

TEAS Working Group  
Internet Draft  
Intended status: Standard Track

Italo Busi  
Haomian Zheng  
Huawei  
Aihua Guo  
Futurewei  
Xufeng Liu  
Volta Networks

Expires: January 2021

July 13, 2020

A YANG Data Model for MPLS-TE Topology  
draft-busizheng-teas-yang-te-mpls-topology-00

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

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

(<http://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.

## Abstract

This document describes a YANG data model for Multi-Protocol Label Switching (MPLS) with Traffic Engineering (MPLS-TE) networks.

## Table of Contents

1. Introduction.....	2
1.1. Tree Diagram.....	3
1.2. Prefixes in Data Node Names.....	3
2. MPLS-TE Types Overview.....	3
3. MPLS-TE Topology Model Overview.....	4
3.1. TE Label Augmentations.....	6
3.2. MPLS-TP Topology.....	6
4. YANG model for common MPLS-TE Types.....	7
5. YANG model for MPLS-TE Topology.....	12
5.1. YANG Tree.....	12
5.2. YANG Code.....	16
6. Security Considerations.....	27
7. IANA Considerations.....	27
8. References.....	27
8.1. Normative References.....	27
8.2. Informative References.....	28
Acknowledgments.....	28
Authors' Addresses.....	29

## 1. Introduction

This document describes a YANG data model for Multi-Protocol Label Switching (MPLS) with Traffic Engineering (MPLS-TE) networks.

This document also defines a collection of common data types and groupings in YANG data modeling language for MPLS-TE networks. These derived common types and groupings are intended to be imported by the MPLS-TE topology model, defined in this document, as well as by the MPLS-TE tunnel model, defined in [TE-MPLS].

Multi-Protocol Label Switching - Transport Profile (MPLS-TP) is a profile of the MPLS protocol that is used in packet switched transport networks and operated in a similar manner to other existing transport technologies (e.g., OTN), as described in RFC5921. The YANG model defined in this document can also be for MPLS-TP networks.

### 1.1. Tree Diagram

A simplified graphical representation of the data model is used in section 5.1 of this this document. The meaning of the symbols in these diagrams is defined in [RFC8340].

### 1.2. Prefixes in Data Node Names

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

Prefix	YANG module	Reference
rt-types	ietf-routing-types	[RFC8294]
tet	ietf-te-topology	[TE-Topology]
tet-pkt	ietf-te-topology-packet	[L3-TE-Topology]
mte-types	ietf-mpls-te-types	This document
tet-mpls	ietf-te-mpls-topology	This document

Table 1: Prefixes and corresponding YANG modules

## 2. MPLS-TE Types Overview

The module `ietf-mpls-te-types` contains the following YANG reusable types and groupings:

`bandwidth-profile-type`:

This identity defines various bandwidth profiles specified by IETF and other organizations that may be used to limit bandwidth utilization of MPLS-TE LSPs.

`load-balancing-type`

This identify defines the types of load-balancing algorithms used on bundled MPLS-TE link.

#### te-packet-path-bandwidth

This grouping defines the path bandwidth information and could be used in MPLS-TE topology model for the representation of MPLS-TE LSP bandwidth. All the path and LSP bandwidth related sections in generic module, [RFC8776], need to be augmented with this grouping for the usage of MPLS-TE. This grouping is also applicable to set up the MPLS-TE tunnel.

#### te-packet-link-bandwidth

This grouping defines the link bandwidth information and could be used in MPLS-TE topology model for link bandwidth representation. All the link bandwidth related sections in generic module, [RFC8776], need to be augmented with this grouping for the usage of MPLS-TE.

#### te-mpls-label-hop

This grouping is used for the augmentation of TE label for MPLS-TE path.

### 3. MPLS-TE Topology Model Overview

The MPLS-TE technology specific topology model augments the ietf-te-topology-packet YANG module, defined in [L3-TE-Topology], which in turns augment the generic ietf-te-topology YANG module, defined in [TE-Topology].

