Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 13, 2021

B. Wu
D. Dhody
Huawei Technologies
L. Han
China Mobile
R. Rokui
Nokia Canada

July 12, 2020

A Yang Data Model for Transport Slice NBI
draft-wd-teas-transport-slice-yang-02

Abstract

This document provides a YANG data model for the Transport Slice NBI. The model can be used by a higher level system which is the Transport slice consumer of a Transport Slice Controller (TSC) to request, configure, and manage the components of a transport slices.

The YANG modules in this document conforms to the Network Management Datastore Architecture (NMDA) defined in RFC 8342.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 13, 2021.

Copyright Notice

Copyright (c) 2020 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

This document provides a YANG [RFC7950] data model for the transport Slice NBI.

The YANG model discussed in this document is defined based on the description of the transport slice in [I-D.nsdt-teas-transport-slice-definition] and [I-D.nsdt-teas-ns-framework], which is used to operate customer-

driven Transport Slice during the Transport Slice instantiation, and the operations includes modification, deletion, and monitoring.

The YANG model discussed in this document describes the requirements of a Transport Slice that interconnects a set of Transport Slice

Endpoints from the point of view of the consumer, which is classified as Customer Service Model in [RFC8309].

It will be up to the management system or TSC (Transport Slice controller) to take this model as an input and use other management system or specific configuration models to configure the different network elements to deliver a Transport Slice. The YANG models can be used with network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. How the configuration of network elements is done is out of scope for this document.

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

2. Conventions used in this document

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

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

- client
- configuration data
- state data

This document also makes use of the following terminology introduced in the YANG 1.1 Data Modeling Language [RFC7950]:

- augment

o data model

o data node

This document also makes use of the following terminology introduced in the Transport Slice definition draft [I-D.nsdt-teas-transport-slice-definition]:

o Transport Slice: A transport slice is a logical network topology connecting a number of endpoints and a set of shared or dedicated network resources, which are used to satisfy specific Service

Level Objectives (SLO). The definition is from Section 3 of [I-D.nsdt-teas-transport-slice-definition].

o Transport Slice Endpoint (TSE): A Transport Slice Endpoint is a logical identifier at an external interface of Transport Network to identify the logical access to which, a particular subset of traffic traversing the external interface, is mapped to a specific TS and it follows the definition of TSE (Transport Slice Endpoint) in Section 4.2 of [I-D.nsdt-teas-transport-slice-definition].

o SLO: An SLO is a service level objective

o DAN: Device,Application,Network Function

o TSC: Transport Slice Controller

o NBI: NorthBound Interface

In addition, this document defines the following terminology:

o Transport Slice Member (TS-Member): A TS member is an abstract entity which represents the transport resources mapped to a particular connection between a pair of TSEs belonging to a Transport slice. Note that different SLO requirement per-TS-Member could be applied.

o TS-SLO-Group: Indicates a group of TS-members with same SLOs in one transport slice.

2.1. Tree Diagrams

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

3. Transport Slice NBI Model Usage

The intention of the transport slice NBI model is to allow the consumer, e.g. A higher level management system, to request and monitor transport slices. In particular, the model allows consumers to operate in an abstract, technology-agnostic manner, with implementation details hidden.

In the use case of 5G transport application, the E2E network slice orchestrator acts as the higher layer system to request the transport slices. The interface is used to support dynamic transport slice creation and its lifecycle management to facilitate end-to-end network slice services.


```
             +------------------------------------------+
             |   Transport Slice Consumer               |
             |                                          |
             +----------------+-------------------------+
                              |
                              | Transport Slice NBI YANG
                              |
                              |
        +---------------------+--------------------------+
        |        Transport Slice Controller              |
        +------------------------------------------------+
```

Figure 1 Transport Slice NBI Model Context

4. Transport Slice NBI Model Overview

From a consumer perspective, an example of a transport slice is shown in figure 2.

```
                        Transport Network
    DAN1        +----------------------------------+        DAN3
  +---------+   |                                  |    +---------+
```

```
   |        /--\  |                                 |  /--\      |
   |       |TSE1+-+                                 +-+TSE3|      |
   +--------\--/  |                                 |  \--/      |
                  |                                 |    |       |
   +--------+     |                                      |       |
   |        /--\  |                                 |  /--\      |
   |       |TSE2+-+                                 +-+TSE4|      |
   |        \--/  |                                 |  \--/      |
   +--------+     |                                 |    +--------+
     DAN2         +---------------------------------+
            |                                            |
            |                                            |
            |<------------Transport Slice 1------------->|

   Legend:DAN (Device,Application,Network Function)

           TS-SLO-Group Red                  TS-SLO-Group Blue
         TS-Member 2     TSE1-TSE3          TS-Member 1     TSE1-TSE2
         TS-Member 3     TSE1-TSE4
         TS-Member 4     TSE2-TSE3
         TS-Member 5     TSE2-TSE4
```

