

Workgroup: CCAMP Working Group  
Internet-Draft:  
draft-zheng-ccamp-yang-otn-slicing-03  
Published: 22 October 2021

Intended Status: Standards Track

Expires: 25 April 2022

Authors: H. Zheng I. Busi  
Huawei Technologies Huawei Technologies  
A. Guo L.M. Contreras  
Futurewei Technologies Telefonica  
O.G.d. Dios V. Lopez S. Belotti D. Beller  
Telefonica Nokia Nokia Nokia  
R. Rokui Y. Xu Y. Zhao X. Liu  
Nokia CAICT China Mobile Volta Networks  
**Framework and Data Model for OTN Network Slicing**

## Abstract

The requirement of slicing network resources with desired quality of service is emerging at every network technology, including the Optical Transport Networks (OTN). As a part of the transport network, OTN can provide hard pipes with guaranteed data isolation and deterministic low latency, which are highly demanded in the Service Level Agreement (SLA).

This document describes a framework for OTN network slicing and a YANG data model augmentation of the OTN topology model. Additional YANG data model augmentations will be defined in a future version of this draft.

## 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 25 April 2022.

## Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

- [1. Introduction](#)
  - [1.1. Definition of OTN Slice](#)
- [2. Use Cases for OTN Network Slicing](#)
  - [2.1. Leased Line Services with OTN](#)
  - [2.2. Co-construction and Sharing](#)
  - [2.3. Wholesale of optical resources](#)
  - [2.4. Vertical dedicated network with OTN](#)
  - [2.5. End-to-end network slicing](#)
- [3. Framework for OTN slicing](#)
- [4. YANG Data Model for OTN Slicing Configuration](#)
  - [4.1. OTN Slicing YANG Model for MPI](#)
    - [4.1.1. MPI YANG Model Overview](#)
    - [4.1.2. MPI YANG Model Tree](#)
    - [4.1.3. MPI YANG Code](#)
  - [4.2. OTN Slicing YANG Model for OTN-SC NBI](#)
    - [4.2.1. NBI YANG Model Overview](#)
    - [4.2.2. NBI YANG Model Tree for Transport Network Slice](#)
    - [4.2.3. NBI YANG Code for Transport Network Slice](#)
    - [4.2.4. NBI YANG Model Tree for OTN slice](#)
    - [4.2.5. NBI YANG Code for OTN Slice](#)
- [5. Manageability Considerations](#)
- [6. Security Considerations](#)
- [7. IANA Considerations](#)
- [8. References](#)
  - [8.1. Normative References](#)
  - [8.2. Informative References](#)
- [Acknowledgments](#)
- [Contributors' Addresses](#)
- [Authors' Addresses](#)

## 1. Introduction

The requirement of slicing network resources with desired quality of service is emerging at every network technology, including the Optical Transport Networks (OTN). As a part of the transport network, OTN can provide hard pipes with guaranteed data isolation and deterministic low latency, which are highly demanded in the Service Level Agreement (SLA). This document describes a framework for OTN network slicing and a YANG data model augmentation of the OTN topology model. Additional YANG data model augmentations will be defined in a future version of this draft.

### 1.1. Definition of OTN Slice

An OTN slice is an OTN virtual network topology connecting a number of OTN endpoints using a set of shared or dedicated OTN network resources to satisfy specific service level objectives (SLOs).

An OTN slice is a technology-specific realization of an IETF network slice [[I-D.ietf-teas-ietf-network-slices](#)] in the OTN domain, with the capability of configuring slice resources in the term of OTN technologies. Therefore, all the terms and definitions concerning network slicing as defined in [[I-D.ietf-teas-ietf-network-slices](#)] apply to OTN slicing.

An OTN slice can span multiple OTN administrative domains, encompassing access links, intra-domain paths, and inter-domain links. An OTN slice may include multiple endpoints, each associated with a set of physical or logical resources, e.g. optical port or time slots, at the termination point (TP) of an access link or inter-domain link at an OTN provider edge (PE) equipment.

An end-to-end OTN slice may be composed of multiple OTN segment slices in a hierarchical or sequential (or stitched) combination.

[Figure 1](#) illustrates the scope of OTN slices in multi-domain environment.

