as a single abstract node with an underlay topology, allowing configuration of intended underlay paths for connections within the single abstract node.

The topologies for VNs, including both the single-node abstract topology and the underlay topology, can either be mutually agreed upon between the Customer Network Controller (CNC) and the Multi-Domain Service Coordinator (MDSC) prior to VN creation, or they can be created as part of VN instantiation by the customer.

In the context of network slicing, [RFC9543] defines a network slice service as a collection of connectivity constructs between pairs of Service Demarcation Points (SDPs). This concept closely resembles the Type 1 VN, which is implemented as a single abstract node.

[I-D.ietf-teas-ietf-network-slice-nbi-yang] further elaborates on network slices by incorporating references to a customer intent topology based on [RFC8345]. This approach aligns with the ACTN Type 2 VN, although without specifying the explicit use of such a topology.

Consequently, the data model defined in this document serves as a complementary option to the data model outlined in [I-D.ietf-teas-ietf-network-slice-nbi-yang]. It empowers customers to define a customized intent topology specifically tailored for their network slices.

Reusing the Type 2 VN for defining customer intent topologies alongside the RFC9543 network slice service model would result in duplicated information for connectivity intents (SDPs and connectivity-constructs vs. LTPs and connectivity matrices), and additionally, would bind the network slice solution to TE technologies (as discussed in Appendix D of [I-D.ietf-teas-ietf-network-slice-nbi-yang] for VN Type 1).

Appendix B. Data Tree for the Example in Section 3

B.1. Native Topology

This section contains an example of an instance data tree in the JSON encoding [RFC7951]. The example instantiates "ietf-network" for the topology of Network Slice Blue depicted in Figure 2.