Figure 2: An example of TSEs and TS-Members of a transport slice

As shown in figure 2, a Transport Slice (TS) links together TSEs at external Interfaces to the DANs, which are customer endpoints that

request a transport slice. At each customer DAN, one or multiple TSEs could be connected to the Transport Slice.

A TS is a connectivity service with specific SLO characteristics, including bandwidth, QoS metric, etc. The connectivity service is a combination of logical connections, represented by TS-members. When some parts of a slice have different SLO requirements, a group of TS-Members with the same SLO is described by TS-SLO-Group.

Based on this design, the Transport Slice YANG module consists of the main containers: "transport-slice", "ts-endpoint", "ts-member",and "ts-slo-group".

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

```
module: ietf-transport-slice
  +--rw transport-slices
     +--rw slice-templates
     |  +--rw slice-template* [id]
     |     +--rw id                        string
     |     +--rw template-description?   string
     +--rw transport-slice* [ts-id]
        +--rw ts-id           uint32
        +--rw ts-name?        string
        +--rw ts-topology*    identityref
        +--rw ts-slo-group* [slo-group-name]
        |  +--rw slo-group-name          string
        |  +--rw default-slo-group?      boolean
        |  +--rw slo-tag?                string
        |  +--rw (slo-template)?
        |  |  +--:(standard)
        |  |  |  +--rw template?         leafref
        |  |  +--:(custom)
        |  |     +--rw ts-slo-policy
        |  |        +--rw latency
        |  |        |  +--rw one-way-latency?   uint32
        |  |        |  +--rw two-way-latency?   uint32
        |  |        +--rw jitter
        |  |        |  +--rw one-way-jitter?   uint32
        |  |        |  +--rw two-way-jitter?   uint32
        |  |        +--rw loss
        |  |        |  +--rw one-way-loss?   decimal64
        |  |        |  +--rw two-way-loss?   decimal64
        |  |        +--rw availability-type?   identityref
        |  |        +--rw isolation-type?      identityref
        |  +--rw ts-member-group* [ts-member-id]
        |  |  +--rw ts-member-id    leafref
        |  +--ro slo-group-monitoring
```



```
        |     +--ro latency?   uint32
        |     +--ro jitter?    uint32
        |     +--ro loss?      decimal64
        +--rw status
        |  +--rw admin-enabled?   boolean
        |  +--ro oper-status?     operational-type
        +--rw ts-endpoint* [ep-id]
        |  +--rw ep-id                uint32
```

```
           |  +--rw ep-name?              string
           |  +--rw ep-role*              identityref
           |  +--rw geolocation
           |  |  +--rw altitude?    int64
           |  |  +--rw latitude?    decimal64
           |  |  +--rw longitude?   decimal64
           |  +--rw node-id?              string
           |  +--rw port-id?                string
           |  +--rw ts-filter-criteria
           |  |  +--rw ts-filter-criteria* [match-type]
           |  |     +--rw match-type    identityref
           |  |     +--rw value?        string
           |  +--rw bandwidth
           |  |  +--rw incoming-bandwidth
           |  |  |  +--rw guaranteed-bandwidth?   te-types:te-bandwidth
           |  |  +--rw outgoing-bandwidth
           |  |     +--rw guaranteed-bandwidth?   te-types:te-bandwidth
           |  +--rw mtu                   uint16
           |  +--rw protocol
           |  |  +--rw bgp
           |  |  |  +--rw bgp-peer-ipv4*   inet:ipv4-prefix
           |  |  |  +--rw bgp-peer-ipv6*   inet:ipv6-prefix
           |  |  +--rw static
           |  |     +--rw static-route-ipv4*   inet:ipv4-prefix
           |  |     +--rw static-route-ipv6*   inet:ipv6-prefix
           |  +--rw status
           |  |  +--rw admin-enabled?   boolean
           |  |  +--ro oper-status?     operational-type
           |  +--ro ep-monitoring
           |     +--ro incoming-utilized-bandwidth?
           |     |       te-types:te-bandwidth
           |     +--ro incoming-bw-utilization        decimal64
           |     +--ro outgoing-utilized-bandwidth?
           |     |       te-types:te-bandwidth
           |     +--ro outgoing-bw-utilization        decimal64
           +--rw ts-member* [ts-member-id]
              +--rw ts-member-id            uint32
              +--rw src
              |  +--rw src-ts-ep-id?   leafref
              +--rw dest
```



```
              |  +--rw dest-ts-ep-id?   leafref
```

```
          +--rw monitoring-type?        ts-monitoring-type
          +--ro ts-member-monitoring
             +--ro latency?   uint32
             +--ro jitter?    uint32
             +--ro loss?      decimal64
```