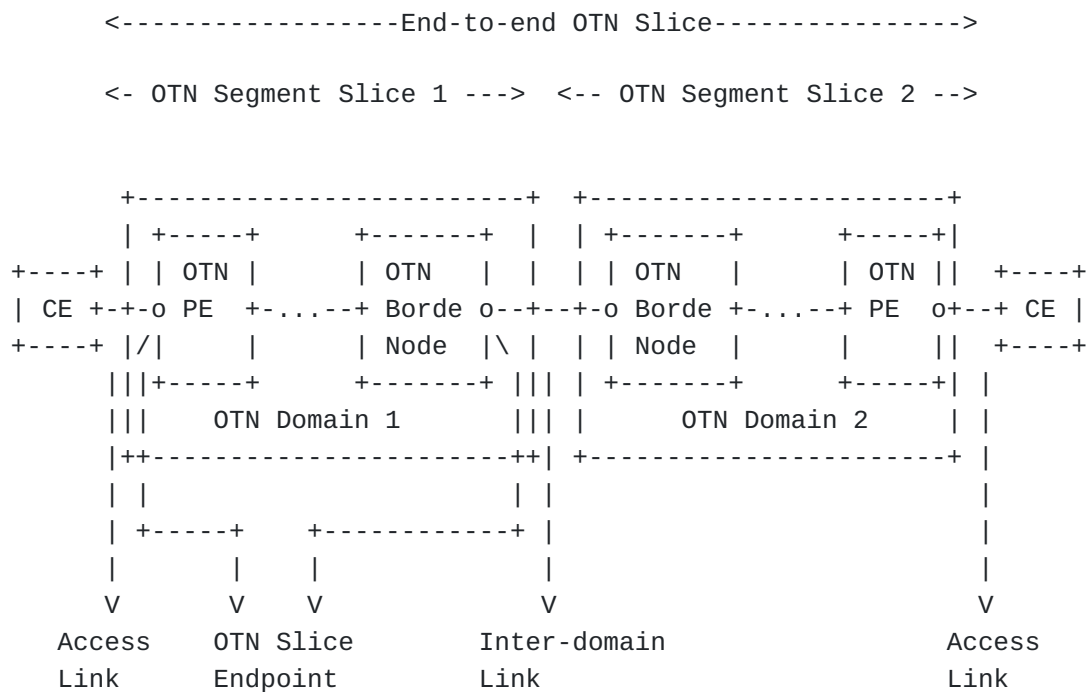


Figure 1: OTN Slice

OTN slices may be pre-configured by the management plane and presented to the customer via the northbound interface (NBI), or be dynamically provisioned by a higher layer slice controller, e.g. an IETF network slice controller (IETF NSC) through the NBI. The OTN slice is provided by a service provider to a customer to be used as though it was part of the customer's own networks.

## 2. Use Cases for OTN Network Slicing

### 2.1. Leased Line Services with OTN

For end business customers (like OTT or enterprises), leased lines have the advantage of providing high-speed connections with low costs. On the other hand, the traffic control of leased lines is very challenging due to rapid changes in service demands. Carriers are recommended to provide network-level slicing capabilities to meet this demand. Based on such capabilities, private network users have full control over the sliced resources which have been allocated to them and which could be used to support their leased lines, when needed. Users may formulate policies based on the demand for services and time to schedule the resources from the entire network's perspective flexibly. For example, the bandwidth between any two points may be established or released based on the time or monitored traffic characteristics. The routing and bandwidth may be adjusted at a specific time interval to maximize network resource utilization efficiency.

## **2.2. Co-construction and Sharing**

Co-construction and sharing of a network are becoming a popular means among service providers to reduce networking building CAPEX. For Co- construction and sharing case, there are typically multiple co- founders for the same network. For example, one founder may provide optical fibres and another founder may provide OTN equipment, while each occupies a certain percentage of the usage rights of the network resources. In this scenario, the network O&M is performed by a certain founder in each region, where the same founder usually deploys an independent management and control system. The other founders of the network use each other's management and control system to provision services remotely. In this scenario, different founders' network resources need to be automatically (associated) divided, isolated, and visualized. All founders may share or have independent O&M capabilities, and should be able to perform service- level provisioning in their respective slices.

## **2.3. Wholesale of optical resources**

In the optical resource wholesale market, smaller, local carriers and wireless carriers may rent resources from larger carriers, or infrastructure carriers instead of building their networks. Likewise, international carriers may rent resources from respective local carriers and local carriers may lease their owned networks to each other to achieve better network utilization efficiency. From the perspective of a resource provider, it is crucial that a network slice is timely configured to meet traffic matrix requirements requested by its tenants. The support for multi-tenancy within the resource provider's network demands that the network slices are qualitatively isolated from each other to meet the requirements for transparency, non-interference, and security. Typically, a resource purchaser expects to use the leased network resources flexibly, just like they are self-constructed. Therefore, the purchaser is not only provided with a network slice, but also the full set of functionalities for operating and maintaining the network slice. The purchaser also expects to, flexibly and independently, schedule and maintain physical resources to support their own end-to-end automation using both leased and self-constructed network resources.

## **2.4. Vertical dedicated network with OTN**

Vertical industry slicing is an emerging category of network slicing due to the high demand for private high-speed network interconnects for industrial applications. In this scenario, the biggest challenge is to implement differentiated optical network slices based on the requirements from different industries. For example, in the financial industry, to support high-frequency transactions, the

slice must ensure to provide the minimum latency along with the mechanism for latency management. For the healthcare industry, online diagnosis network and software capabilities to ensure the delivery of HD video without frame loss. For bulk data migration in data centers, the network needs to support on-demand, large-bandwidth allocation. In each of the aforementioned vertical industry scenarios, the bandwidth shall be adjusted as required to ensure flexible and efficient network resource usage.

### **2.5. End-to-end network slicing**

In an end-to-end network slicing scenario such as 5G network slicing [TS.28.530-3GPP], an IETF network slice [I-D.ietf-teas-ietf-network-slices] provides the required connectivity between other different segments of an end-to-end network slice, such as the Radio Access Network (RAN) and the Core Network (CN) segments, with a specific performance commitment. An IETF network slice could be composed of network slices from multiple technological and administrative domains. An IETF network slice can be realized by using or combining multiple underlying OTN slices with OTN resources, e.g. ODU time slots or ODU containers, to achieve end-to-end slicing across the transport domain.