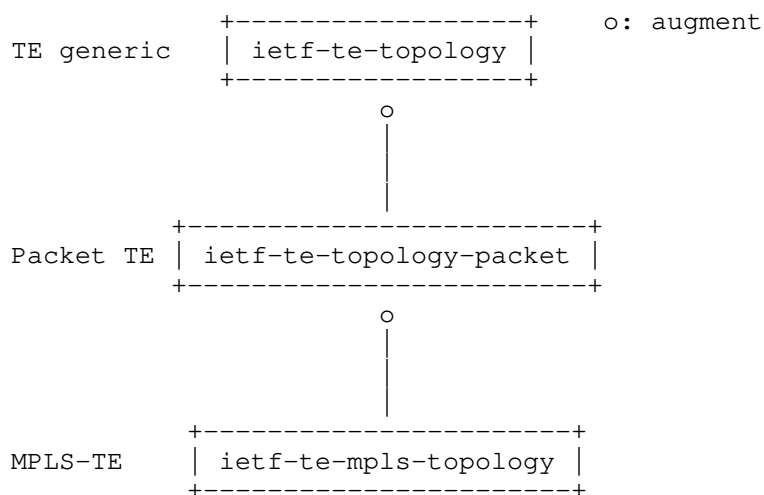


Figure 1 - Relationship between MPLS-TE, Packet-TE and TE topology models

Given the guidance for augmentation in [TE-Topology], the following technology-specific augmentations need to be provided:

- A network-type to indicate that the TE topology is an MPLS-TE Topology, as follow:

```

augment /nw:networks/nw:network/nw:network-types/tet:te-topology
  /tet-pkt:packet:
  +--rw mpls-topology!
  
```

- TE Bandwidth Augmentations for paths and LSPs:

```

+--:(packet)
  +--rw bandwidth-profile-name?  string
  +--rw bandwidth-profile-type?  identityref
  +--rw CIR?                      uint64
  +--rw EIR?                      uint64
  +--rw CBS?                      uint64
  +--rw EBS?                      uint64
  
```

- TE Bandwidth Augmentations for links:

```

+--:(packet)
  +--rw packet-bandwidth?  bandwidth-kbps
  
```

- TE Label Augmentations as described in section 3.2;

### 3.1. TE Label Augmentations

In MPLS-TE, the label allocation is done by NE, information about label values availability is not necessary to be provided to the controller. Moreover, MPLS-TE tunnels are currently established within a single domain.

Therefore this document does not define any MPLS-TE technology-specific augmentations, of the TE Topology model, for the TE label since no TE label related attributes should be instantiated for MPLS-TE Topologies.

Open issue: shall this module allows the setup of MPLS-TE multi-domain tunnels?

### 3.2. MPLS-TP Topology

Multi-Protocol Label Switching - Transport Profile (MPLS-TP) is a profile of the MPLS protocol that is used in packet switched transport networks and operated in a similar manner to other existing transport technologies (e.g., OTN), as described in [RFC5921].

Therefore YANG model defined in this document can also be applicable for MPLS-TP networks.

However, as described in [RFC5921], MPLS-TP networks support bidirectional LSPs and require no ECMP and no PHP. When reporting the topology for an MPLS-TP network, additional information is required to indicate whether the network support these MPLS-TP characteristics.

It is worth noting that [TE-Topology] is already capable to model TE topologies supporting either unidirectional or bidirectional LSPs: all bidirectional TE links can support bidirectional LSPs and all the links can support unidirectional LSPs and it is always possible to associated unidirectional LSPs as long as they belong to the same tunnel.

When setting up bidirectional LSPs (e.g., MPLS-TP LSPs) only bidirectional TE Links are selected by path computation.

In order to allow reporting that ECMP is not affecting forwarding the packets of a given LSP, the load-balancing-type attribute reports

whether a LAG or TE Bundled Link performs load-balancing on a per-flow or per-top-label:

```
augment /nw:networks/nw:network/nt:link/tet:te:
  +--rw load-balancing-type?  mte-types:load-balancing-type
```

When setting up LSPs which do not requires ECMP (e.g., MPLS-TP LSPs) only Links that are not part of a LAG or TE Bundle or that performs per-top-label load balancing are selected by path computation.

It is assumed that almost all the MPLS-TE nodes are capable to support Ultimate Hop Popping (UHP). However, if some interfaces are not able to support UHP, they can report it in the MPLS-TE topology:

```
augment /nw:networks/nw:network/nw:node/nt:termination-point
  /tet:te:
  +--ro uhp-incapable?  empty
```

When setting up LSPs which do not requires PHP (e.g., MPLS-TP LSPs) only the interfaces (LTPs) which are capable to support UHP in the destination node are selected by path computation.