5. Transport Slice NBI Model Description

A Transport Slice consists of a group of interconnected TSEs, and the connections between TSEs may have different SLO requirements, including symmetrical or asymmetrical traffic throughput, different traffic delay, etc.

5.1. Transport Slice Connection Pattern

A Transport Slice can be point-to-point (P2P), point-to-multipoint (P2MP), multipoint-to-point (MP2P), or multipoint-to-multipoint (MP2MP) based on the consumer's traffic pattern requirements.

Therefore, the "ts-topology" under the node "transport-slice" is required for configuration. The model supports any-to-any, Hub and Spoke (where Hubs can exchange traffic), and the different combinations. New topologies could be added via augmentation. By default, the any-to-any topology is used.

In addition, "ep-role" under the node "ts-endpoint" also needs to be defined, which specifies the role of the TSE in a particular TS topology. In the any-to-any topology, all TSEs MUST have the same role, which will be "any-to-any-role". In the Hub-and-Spoke topology, TSEs MUST have a Hub role or a Spoke role.

5.2. Transport Slice EndPoint (TSE)

A TSE belong to a single Transport Slice. A TS involves two or more TSEs.

A TSE is used to define the limit on the user traffic that can be injected to a TS. For example, in some scenarios, the access traffic of a DAN is allowed only when it matches the logical Layer 2 connection identifier. In some scenarios, the access traffic of a DAN is allowed only when the traffic matches a source IP address. Sometimes, the traffic from a distinct physical connection of a DAN is allowed.

Therefore, to ensure that the TSE is uniquely identified, the model use the following parameters including "node-id", "port-id" and "ts-

filter-criteria". The "node-id" identifies a DAN node, the "tp-id" identifies a port, and the "ts-filter-criteria" identifies a possible logical L2 ID or IP address or other possible traffic identifier in the user traffic.

Additionally, a number of slice interconnection parameters need to be agreed with a customer DAN and the transport network, such as IP address (v4 or v6) etc.

5.3. Transport Slice SLO

As defined in [I-D.nsdt-teas-transport-slice-definition]

This model defines the minimum Transport Slice SLO attributes, and other SLO nodes can be augmented as needed. TS SLO assurance is implemented through the following mechanisms:

- TS SLO list: Which defines the performance objectives of the TS. Performance objectives can be specified for various performance metrics,and different objectives are as follows:

  Latency: Indicates the maximum latency between two TSE. The unit is micro seconds. The latency could be round trip times or one-way metrics.

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

  Loss: Indicates maximum permissible packet loss rate, which is defined by the ratio of packets dropped to packets transmitted between two endpoints.

  Availability: Is defined as the ratio of up-time to total_time(up-time+down-time), where up-time is the time the transport slice is available in accordance with the SLOs associated with it.

  Isolation: Whether the isolation needs to be explicitly requested is still in discussion.

- Bandwidth: Indicates the guaranteed minimum bandwidth between any two TSE. The unit is data rate per second. And the bandwidth is unidirectional. The bandwidth is specified at each TSE and can be applied to incoming TS traffic or outgoing TS traffic. When applied in the incoming direction, the Bandwidth is applicable to

the traffic from the TSE to the Transport Network that passes through the external interface. When Bandwidth is applied to the

outgoing direction, it is applied to the traffic from the TN to the TSE of that particular TS.

Note: About the definition of SLO parameters, the author is discussing to reuse the TE-Types grouping definition as much as possible, to avoid duplication of definitions.