### **3. Framework for OTN slicing**

OTN slices may be abstracted differently depending on the requirement contained in the configuration provided by the slice customer. Whereas the customer requests an OTN slice to provide connectivities between specified endpoints, an OTN slice can be abstracted as a set of endpoint-to-endpoint links, with each link formed by an end-to-end tunnel across the underlying OTN networks. The resources associated with each link of the slice is reserved and commissioned in the underlying physical network upon the completion of configuring the OTN slice and all the links are active.

An OTN slice can also be abstracted as an abstract topology when the customer requests the slice to share resources between multiple endpoints and to use the resources on demand. The abstract topology may consist of virtual nodes and virtual links, whose associated resources are reserved but not commissioned across the underlying OTN networks. The customer can later commission resources within the slice dynamically using the NBI provided by the service provider. An OTN slice could use abstract topology to connect endpoints with shared resources to optimize the resource utilization, and connections can be activated within the slice as needed.

It is worth noting that those means to abstract an OTN slice are similar to the Virtual Network (VN) abstraction defined for higher-level interfaces in [RFC8453], in which context a connectivity-based

slice corresponds to Type 1 VN and a resource-based slice corresponds to Type 2 VN, respectively.

A particular resource in an OTN network, such as a port or link, may be sliced with one of the two granularity levels:

- \*Link-based slicing, in which a link and its associated link termination points (LTPs) are dedicatedly allocated to a particular OTN slice.
- \*Tributary-slot based slicing, in which multiple OTN slices share the same link by allocating different OTN tributary slots in different granularities.

Furthermore, an OTN switch is typically fully non-blockable switching at the lowest ODU container granularity, it is desirable to specify just the total number of ODU containers in the lowest granularity (e.g. ODU0), when configuring tributary-slot based slicing on links and ports internal to an OTN network. In multi-domain OTN network scenarios where separate OTN slices are created on each of the OTN networks and are stitched at inter-domain OTN links, it is necessary to specify matching tributary slots at the endpoints of the inter-domain links. In some real network scenarios, OTN network resources including tributary slots are managed explicitly by network operators for network maintenance considerations. Therefore an OTN slice controller shall support configuring an OTN slice with both options.

An OTN slice controller (OTN-SC) is a logical function responsible for the life-cycle management of OTN slices instantiated within the corresponding OTN network domains. The OTN-SC provides technology-specific interfaces at its northbound (OTN-SC NBI) to allow a higher-layer slice controller, such as an IETF network slice controller (NSC), or an orchestrator, to request OTN slices with OTN-specific requirements. The OTN-SC interfaces at the southbound using the MDSC-to-PNC interface (MPI) with a Physical Network Controller (PNC) or Multi-Domain Service Orchestrator (MDSC), as defined in the ACTN control framework [[RFC8453](#)]. The logical function within the OTN-SC is responsible for translating the OTN slice requests into concrete slice realization which can be understood and provisioned at the southbound by the PNC or MDSC.

When realizing OTN slices, the OTN-SC may translate a connectivity-based OTN slice into a set of end-to-end tunnels using the Traffic-engineering(TE) tunnel interface defined in [[I-D.ietf-teas-yang-te](#)]. For a resource-based OTN slice, the OTN-SC may translate the abstract topology representing the slice into a colored graph on an abstract TE topology using the TE topology interface defined in [[RFC8795](#)].

The OTN-SC NBI is technology-specific, while the IETF NSC-NBI is technology- agnostic. An IETF NSC may translate its customer's technology-agnostic slice request into an OTN slice request and utilize the OTN-SC NBI to realize the IETF network slice. Alternatively, the IETF NSC may translate the slicing request into tunnel or topology configuration commands and communicate directly with the underlying PNC or MDSC to provision the IETF network slice.

[Figure 2](#) illustrates the OTN slicing control hierarchy and the positioning of the OTN slicing interfaces.