#### 4. YANG model for common MPLS-TE Types

```
<CODE BEGINS>file "ietf-mpls-te-types@2020-07-13.yang"
module ietf-mpls-te-types {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-mpls-te-types";

  prefix "mte-types";

  import ietf-routing-types {
    prefix "rt-types";
  }

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

  organization
    "Internet Engineering Task Force (IETF) TEAS WG";
```

## contact

"WG Web: <<https://datatracker.ietf.org/wg/teas/>>

WG List: <<mailto:teas@ietf.org>>

Editor: Italo Busi

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

Editor: Haomian Zheng

<<mailto:zhenghaomian@huawei.com>>

Editor: Aihua Guo

<<mailto:aihuaguo.ietf@gmail.com>>

Editor: Xufeng Liu

<<mailto:xufeng.liu.ietf@gmail.com>>";

## description

"This module defines technology-specific MPLS-TE types data model.

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 2020-07-13 {

description

"Initial Version";

reference

"draft-busizheng-teas-yang-te-mpls-topology";

}

/\*



```
* Identities (to be moved to te-packet-types?)
*/

identity bandwidth-profile-type {
  description
    "Bandwidth Profile Types";
}

identity mef-10-bwp {
  base bandwidth-profile-type;
  description
    "MEF 10 Bandwidth Profile";
}

identity rfc-2697-bwp {
  base bandwidth-profile-type;
  description
    "RFC 2697 Bandwidth Profile";
}

identity rfc-2698-bwp {
  base bandwidth-profile-type;
  description
    "RFC 2698 Bandwidth Profile";
}

identity rfc-4115-bwp {
  base bandwidth-profile-type;
  description
    "RFC 4115 Bandwidth Profile";
}

/*
* Type Definitions (MPLS-TE)
*/

typedef load-balancing-type {
  type enumeration {
    enum per-flow {
      description
        "The load-balancing algorithm ensures that packets
```

```
        characterized as the same flow (e.g. based on IP 5-tuple)
        that egress on a LAG or a bundled TE link are forwarded
        on the same component link.

        Packets for different flows within the same LSP can be
        forwarded on different component links.";
    }
    enum per-top-label {
        description
            "The load-balancing algorithm ensures incoming MPLS
            packets with the same top MPLS label and that egress on
            a on a LAG or bundled TE link are forwarded on the same
            component link.

            Packets for different flows within the same LSP are
            forwarded on the same component link.";
    }
}
description
    "The type of load balancing used on bundled links.";
} // typedef load-balancing-type

/*
 * Groupings (to be moved to te-packet-types?)
 */

grouping te-packet-path-bandwidth {
    description
        "Path bandwidth for Packet. ";
    leaf bandwidth-profile-name{
        type string;
        description "Name of Bandwidth Profile.";
    }
    leaf bandwidth-profile-type {
        type identityref {
            base bandwidth-profile-type;
        }
        description "Type of Bandwidth Profile.";
    }
}
```

```
leaf CIR {
  type uint64;
  description
    "Committed Information Rate in Kbps";
}

leaf EIR {
  type uint64;
  /*
   * Need to indicate that EIR is not supported by RFC 2697
   */
  must
    '../bw-profile-type = "etht-types:mef-10-bwp" or ' +
    '../bw-profile-type = "etht-types:rfc-2698-bwp" or ' +
    '../bw-profile-type = "etht-types:rfc-4115-bwp"'
  must
    '../bw-profile-type != "etht-types:rfc-2697-bwp"'
  /*
  description
    "Excess Information Rate in Kbps
    In case of RFC 2698, PIR = CIR + EIR";
}

leaf CBS {
  type uint64;
  description
    "Committed Burst Size in in KBytes";
}

leaf EBS {
  type uint64;
  description
    "Excess Burst Size in KBytes.
    In case of RFC 2698, PBS = CBS + EBS";
}
}

grouping te-packet-link-bandwidth {
  description
```

```

        "Link Bandwidth for Packet. ";
    leaf packet-bandwidth {
        type te-packet-types:bandwidth-kbps;
        description
            "Available bandwidth value expressed in kilobits per
            second";
    }
}

/*
 * Groupings (MPLS-TE)
 */

grouping te-mpls-label-hop {
    description
        "MPLS-TE Label Hop.";

    leaf mpls-label {
        type rt-types:mpls-label;
        description
            "MPLS Label.";
    }
} // grouping te-mpls-label-hop
}
<CODE ENDS>