Consumers' Tranport Slices can be very different, e.g. some slices has the same SLO requirements of connections, some slices has the different SLO requirements for different parts of the slice. In some slices, the bandwidth of one endpoint is different from that of other endpoints, for example, one is central endpoint, the other endpoints are access endpoints.

The list "ts-slo-group" defines a group of different SLOs, which are used to describe that different parts of the slice have different SLOs. The specific SLO of the slice SLO group may use a standard SLO template, or may use different customized parameters. A group of "ts-member" is used to describe which connections of the slice use the SLO.

For some simplest Transport Slices, only one category SLO of "ts-slo-group" needs to be defined. For some complicated slices, in addition to the configurations above, multiple "ts-slo-group" needs to be defined, and "ts-member-group" under the "ts-slo-group" or "slo-group" under the "ts-member" describe details of the per-connection SLO.

In addition to SLO performance objectives, there are also some other TS objectives, such as MTU and security which can be augmented when needed. MTU specifies the maximum packet length that the slice guarantee to be able to carry across.

Note: In some use cases, the number of connections represented by "ts-member-group" may be huge, which may lead to configuration issues, for example, the scalability or error-prone.

6. Transport Slice Monitoring

This model also describes performance status of a transport slice. The statistics are described in the following granularity:

o Per TS SLO group: specified in 'ts-member-group-monitoring' under the "ts-slo-groupr"

o Per TS connection: specified in 'ts-member-monitoring' under the "ts-member"

o Per TS Endpoint: specified in 'ep-monitoring' under the "ts-endpoint"

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

By specifying subtree filters or xpath filters to 'ts-member' or 'endpoint' ,so that only interested contents will be sent. These mechanisms can be used for monitoring the transport slice performance status so that the client management system could initiate modification based on the transport slice running status.

7. Transport Slice NBI Model Usage Example

TBD

8. Transport Slice NBI Module

<CODE BEGINS> file "ietf-transport-slice@2020-07-12.yang"

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

  import ietf-inet-types {
    prefix inet;
  }
  import ietf-te-types {
    prefix te-types;
  }
```

```
  organization
    "IETF Traffic Engineering Architecture and Signaling (TEAS)
     Working Group";
  contact
    "WG Web:  <https://tools.ietf.org/wg/teas/>
     WG List:  <mailto:teas@ietf.org>
     Editor: Bo Wu <lana.wubo@huawei.com>
           : Dhruv Dhody <dhruv.ietf@gmail.com>";
  description
    "This module contains a YANG module for the Transport Slice NBI.

     Copyright (c) 2020 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
```



```
     without modification, is permitted pursuant to, and subject to
     the license terms contained in, the Simplified BSD License set
     forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

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

  revision 2020-07-12 {
    description
      "initial version.";
    reference
      "RFC XXXX: A Yang Data Model for Transport Slice NBI Operation";
  }

  /* Features */
  /* Identities */

  identity ts-topology {
    description
      "Base identity for Transport Slice topology.";
  }

  identity any-to-any {
```

```
    base ts-topology;
    description
      "Identity for any-to-any Transport Slice topology.";
  }

  identity hub-spoke {
    base ts-topology;
    description
      "Identity for Hub-and-Spoke Transport Slice topology.";
  }

  identity ep-role {
    description
      "TSE Role in a Transport Slice topology ";
  }

  identity any-to-any-role {
    base ep-role;
    description
      "TSE as the any-to-any role in an any-to-any Transport Slice.";
  }

  identity hub {
```



```
    base ep-role;
    description
      "TSE as the hub role in a Hub-and-Spoke Transport Slice.";
  }

  identity spoke {
    base ep-role;
    description
      "TSE as the spoke role in a Hub-and-Spoke transport slice.";
  }

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

  identity physical-isolation {
    base isolation-type;
```

```
    description
      "physical isolation.";
  }

  identity logical-isolation {
    base isolation-type;
    description
      "logical-isolation.";
  }

  identity ts-slo-metric-type {
    description
      "Base identity for TS SLO metric type";
  }

  identity ts-match-type {
    description
      "Base identity for TS metric type";
  }

  identity ts-vlan-match {
    base ts-match-type;
    description
      "logical-isolation.";
  }