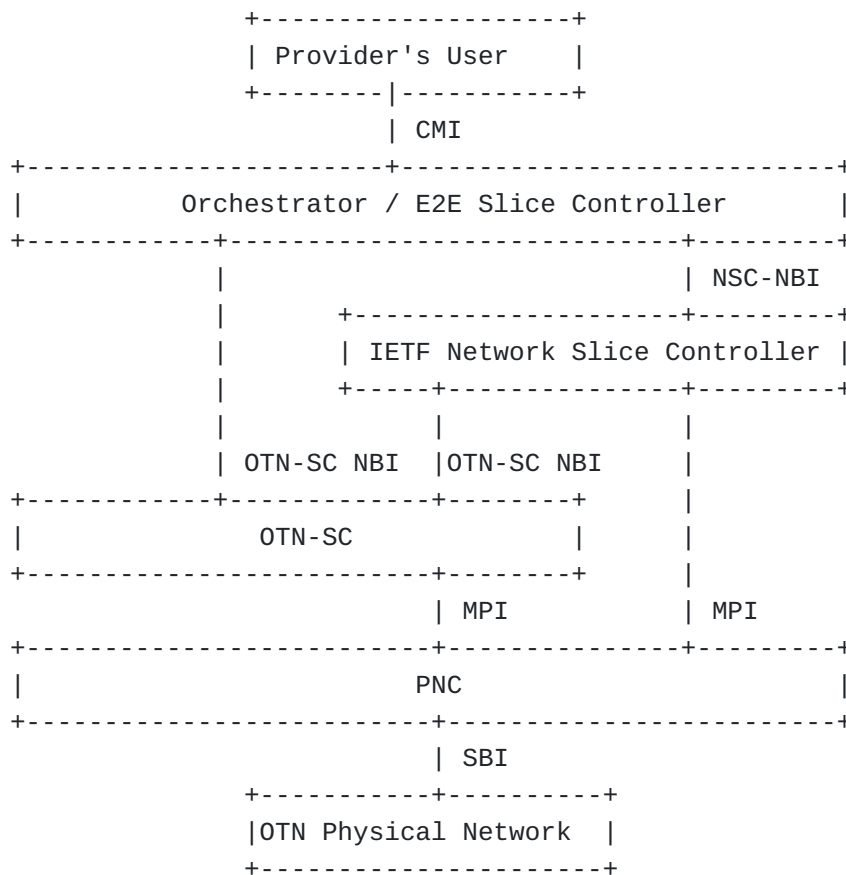


Figure 2: Positioning of OTN Slicing Interfaces

OTN-SC functionalities may be recursive such that a higher-level OTN-SC may designate the creation of OTN slices to a lower-level OTN-SC in a recursive manner. This scenario may apply to the creation of OTN slices in multi-domain OTN networks, where multiple domain-wide OTN slices provisioned by lower-layer OTN-SCs are stitched to support a multi-domain OTN slice provisioned by the higher-level OTN-SC. Alternatively, the OTN-SC may interface with an MDSC, which in turn interfaces with multiple PNCs through the MPI to realize OTN slices in multi-domain OTN networks without OTN-SC



recursion. [Figure 3](#) illustrates both options for OTN slicing in multi-domain.

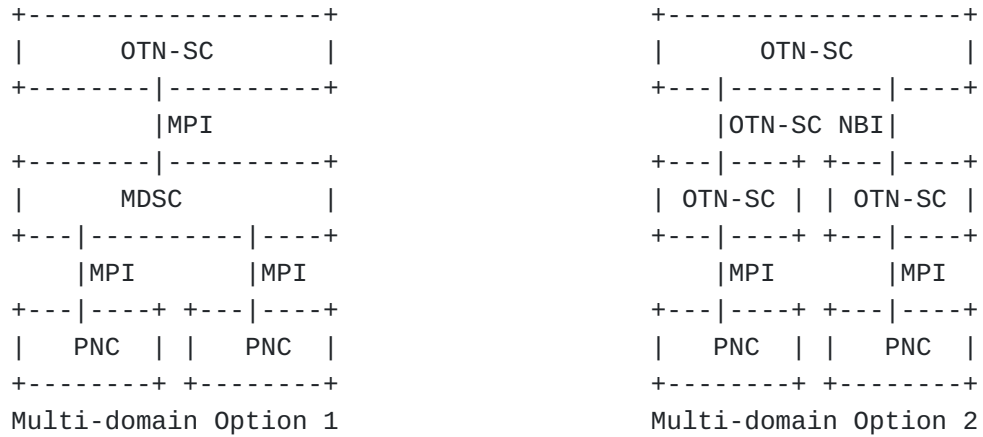


Figure 3: OTN-SC for multi-domain

OTN-SC functionalities are logically independent and may be deployed in different combinations to cater to the realization needs. In reference with the ACTN control framework [[RFC8453](#)], an OTN-SC may be deployed - as an independent network function; - together with a Physical Network Controller (PNC) for single-domain or with a Multi-Domain Service Orchestrator (MDSC) for multi-domain; - together with a higher-level network slice controller to support end-to-end network slicing;

#### 4. YANG Data Model for OTN Slicing Configuration

##### 4.1. OTN Slicing YANG Model for MPI

###### 4.1.1. MPI YANG Model Overview

For the configuration of connectivity-based OTN slices, existing models such as the TE tunnel interface [[I-D.ietf-teas-yang-te](#)] may be used and no addition is needed. This model is addressing the case for configuring resource-based OTN slices, where the model permits to reserve resources exploiting the common knowledge of an underlying virtual topology between the OTN-SC and the subtended network controller (MDSC or PNC). The slice is configured by marking corresponding link resources on the TE topology received from the underlying MDSC or PNC with a slice identifier and OTN-specific resource requirements, e.g. the number of ODU time slots or the type/number of ODU containers. The MDSC or PNC, based on the marked resources by the OTN-SC, will update the underlying TE topology with new TE link for each of the colored links to keep booked the reserved OTN resources e.g. time slots or ODU containers.