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

```
{
  "ietf-network:networks": {
    "network": [
      {
        "network-id": "example-customized-blue-topology",
        "network-types": {
          "ietf-ns-topo:network-slice": {
```

```
            }
          },
          "node": [
            {
              "node-id": "VR1",
              "ietf-ns-topo:service-slo-sle-policy": {
                "sle-policy": {
                  "isolation": [
                    {
                      "ietf-network-slice-service:service-traffic-iso\
   lation"
                    }
                  ]
                }
              },
              "ietf-network-topology:termination-point": [
                {
                  "tp-id": "1-0-1"
                },
                {
                  "tp-id": "1-3-1"
                }
              ]
            },
            {
              "node-id": "VR3",
              "ietf-ns-topo:service-slo-sle-policy": {
                "sle-policy": {
                  "isolation": [
                    {
                      "ietf-network-slice-service:service-traffic-iso\
   lation"
                    }
                  ]
                }
              },
              "ietf-network-topology:termination-point": [
                {
                  "tp-id": "3-1-1"
                },
                {
                  "tp-id": "3-5-1"
                }
              ]
            },
            {
              "node-id": "VR5",
              "ietf-ns-topo:service-slo-sle-policy": {
```

```
                 "sle-policy": {
                   "isolation": [
                     {
                       "ietf-network-slice-service:service-traffic-iso\
   lation"
                     }
                   ]
                 }
               },
               "ietf-network-topology:termination-point": [
                 {
                   "tp-id": "5-3-1"
                 },
                 {
                   "tp-id": "5-0-1"
                 }
               ]
             }
           ],
           "ietf-network-topology:link": [
             {
               "link-id": "VR1,1-0-1,,",
               "source": {
                 "source-node": "VR1",
                 "source-tp": "1-0-1"
               },
               "ietf-ns-topo:service-slo-sle-policy": {
                 "slo-policy": {
                   "metric-bounds": {
                     "metric-bound": [
                       {
                         "metric-type": "ietf-network-slice-service:se\
   rvice-slo-two-way-delay",
                         "metric-unit": "ms",
                         "bound": 60
                       }
                     ]
                   }
                 },
                 "sle-policy": {
                   "isolation": [
                     {
                       "ietf-network-slice-service:service-traffic-iso\
   lation"
                     }
                   ]
                 }
               }
```

```
             },
             {
               "link-id": ",,VR1,1-0-1",
               "destination": {
                 "dest-node": "VR1",
                 "dest-tp": "1-0-1"
               },
               "ietf-ns-topo:service-slo-sle-policy": {
                 "slo-policy": {
                   "metric-bounds": {
                     "metric-bound": [
                       {
                         "metric-type": "ietf-network-slice-service:se\
   rvice-slo-two-way-delay",
                         "metric-unit": "ms",
                         "bound": 30
                       }
                     ]
                   }
                 },
                 "sle-policy": {
                   "isolation": [
                     {
                       "ietf-network-slice-service:service-traffic-iso\
   lation"
                     }
                   ]
                 }
               }
             },
             {
               "link-id": "VR1,1-3-1,VR3,3-1-1",
               "source": {
                 "source-node": "VR1",
                 "source-tp": "1-3-1"
               },
               "destination": {
                 "dest-node": "VR3",
                 "dest-tp": "3-1-1"
               },
               "ietf-ns-topo:service-slo-sle-policy": {
                 "slo-policy": {
                   "metric-bounds": {
                     "metric-bound": [
                       {
                         "metric-type": "ietf-network-slice-service:se\
   rvice-slo-two-way-delay",
                         "metric-unit": "ms",
```

```
                        "bound": 30
                      }
                    ]
                  }
                },
                "sle-policy": {
                  "isolation": [
                    {
                      "ietf-network-slice-service:service-traffic-iso\
   lation"
                    }
                  ]
                }
              }
            },
            {
              "link-id": "VR3,3-1-1,VR1,1-3-1",
              "source": {
                "source-node": "VR3",
                "source-tp": "3-1-1"
              },
              "destination": {
                "dest-node": "R1",
                "dest-tp": "1-3-1"
              },
              "ietf-ns-topo:service-slo-sle-policy": {
                "slo-policy": {
                  "metric-bounds": {
                    "metric-bound": [
                      {
                        "metric-type": "ietf-network-slice-service:se\
   rvice-slo-two-way-delay",
                        "metric-unit": "ms",
                        "bound": 30
                      }
                    ]
                  }
                },
                "sle-policy": {
                  "isolation": [
                    {
                      "ietf-network-slice-service:service-traffic-iso\
   lation"
                    }
                  ]
                }
              }
            },
```

```
             {
               "link-id": "VR3,3-5-1,VR5,5-3-1",
               "source": {
                 "source-node": "VR3",
                 "source-tp": "3-5-1"
               },
               "destination": {
                 "dest-node": "VR5",
                 "dest-tp": "5-3-1"
               },
               "ietf-ns-topo:service-slo-sle-policy": {
                 "slo-policy": {
                   "metric-bounds": {
                     "metric-bound": [
                       {
                         "metric-type": "ietf-network-slice-service:se\
   rvice-slo-two-way-delay",
                         "metric-unit": "ms",
                         "bound": 35
                       }
                     ]
                   }
                 },
                 "sle-policy": {
                   "isolation": [
                     {
                       "ietf-network-slice-service:service-traffic-iso\
   lation"
                     }
                   ]
                 }
               }
             },
             {
               "link-id": "VR5,5-3-1,VR3,3-5-1",
               "source": {
                 "source-node": "VR5",
                 "source-tp": "5-3-1"
               },
               "destination": {
                 "dest-node": "VR3",
                 "dest-tp": "3-5-1"
               },
               "ietf-ns-topo:service-slo-sle-policy": {
                 "slo-policy": {
                   "metric-bounds": {
                     "metric-bound": [
                       {
```

```
                         "metric-type": "ietf-network-slice-service:se\
   rvice-slo-two-way-delay",
                         "metric-unit": "ms",
                         "bound": 35
                       }
                     ]
                   }
                 },
                 "sle-policy": {
                   "isolation": [
                     {
                       "ietf-network-slice-service:service-traffic-iso\
   lation"
                     }
                   ]
                 }
               }
             },
             {
               "link-id": "VR5,5-0-1,,",
               "source": {
                 "source-node": "VR5",
                 "source-tp": "5-0-1"
               },
               "ietf-ns-topo:service-slo-sle-policy": {
                 "slo-policy": {
                   "metric-bounds": {
                     "metric-bound": [
                       {
                         "metric-type": "ietf-network-slice-service:se\
   rvice-slo-two-way-delay",
                         "metric-unit": "ms",
                         "bound": 25
                       }
                     ]
                   }
                 },
                 "sle-policy": {
                   "isolation": [
                     {
                       "ietf-network-slice-service:service-traffic-iso\
   lation"
                     }
                   ]
                 }
         }
             },
             {
```

```
               "link-id": ",,VR5,5-0-1",
               "destination": {
                 "dest-node": "VR5",
                 "dest-tp": "5-0-1"
               },
               "ietf-ns-topo:service-slo-sle-policy": {
                 "slo-policy": {
                   "metric-bounds": {
                     "metric-bound": [
                       {
                         "metric-type": "ietf-network-slice-service:se\
   rvice-slo-two-way-delay",
                         "metric-unit": "ms",
                         "bound": 25
                       }
                     ]
                   }
                 },
                 "sle-policy": {
                   "isolation": [
                     {
                       "ietf-network-slice-service:service-traffic-iso\
   lation"
                     }
                   ]
                 }
               }
             }
           ],
           "ietf-ns-topo:service-slo-sle-policy": {
             "sle-policy": {
               "isolation": [
                 {
                   "ietf-network-slice-service:service-traffic-isolati\
   on"
                 }
               ]
             }
           }
         }
       ]
     }
   }
```


Contributors

Reza Rokui
Ciena

Email: rrokui@ciena.com

Jeff Tantsura
Microsoft
Email: jefftant.ietf@gmail.com

Igor Bryskin
Individual
Email: i_bryskin@yahoo.com

Qin Wu
Huawei
Email: bill.wu@huawei.com

Authors’ Addresses

Xufeng Liu
Alef Edge
Email: xufeng.liu.ietf@gmail.com

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

Sergio Belotti
Nokia
Email: Sergio.belotti@nokia.com

Aihua Guo
Futurewei
Email: aihuaguo.ietf@gmail.com

Italo Busi
Huawei
Email: italo.busi@huawei.com