  /*
   * Identity for availability-type
   */
```



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

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

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

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

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

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

  /* typedef */

  typedef operational-type {
    type enumeration {
      enum up {
        value 0;
        description
          "Operational status UP.";
      }
      enum down {
        value 1;
        description
          "Operational status DOWN";
```



```
      }
      enum unknown {
        value 2;
        description
```

```
          "Operational status UNKNOWN";
      }
    }
    description
      "This is a read-only attribute used to determine the
       status of a particular element";
  }

  typedef ts-monitoring-type {
    type enumeration {
      enum one-way {
        description
          "represents one-way monitoring type";
      }
      enum two-way {
        description
          "represents two-way monitoring type";
      }
    }
    description
      "enumerated type of monitoring on a ts-member ";
  }

  /* Groupings */

  grouping status-params {
    description
      "Grouping used to join operational and administrative status";
    container status {
      description
        "Container for status of administration and operational";
      leaf admin-enabled {
        type boolean;
        description
          "Administrative Status UP/DOWN";
      }
      leaf oper-status {
        type operational-type;
        config false;
        description
          "Operations status";
      }
    }
  }
```

```
  grouping ts-filter-criteria {
    description
      "Grouping for TS filter definition.";
    container ts-filter-criteria {
      description
        "Describes TS filter criteria.";
      list ts-filter-criteria {
        key "match-type";
        description
          "List of TS traffic criteria";
        leaf match-type {
          type identityref {
            base ts-match-type;
          }
          description
            "Identifies an entry in the list of match-type for the TS.";
        }
        leaf value {
          type string;
          description
            "Describes TS filter criteria,e.g. IP address, VLAN, etc.";
        }
      }
    }
  }

  grouping routing-protocols {
    description
      "Grouping for endpoint protocols definition.";
    container protocol {
      description
        "Describes protocol between TSE and transport network edge device.";
      container bgp {
        description
          "BGP-specific configuration.";
        leaf-list bgp-peer-ipv4 {
          type inet:ipv4-prefix;
          description
            "BGP peer ipv4 address.";
        }
        leaf-list bgp-peer-ipv6 {
          type inet:ipv6-prefix;
          description
            "BGP peer ipv6 address.";
        }
      }
      container static {
```

```
       description
```



```
         "Only applies when protocol is static.";
       leaf-list static-route-ipv4 {
         type inet:ipv4-prefix;
         description
           "ipv4 static route";
       }
       leaf-list static-route-ipv6 {
         type inet:ipv6-prefix;
         description
           "ipv6 static route";
       }
     }
   }
 }

 grouping ep-monitoring-parameters {
   description
     "Grouping for ep-monitoring-parameters.";
   container ep-monitoring {
     config false;
     description
       "Container for ep-monitoring-parameters.";
     leaf incoming-utilized-bandwidth {
       type te-types:te-bandwidth;
       description
         "Bandwidth utilization that represents the actual
          utilization of the incoming endpoint.";
     }
     leaf incoming-bw-utilization {
       type decimal64 {
         fraction-digits 5;
         range "0..100";
       }
       units "percent";
       mandatory true;
       description
         "To be used to define the bandwidth utilization
          as a percentage of the available bandwidth.";
     }
     leaf outgoing-utilized-bandwidth {
```

```
        type te-types:te-bandwidth;
        description
          "Bandwidth utilization that represents the actual
           utilization of the incoming endpoint.";
      }
      leaf outgoing-bw-utilization {
        type decimal64 {
          fraction-digits 5;
```



```
          range "0..100";
        }
        units "percent";
        mandatory true;
        description
          "To be used to define the bandwidth utilization
           as a percentage of the available bandwidth.";
      }
    }
  }

  grouping common-monitoring-parameters {
    description
      "Grouping for link-monitoring-parameters.";
    leaf latency {
      type uint32;
      units "usec";
      description
        "The latency statistics per TS member.";
    }