#### 4.1.2. MPI YANG Model Tree

module: ietf-otn-slice

```
augment /nw:networks/nw:network/nt:link/tet:te/tet:te-link-attributes:
  +--rw (otn-slice-granularity)?
    +--:(link)
      | +--rw slice-id?      uint32
    +--:(link-resource)
      +--rw slices* [slice-id]
        +--rw slice-id          uint32
        +--rw (technology)?
          | +--:(otn)
          |   +--rw (slice-bandwidth)?
          |   | +--:(containers)
          |   |   +--rw odulist* [odu-type]
          |   |   | +--rw odu-type  identityref
          |   |   | +--rw number?   uint16
          |   |   +--:(time-slots)
          |   |   +--rw otn-ts-num?  uint32
          +--ro sliced-link-ref?    -> ../../../../../../nt:link/1
```

Figure 4: OTN slicing tree diagram

#### 4.1.3. MPI YANG Code

<CODE BEGINS> file "ietf-otn-slice@2021-10-22.yang"

```
module ietf-otn-slice {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-otn-slice";
  prefix "otnslice";

  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-te-topology {
    prefix "tet";
    reference
      "RFC8795: YANG Data Model for Traffic Engineering
      (TE) Topologies";
  }

  import ietf-otn-topology {
    prefix "otntopo";
    reference
      "I-D.ietf-ccamp-otn-topo-yang: A YANG Data Model
      for Optical Transport Network Topology";
  }

  import ietf-layer1-types {
    prefix "l1-types";
    reference
      "I-D.ietf-ccamp-layer1-types: A YANG Data Model
      for Layer 1 Types";
  }

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

    Editor: Haomian Zheng
           <mailto:zhenghaomian@huawei.com>

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

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

Editor: Victor Lopez  
<mailto:victor.lopezalvarez@telefonica.com>;

description

"This module defines a YANG data model to configure an OTN network slice realization.

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

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

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<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 "2021-10-22" {  
  description  
    "Latest revision of MPI YANG model for OTN slicing.";  
  reference  
    "draft-zheng-ccamp-yang-otn-slicing-03: Framework and Data  
    Model for OTN Network Slicing";  
}
```

```
/*  
 * Groupings  
 */
```

```
grouping otn-link-slice-profile {  
  description  
    "Profile of an OTN link slice.";  
  choice otn-slice-granularity {  
    default "link";  
    description  
      "Link slice granularity.";  
    case link {  
      leaf slice-id {  
        type uint32;  
        description  
          "Slice identifier";  
      }  
    }  
  }  
}
```

```

}
case link-resource {
  list slices {
    key slice-id;
    description
      "List of slices.";
    leaf slice-id {
      type uint32;
      description
        "Slice identifier";
    }
    choice technology {
      description
        "Data plane technology types.";
      case otn {
        choice slice-bandwidth {
          description
            "Bandwidth specification for OTN slices.";
          case containers {
            uses l1-types:otn-link-bandwidth;
          }
          case time-slots {
            leaf otn-ts-num {
              type uint32;
              description
                "Number of OTN tributary slots allocated for the
                 slice.";
            }
          }
        }
      }
    }
  }
  leaf sliced-link-ref {
    type leafref {
      path "../..../nt:link/nt:link-id";
    }
    config false;
    description
      "Relative reference to virtual links generated from
       this TE link.";
  }
}
}
}
}
}
/*
 * Augments
 */

```

```

augment "/nw:networks/nw:network/nt:link/tet:te/"
  + "tet:te-link-attributes" {
    when "../../../nw:network-types/tet:te-topology/"
      + "otntopo:otn-topology" {
        description
          "Augmentation parameters apply only for networks with
            OTN topology type.";
      }
    description
      "Augment OTN TE link attributes with slicing profile.";
    uses otn-link-slice-profile;
  }
}

```

<CODE ENDS>

Figure 5: OTN slicing YANG model

## 4.2. OTN Slicing YANG Model for OTN-SC NBI

### 4.2.1. NBI YANG Model Overview

The YANG model for OTN-SC NBI is OTN-technology specific, but shares many common constructs and attributes with generic network slicing YANG models. Furthermore, the OTN-SC NBI YANG is expected to support both connectivity-based and resource-based slice configuration, which is likely a common requirement for supporting slicing at other transport network layers, e.g. WDM or MPLS-TP. Therefore, the OTN-SC NBI YANG model is designed into two models, a common base model for transport network slicing, and an OTN slicing model which augments the base model with OTN technology-specific constructs.

The base model defines a transport network slice (TNS) with the following constructs and attributes:

- Common attributes, which include a set of common attributes like slice identifier, name, description and names of customers who use the slice.
- Endpoints, which represent conceptual points of connection from a customer device to the TNS. An endpoint is mapped to specific physical or virtual resources of the customer and provider, and such mapping is pre-negotiated and known to both the customer and provider prior to the slice configuration. The mechanism for endpoint negotiation is outside the scope of this draft.
- Network topology, which represent set of shared, reserved resources organized as a virtual topology between all of the endpoints. A customer could use such network topology to define detailed connectivity path traversing the topology, and allow sharing of resources between its multiple endpoint pairs.
- Connectivity matrix, which represent the intended virtual connections between the endpoints within a TNS. A connectivity matrix entry could be associated with an explicit path

over the above network topology. - Service-level objectives (SLOs) associated with different objects, including the TNS, node, link, termination point, and explicit path, within a TNS.

#### **4.2.2. NBI YANG Model Tree for Transport Network Slice**

```

module: ietf-transport-network-slice
+--rw network-slices
  +--rw network-slice* [ns-id]
    +--rw ns-id string
    +--rw ns-name? string
    +--rw ns-description? string
    +--rw customer-name* string
    +--rw slo
      | +--rw optimization-criterion? identityref
      | +--rw delay-tolerance? boolean
      | +--rw periodicity* uint64
      | +--rw isolation-level? identityref
    +--rw endpoints
      | +--rw endpoint* [endpoint-id]
      |   +--rw endpoint-id string
    +--rw network-topologies
      | +--rw network-topology* [topology-id]
      |   +--rw topology-id string
      |   +--rw node* [node-id]
      |     | +--rw node-id inet:uri
      |     | +--rw slo
      |     | | +--rw isolation-level? identityref
      |     | | +--rw termination-point* [tp-id]
      |     | |   +--rw tp-id inet:uri
      |     | |   +--rw endpoint-id? leafref
      |   +--rw link* [link-id]
      |     +--rw link-id inet:uri
      |     +--rw slo
      |       | +--rw delay-tolerance? boolean
      |       | +--rw periodicity* uint64
      |       | +--rw isolation-level? identityref
      |       +--rw source
      |         | +--rw source-node? -> ../../../../node/node-id
      |         | +--rw source-tp? leafref
      |       +--rw destination
      |         +--rw dest-node? -> ../../../../node/node-id
      |         +--rw dest-tp? leafref
    +--rw connectivity-matrices
      +--rw connectivity-matrix* [connectivity-matrix-id]
        +--rw connectivity-matrix-id uint32
        +--rw topology-id? leafref
        +--rw src-endpoint?
          | -> ../../../../endpoints/endpoint/endpoint-id
        +--rw dst-endpoint?
          | -> ../../../../endpoints/endpoint/endpoint-id
        +--rw slo
        +--rw explicit-path* [tp-id]
          +--rw tp-id leafref

```



Figure 6: Tree diagram for transport network slice

#### 4.2.3. NBI YANG Code for Transport Network Slice

<CODE BEGINS> file "ietf-transport-network-slice@2021-10-22.yang"

```
module ietf-transport-network-slice {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-transport-network-slice";
  prefix "tns";

  import ietf-inet-types {
    prefix inet;
    reference "RFC 6991";
  }

  import ietf-te-types {
    prefix "te-types";
    reference
      "RFC 8776: Traffic Engineering Common YANG Types";
  }

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

    Editor: Haomian Zheng
           <mailto:zhenghaomian@huawei.com>

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

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

    Editor: Victor Lopez
           <mailto:victor.lopezalvarez@telefonica.com>";

  description
    "This module defines a YANG data model to configure an OTN
    network slice realization.

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

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

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject
    to the license terms contained in, the Simplified BSD License
    set forth in Section 4.c of the IETF Trust's Legal Provisions
```

Relating to IETF Documents  
(<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 "2021-10-22" {
  description
    "Latest revision of NBI YANG model for OTN slicing.";
  reference
    "draft-zheng-ccamp-yang-otn-slicing-03: Framework and Data
    Model for OTN Network Slicing";
}

/*
 * Identities
 */
identity isolation-level {
  description
    "Base identity for the isolation-level.";
  reference
    "GSMA-NS-Template: Generic Network Slice Template,
    Version 3.0.";
}
identity no-isolation {
  base isolation-level;
  description
    "Network slices are not separated.";
}
identity physical-isolation {
  base isolation-level;
  description
    "Network slices are physically separated (e.g. different rack,
    different hardware, different location, etc.).";
}
identity logical-isolation {
  base isolation-level;
  description
    "Network slices are logically separated.";
}
identity process-isolation {
  base physical-isolation;
  description
    "Process and threads isolation.";
}
identity physical-memory-isolation {
  base physical-isolation;
  description
    "Process and threads isolation.";
}
```

```

identity physical-network-isolation {
    base physical-isolation;
    description
        "Process and threads isolation.";
}
identity virtual-resource-isolation {
    base logical-isolation;
    description
        "A network slice has access to specific range of resources
        that do not overlap with other network slices
        (e.g. VM isolation).";
}
identity network-functions-isolation {
    base logical-isolation;
    description
        "NF (Network Function) is dedicated to the network slice, but
        virtual resources are shared.";
}
identity service-isolation {
    base logical-isolation;
    description
        "NSC data are isolated from other NSCs, but virtual
        resources and NFs are shared.";
}

/*
 * Groupings
 */

grouping ns-generic-info {
    description
        "Generic configuration of a network slice";
    leaf ns-name {
        type string;
        description
            "Name of the specific network slice";
    }
    leaf ns-description {
        type string;
        description
            "Description regarding the specific network slice";
    }
    leaf-list customer-name {
        type string;
        description
            "List of customers using the slice";
    }
}

```