    leaf jitter {
      type uint32 {
        range "0..16777215";
      }
      description
        "The jitter statistics per TS member.";
    }
    leaf loss {
      type decimal64 {
        fraction-digits 6;
        range "0 .. 50.331642";
      }
      description
```

```
      "Packet loss as a percentage of the total traffic
       sent over a configurable interval. The finest precision is
       0.000003%. where the maximum 50.331642%.";
    reference
      "RFC 7810, section-4.4";
  }
}

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



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

// geolocation-container
```

```
  grouping endpoint {
    description
      "Transport Slice endpoint related information";
    leaf ep-id {
      type uint32;
      description
        "unique identifier for the referred Transport Slice endpoint";
    }
    leaf ep-name {
      type string;
      description
        "ep name";
    }
    leaf-list ep-role {
      type identityref {
        base ep-role;
      }
      default "any-to-any-role";
      description
        "Role of the endpoint in the Transport Slice.";
    }
```



```
    uses geolocation-container;
    leaf node-id {
      type string;
      description
        "Uniquely identifies an edge customer node.";
    }
    leaf port-id {
      type string;
      description
        "Reference to the Port-id of the customer node.";
    }
    uses ts-filter-criteria;
    container bandwidth {
      container incoming-bandwidth {
        leaf guaranteed-bandwidth {
          type te-types:te-bandwidth;
          description
            "If guaranteed-bandwidth is 0, it means best effort, no
             minimum throughput is guaranteed.";
```

```
        }
        description
          "Container for the incoming bandwidth policy";
      }
      container outgoing-bandwidth {
        leaf guaranteed-bandwidth {
          type te-types:te-bandwidth;
          description
            "If guaranteed-bandwidth is 0, it means best effort, no
             minimum throughput is guaranteed.";
        }
        description
          "Container for the bandwidth policy";
      }
      description
        "Container for the bandwidth policy";
    }
    leaf mtu {
      type uint16;
      units "bytes";
      mandatory true;
      description
        "MTU of TS traffic. If the traffic type is IP,
         it refers to the IP MTU. If the traffic type is Ethertype,
         will refer to the Ethernet MTU. ";
    }
    uses routing-protocols;
    uses status-params;
    uses ep-monitoring-parameters;
```



```
  }

  //ts-ep

  grouping ts-member {
    description
      "ts-member is described by this container";
    leaf ts-member-id {
      type uint32;
      description
        "ts-member identifier";
    }
```

```
    container src {
      description
        "the source of TS link";
      leaf src-ts-ep-id {
        type leafref {
          path "/transport-slices/transport-slice/ts-endpoint/ep-id";
        }
        description
          "reference to source TS endpoint";
      }
    }
    container dest {
      description
        "the destination of TS link ";
      leaf dest-ts-ep-id {
        type leafref {
          path "/transport-slices/transport-slice/ts-endpoint/ep-id";
        }
        description
          "reference to dest TS endpoint";
      }
    }
    leaf monitoring-type {
      type ts-monitoring-type;
      description
        "One way or two way monitoring type.";
    }
    container ts-member-monitoring {
      config false;
      description
        "SLO status Per ts endpoint to endpoint ";
      uses common-monitoring-parameters;
    }
  }

  //ts-member
```



```
  grouping transport-slice-slo-group {
    description
      "Grouping for SLO definition of TS";
    list ts-slo-group {
      key "slo-group-name";
```

```
      description
        "List of TS SLO groups, the SLO group is used to
         support different SLO objectives between different ts-members
         in the same slice.";
      leaf slo-group-name {
        type string;
        description
          "Identifies an entry in the list of SLO group for the TS.";
      }
      leaf default-slo-group {
        type boolean;
        default "false";
        description
          "Is the SLO group is selected as the default-slo-group";
      }
      leaf slo-tag {
        type string;
        description
          "slo tag for operational management";
      }
      choice slo-template {
        description
          "Choice for SLO template.
           Can be standard template or customized template.";
        case standard {
          description
            "Standard SLO template.";
          leaf template {
            type leafref {
              path "/transport-slices/slice-templates/slice-template/id";
            }
            description
              "QoS template to be used.";
          }
        }
        case custom {
          description
            "Customized SLO template.";
          container ts-slo-policy {
            container latency {
              leaf one-way-latency {
                type uint32 {
                  range "0..16777215";
```

```
        }
        units "usec";
        description
          "Lowest latency in micro seconds.";
      }
      leaf two-way-latency {
        type uint32 {
          range "0..16777215";
        }
        description
          "Lowest-way delay or latency in micro seconds.";
      }
      description
        "Latency constraint on the traffic class.";
    }
    container jitter {
      leaf one-way-jitter {
        type uint32 {
          range "0..16777215";
        }
        description
          "lowest latency in micro seconds.";
      }
      leaf two-way-jitter {
        type uint32 {
          range "0..16777215";
        }
        description
          "lowest-way delay or latency in micro seconds.";
      }
      description
        "Jitter constraint on the traffic class.";
    }
    container loss {
      leaf one-way-loss {
        type decimal64 {
          fraction-digits 6;
          range "0 .. 50.331642";
        }
        description
          "Packet loss as a percentage of the total traffic sent
           over a configurable interval. The finest precision is
           0.000003%. where the maximum 50.331642%.";
        reference
          "RFC 7810, section-4.4";
      }
      leaf two-way-loss {
        type decimal64 {
```

```
                  fraction-digits 6;
                  range "0 .. 50.331642";
                }
                description
                  "Packet loss as a percentage of the total traffic sent
                   over a configurable interval. The finest precision is
                   0.000003%. where the maximum 50.331642%.";
                reference
                  "RFC 7810, section-4.4";
              }
              description
                "Loss constraint on the traffic class.";
            }
            leaf availability-type {
              type identityref {
                base availability-type;
              }
              description
                "Availability Requirement for the TS";
            }
            leaf isolation-type {
              type identityref {
                base isolation-type;
              }
              default "logical-isolation";
              description
                "TS isolation-level.";
            }
            description
              "container for customized policy constraint on the slice
               traffic.";
          }
        }
      }
      list ts-member-group {
        key "ts-member-id";
        description
          "List of included TS Member groups for the SLO.";
        leaf ts-member-id {
          type leafref {
            path "/transport-slices/transport-slice/ts-member/ts-member-id";
          }
```

```
          description
            "Identifies the included list of TS member.";
        }
      }
      container slo-group-monitoring {
        config false;
```



```
        description
          "SLO status Per slo group ";
        uses common-monitoring-parameters;
      }
    }
  }

  grouping slice-template {
    description
      "Grouping for slice-templates.";
    container slice-templates {
      description
        "Container for slice-templates.";
      list slice-template {
        key "id";
        leaf id {
          type string;
          description
            "Identification of the SLO Template to be used.
             Local administration meaning.";
        }
        leaf template-description {
          type string;
          description
            "Description of the SLO template.";
        }
        description
          "List for SLO template identifiers.";
      }
    }
  }

  /* Configuration data nodes */

  container transport-slices {
```

```
    description
      "transport-slice configurations";
    uses slice-template;
    list transport-slice {
      key "ts-id";
      description
        "a transport-slice is identified by a ts-id";
      leaf ts-id {
        type uint32;
        description
          "a unique transport-slice identifier";
      }
      leaf ts-name {
```



```
        type string;
        description
          "ts name";
      }
      leaf-list ts-topology {
        type identityref {
          base ts-topology;
        }
        default "any-to-any";
        description
          "TS topology.";
      }
      uses transport-slice-slo-group;
      uses status-params;
      list ts-endpoint {
        key "ep-id";
        uses endpoint;
        description
          "list of endpoints in this slice";
      }
      list ts-member {
        key "ts-member-id";
        description
          "List of ts-member in a slice";
        uses ts-member;
      }
    }
    //ts-list
```

```
  }
}
```