```

grouping ns-slo {
  description
    "SLO configuration of a network slice";

  container slo {
    description
      "SLO configuration of a network slice";

    leaf optimization-criterion {
      type identityref {
        base te-types:objective-function-type;
      }
      description
        "Optimization criterion applied to this topology.";
    }
    leaf delay-tolerance {
      type boolean;
      description
        "'true' if is not too critical how long it takes to deliver
        the amount of data.";
      reference
        "GSMA-NS-Template: Generic Network Slice Template,
        Version 3.0.";
    }
    leaf-list periodicity {
      type uint64;
      units seconds;
      description
        "A list of periodicities supported by the network slice.";
      reference
        "GSMA-NS-Template: Generic Network Slice Template,
        Version 3.0.";
    }
    leaf isolation-level {
      type identityref {
        base isolation-level;
      }
      description
        "A network slice instance may be fully or partly, logically
        and/or physically, isolated from another network slice
        instance. This attribute describes different types of
        isolation:";
    }
  }
}

grouping node-slo {
  description
    "Node SLO";
}

```

```

container slo {
  description
    "SLO configuration of a node";
  leaf isolation-level {
    type identityref {
      base isolation-level;
    }
    description
      "A network slice instance may be fully or partly, logically
      and/or physically, isolated from another network slice
      instance. This attribute describes different types of
      isolation:";
  }
}

```

```

grouping link-slo {
  description
    "Link SLO";
  container slo {
    description
      "SLO configuration of a link";
    leaf delay-tolerance {
      type boolean;
      description
        "'true' if is not too critical how long it takes to deliver
        the amount of data.";
      reference
        "GSMA-NS-Template: Generic Network Slice Template,
        Version 3.0.";
    }
    leaf-list periodicity {
      type uint64;
      units seconds;
      description
        "A list of periodicities supported by the network slice.";
      reference
        "GSMA-NS-Template: Generic Network Slice Template,
        Version 3.0.";
    }
    leaf isolation-level {
      type identityref {
        base isolation-level;
      }
      description
        "A network slice instance may be fully or partly, logically
        and/or physically, isolated from another network slice
        instance. This attribute describes different types of
        isolation:";
    }
  }
}

```

```

    }
  }
}

grouping connectivity-matrix-slo {
  description
    "SLO configuration of a path within a network slice";

  container slo {
    description
      "Path SLO configuration";
  }
  leaf delay-tolerance {
    type boolean;
    description
      "'true' if is not too critical how long it takes to deliver
      the amount of data.";
    reference
      "GSMA-NS-Template: Generic Network Slice Template,
      Version 3.0.";
  }
  leaf-list periodicity {
    type uint64;
    units seconds;
    description
      "A list of periodicities supported by the network slice.";
    reference
      "GSMA-NS-Template: Generic Network Slice Template,
      Version 3.0.";
  }
  leaf isolation-level {
    type identityref {
      base isolation-level;
    }
    description
      "A network slice instance may be fully or partly, logically
      and/or physically, isolated from another network slice
      instance. This attribute describes different types of
      isolation:";
  }
}

grouping connectivity-matrix-entry-slo {
  description
    "SLO configuration of a connectivity matrix entry within a
    network slice";

  container slo {
    description

```

```

        "SLO configuration of a connectivity matrix entry";
    }
}

grouping explicit-path {
    description
        "Explicit path for a connectivity matrix entry";

    list explicit-path {
        key "tp-id";
        description
            "List of TPs within a network topology that form a path.";
        leaf tp-id {
            type leafref {
                path "/network-slices/network-slice[ns-id=current()"+
                    ".../ns-id]/network-topologies"+
                    "/network-topology[topology-id=current()"+
                    ".../topology-id]/node/termination-point/tp-id";
            }
            description
                "Relative reference to TP id.";
        }
    }
}

grouping network-topology-def {
    description
        "Network topology definition";
    list node {
        key "node-id";
        description
            "The inventory of nodes of this topology.";
        leaf node-id {
            type inet:uri;
            description
                "Node identifier.";
        }
    }
    uses node-slo;
    list termination-point {
        key "tp-id";
        description
            "TP identifier";
        leaf tp-id {
            type inet:uri;
            description
                "Termination point identifier.";
        }
        leaf endpoint-id {
            type leafref {

```



```

        path "/network-slices/network-slice[ns-id=current()"+
            "../../../ns-id]/endpoints/endpoint/"+
            "endpoint-id";
    }
    description
        "Relative reference to TP id.";
}
}
list link {
    key "link-id";
    description
        "Link identifier.";
    leaf link-id {
        type inet:uri;
        description
            "Link identifier.";
    }
    uses link-slo;
    container source {
        description
            "Link source node";
        leaf source-node {
            type leafref {
                path "../../../node/node-id";
            }
            description
                "Source node identifier, must be in same topology.";
        }
        leaf source-tp {
            type leafref {
                path "../../../node[node-id=current()]/../"+
                    "source-node]/termination-point/tp-id";
            }
            description
                "Termination point within source node that terminates
                the link.";
        }
    }
}
container destination {
    description
        "Link destination node";
    leaf dest-node {
        type leafref {
            path "../../../node/node-id";
        }
        description
            "Destination node identifier, must be in same topology.";
    }
}

```

```

    leaf dest-tp {
        type leafref {
            path "../../../../../node[node-id=current()../../"+
                "dest-node]/termination-point/tp-id";
        }
        description
            "Termination point within destination node that terminates
            the link.";
    }
}
}
}

/*
 * Configuration data nodes
 */
container network-slices {
    description
        "Generic network slice configurations";
    list network-slice {
        key "ns-id";
        description
            "Network slice identifier";
        leaf ns-id {
            type string;
            description
                "A unique network slice identifier across a slice controller";
        }
        uses ns-generic-info;
        uses ns-slo;

        container endpoints {
            description
                "Endpoints of a network slice";

            list endpoint {
                key "endpoint-id";
                description
                    "List of endpoints";
                leaf endpoint-id {
                    type string;
                    description
                        "Endpoint identifier";
                }
            }
        }
    }
}
container network-topologies {
    description
        "A network slice is described as a network topology";
}

```