<CODE ENDS>

9. Security Considerations

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

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

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

o /ietf-transport-slice/transport-slices/transport-slice

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

10. IANA Considerations

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

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

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

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

11. Acknowledgments

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

12. References

12.1. Normative References

[I-D.nsdt-teas-ns-framework]
Gray, E. and J. Drake, "Framework for Transport Network Slices", draft-nsdt-teas-ns-framework-03 (work in progress), April 2020.

[I-D.nsdt-teas-transport-slice-definition]
Rokui, R., Homma, S., Makhijani, K., and L. Contreras, "IETF Definition of Transport Slice", draft-nsdt-teas-transport-slice-definition-02 (work in progress), April 2020.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997,

<https://www.rfc-editor.org/info/rfc2119>.

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

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <https://www.rfc-editor.org/info/rfc6242>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

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

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

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

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

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

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture

(NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/rfc8342>.

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

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

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

12.2. Informative References

[I-D.geng-teas-network-slice-mapping]
Geng, X., Dong, J., Pang, R., Han, L., Niwa, T., Jin, J., Liu, C., and N. Nageshar, "5G End-to-end Network Slice Mapping from the view of Transport Network", draft-geng-teas-network-slice-mapping-01 (work in progress), April 2020.

[I-D.ietf-teas-actn-vn-yang]
Lee, Y., Dhody, D., Ceccarelli, D., Bryskin, I., and B. Yoon, "A Yang Data Model for VN Operation", draft-ietf-teas-actn-vn-yang-08 (work in progress), March 2020.

[I-D.liu-teas-transport-network-slice-yang]
Liu, X., Tantsura, J., Bryskin, I., Contreras, L., WU, Q., Belotti, S., and R. Rokui, "Transport Network Slice YANG Data Model", draft-liu-teas-transport-network-slice-yang-01 (work in progress), July 2020.

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

Appendix A. Comparison with Other Possible Design choices for Transport Slice NBI (Northbound Interface)

According to the TS framework draft 3.3.1. Northbound Inteface (NBI), the TS NBI is a technology-agnostic interface, which is used for a consumer to express requirements for a particular TS. Consumers operate on abstract transport slices, with details related to their realization hidden. As classified by [RFC8309], the TS NBI is classified as Customer Service Model.

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

A.1. ACTN VN Model Augmentation

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

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

- VN: The VN can be seen as a set of edge-to-edge abstract links (a Type 1 VN).

- AP"links" list and "termination points" list describe how nodes in a network are connected to each other

- VN-AP:vertical layering relationships between transport slice networks and underlay networks

- VN-member: Each abstract link is referred to as a VN member and is formed as an E2E tunnel across the underlying networks

The "VN","VN-AP", and "VN-member" can describe basic consumer connection requirements. However, the TS SLO and TS-Endpoint are not clearly defined and there's no direct equivalent. For example, the SLO requirement of the VN is defined through the IETF TE Topologies YANG model, but the TE Topologies model is related to a specific implementation technology. Also, VN-AP does not define "ts-filter-criteria" to specify a specific TSE belonging to a TS.

A.2. RFC8345 Augmentation Model

The difference between the TS NBI requirements and the IETF basic network model is that the TS NBI requests abstract consumer transport slices, with details related to the Transport Network hidden. But the IETF network model is used to describe the interconnection details of a Transport Network. The customer service model does not need to provide details on the Transport Network.

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

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

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

Appendix B. Appendix B Transport Slice Filter Criteria

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

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

```
       +-------------------------------------------------------+
       |            Higher Layer  System                       |
       +-------------------------------------------------------+
         |                       |                         |
         |             Transport Slice Model               |
    +----------+                 |                    +-----------+
    |          |                 |                    |           |
    |RAN Slice  |         +----------------+          |Core Slice |
    |controller |         | TS  controller |          | controller|
    +----+-----+          +-------+--------+          +-----+-----+
         |                       |                         |
         |                       |                         |
     +---+--+       +------------+----------------+       ++-----+
     |      |       |                             |       |      |
     |      |       |                             |       |      |
     |+----+|       |                             |       |      |
     ||    ||TS1-EP1|         TS1                 |       |+----+|
     ||gNB1|+-------+-----+-----------------------+-------+|UPF1||
     ||    |+**********  /                        |       |+----+|
     |+----+|TS2-EP1|  */                         |TS1-EP3|      |
     |              |  /*                         |       |      |
     |+----+|TS1-EP2| /  *                        |       |      |
     ||    |+--------     *    TS2                |       |+----+|
     ||gNB2|+*********************************************+|UPF2||
     ||    ||TS2-EP2|                             |       |+----+|
     |+----+|       |                             |TS2-EP3|      |
     |      |       |                             |       |      |
     |      |       |                             |       |      |
     +------+       +-----------------------------+       +------+
```

As shown in the figure, gNodeB 1 and gNodeB 2 use IP gNB1 and IP gNB2 to communicate with the transport network, respectively. In addition, the traffic of TS1 and TS2 on gNodeB 1 and gNodeB 2 is transmitted through the same access links to the transport network. The transport network need to to distinguish different Transport Slice traffic of same gNB. Therefore, in addition to using "node-id"

and "port-id" to identify a TS-EP, other information is needed along with these parameters to uniquely distinguish a TS-EPs. For example, VLAN IDs in the user traffic can be used to distinguish the TS-EP1 or TS2-EP1 or other TS-EPs of gNBs and UPFs.

Authors' Addresses

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

Email: lana.wubo@huawei.com

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

Email: dhruv.ietf@gmail.com

Liuyan Han
China Mobile

Email: hanliuyan@chinamobile.com

Reza Rokui
Nokia Canada

Email: reza.rokui@nokia.com