```

list network-topology {
    key "topology-id";
    description
        "List of network topologies";
    leaf topology-id {
        type string;
        description
            "Topology identifier";
    }
    uses network-topology-def;
}
}
container connectivity-matrices {
    description
        "Connectivity matrices";

    list connectivity-matrix {
        key "connectivity-matrix-id";
        description
            "List of connectivity matrix entities";
        leaf connectivity-matrix-id {
            type uint32;
            description
                "Connectivity matrix identifier";
        }
        leaf topology-id {
            type leafref {
                path "../../network-topologies/network-topology/topology-id";
            }
            description
                "Relative reference to network topology id.";
        }
        leaf src-endpoint {
            type leafref {
                path "../../endpoints/endpoint/endpoint-id";
            }
            description
                "Relative reference to endpoint id.";
        }
        leaf dst-endpoint {
            type leafref {
                path "../../endpoints/endpoint/endpoint-id";
            }
            description
                "Relative reference to endpoint id.";
        }
        uses connectivity-matrix-entry-slo;
        uses explicit-path;
    }
}

```

```

        } //connectivity-matrix
    } //connectivity-matrices
} //network-slice
} //network slices
}

```

<CODE ENDS>

Figure 7: YANG model for transport network slice

#### 4.2.4. NBI YANG Model Tree for OTN slice

TBD.

#### 4.2.5. NBI YANG Code for OTN Slice

TBD.

### 5. 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 optical 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.

### 6. Security Considerations

<Add any security considerations>

### 7. IANA Considerations

<Add any IANA considerations>

### 8. References

#### 8.1. Normative References

[I-D.ietf-teas-yang-te] Saad, T., Gandhi, R., Liu, X., Beeram, V. P., Bryskin, I., and O. G. D. Dios, "A YANG Data Model for Traffic Engineering Tunnels, Label Switched Paths and Interfaces", Work in Progress, Internet-Draft, draft-ietf-teas-yang-te-27, 8 July 2021, <<https://www.ietf.org/archive/id/draft-ietf-teas-yang-te-27.txt>>.

[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>>.

[TS.28.530-3GPP] 3rd Generation Partnership Project (3GPP), "3GPP TS 28.530 V15.1.0 Technical Specification Group Services and System Aspects; Management and orchestration; Concepts, use cases and requirements (Release 15)", 3GPP TS 28.530, December 2018, <[http://ftp.3gpp.org/Specs/archive/28\\_series/28.530/28530-f10.zip](http://ftp.3gpp.org/Specs/archive/28_series/28.530/28530-f10.zip)>.

## 8.2. Informative References

### [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-04, 23 August 2021, <<https://www.ietf.org/archive/id/draft-ietf-teas-ietf-network-slices-04.txt>>.

[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>>.

## Acknowledgments

This document was prepared using kramdown.

Previous versions of this document were prepared using 2-Word-v2.0.template.dot.

## Contributors' Addresses

Henry Yu  
Huawei Technologies Canada

Email: [henry.yu@huawei.com](mailto:henry.yu@huawei.com)

Jiang Sun  
China Mobile

Email: [sunjiang@chinamobile.com](mailto:sunjiang@chinamobile.com)

## Authors' Addresses

Haomian Zheng  
Huawei Technologies  
H1, Xiliu Beipo Village, Songsshan Lake

Dongguan  
China

Email: [zhenghaomian@huawei.com](mailto:zhenghaomian@huawei.com)

Italo Busi  
Huawei Technologies

Email: [italo.busi@huawei.com](mailto:italo.busi@huawei.com)

Aihua Guo  
Futurewei Technologies

Email: [aihuaguo.ietf@gmail.com](mailto:aihuaguo.ietf@gmail.com)

Luis M. Contreras  
Telefonica

Email: [luismiguel.contrerasmurillo@telefonica.com](mailto:luismiguel.contrerasmurillo@telefonica.com)

Oscar Gonzalez de Dios  
Telefonica

Email: [oscar.gonzalezdedios@telefonica.com](mailto:oscar.gonzalezdedios@telefonica.com)

Victor Lopez  
Nokia

Email: [victor.lopez@nokia.com](mailto:victor.lopez@nokia.com)

Sergio Belotti  
Nokia

Email: [Sergio.belotti@nokia.com](mailto:Sergio.belotti@nokia.com)

Dieter Beller  
Nokia

Email: [Dieter.Beller@nokia.com](mailto:Dieter.Beller@nokia.com)

Reza Rokui  
Nokia

Email: [reza.rokui@nokia.com](mailto:reza.rokui@nokia.com)

Yunbin Xu  
CAICT

Email: [xuyunbin@caict.ca.cn](mailto:xuyunbin@caict.ca.cn)

Yang Zhao

China Mobile

Email: [zhaoyangyiy@chinamobile.com](mailto:zhaoyangyiy@chinamobile.com)

Xufeng Liu

Volta Networks

Email: [xufeng.liu.ietf@gmail.com](mailto:xufeng.liu.ietf@gmail.com)