```

Figure 2 - MPLS-TE types YANG module

## 5. YANG model for MPLS-TE Topology

### 5.1. YANG Tree

Figure 3 below shows the tree diagram of the YANG model defined in module `ietf-te-mpls-topology.yang`.

```

module: ietf-te-mpls-topology
  augment /nw:networks/nw:network/nw:node/nt:termination-point/tet:te
    /tet:interface-switching-capability/tet:max-lsp-bandwidth
    /tet:te-bandwidth/tet:technology:
    +--:(packet)
      +--rw bandwidth-profile-name?  string

```

```
    +---rw bandwidth-profile-type?  identityref
    +---rw CIR?                      uint64
    +---rw EIR?                      uint64
    +---rw CBS?                      uint64
    +---rw EBS?                      uint64
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:te-node-attributes/tet:connectivity-matrices
    /tet:path-constraints/tet:te-bandwidth/tet:technology:
+---: (packet)
    +---rw packet-bandwidth?  te-packet-types:bandwidth-kbps
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:te-node-attributes/tet:connectivity-matrices
    /tet:connectivity-matrix/tet:path-constraints
    /tet:te-bandwidth/tet:technology:
+---: (packet)
    +---rw packet-bandwidth?  te-packet-types:bandwidth-kbps
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:information-source-entry/tet:connectivity-matrices
    /tet:path-constraints/tet:te-bandwidth/tet:technology:
+---: (packet)
    +---ro packet-bandwidth?  te-packet-types:bandwidth-kbps
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:information-source-entry/tet:connectivity-matrices
    /tet:connectivity-matrix/tet:path-constraints
    /tet:te-bandwidth/tet:technology:
+---: (packet)
    +---ro packet-bandwidth?  te-packet-types:bandwidth-kbps
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:tunnel-termination-point/tet:client-layer-adaptation
    /tet:switching-capability/tet:te-bandwidth
    /tet:technology:
+---: (packet)
    +---rw packet-bandwidth?  te-packet-types:bandwidth-kbps
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:tunnel-termination-point
    /tet:local-link-connectivities/tet:path-constraints
    /tet:te-bandwidth/tet:technology:
+---: (packet)
    +---rw packet-bandwidth?  te-packet-types:bandwidth-kbps
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:tunnel-termination-point
```

```

        /tet:local-link-connectivities
        /tet:local-link-connectivity/tet:path-constraints
        /tet:te-bandwidth/tet:technology:
+--: (packet)
  +--rw packet-bandwidth?   te-packet-types:bandwidth-kbps
augment /nw:networks/nw:network/nt:link/tet:te
  /tet:te-link-attributes
  /tet:interface-switching-capability/tet:max-lsp-bandwidth
  /tet:te-bandwidth/tet:technology:
+--: (packet)
  +--rw bandwidth-profile-name?   string
  +--rw bandwidth-profile-type?   identityref
  +--rw CIR?                       uint64
  +--rw EIR?                       uint64
  +--rw CBS?                       uint64
  +--rw EBS?                       uint64
augment /nw:networks/nw:network/nt:link/tet:te
  /tet:te-link-attributes/tet:max-link-bandwidth
  /tet:te-bandwidth/tet:technology:
+--: (packet)
  +--rw packet-bandwidth?   te-packet-types:bandwidth-kbps
augment /nw:networks/nw:network/nt:link/tet:te
  /tet:te-link-attributes/tet:max-resv-link-bandwidth
  /tet:te-bandwidth/tet:technology:
+--: (packet)
  +--rw packet-bandwidth?   te-packet-types:bandwidth-kbps
augment /nw:networks/nw:network/nt:link/tet:te
  /tet:te-link-attributes/tet:unreserved-bandwidth
  /tet:te-bandwidth/tet:technology:
+--: (packet)
  +--rw packet-bandwidth?   te-packet-types:bandwidth-kbps
augment /nw:networks/nw:network/nt:link/tet:te
  /tet:information-source-entry
  /tet:interface-switching-capability/tet:max-lsp-bandwidth
  /tet:te-bandwidth/tet:technology:
+--: (packet)
  +--ro bandwidth-profile-name?   string
  +--ro bandwidth-profile-type?   identityref
  +--ro CIR?                       uint64
  +--ro EIR?                       uint64
  +--ro CBS?                       uint64

```

```

    +---ro EBS?                               uint64
augment /nw:networks/nw:network/nt:link/tet:te
    /tet:information-source-entry/tet:max-link-bandwidth
    /tet:te-bandwidth/tet:technology:
+---: (packet)
    +---ro packet-bandwidth? te-packet-types:bandwidth-kbps
augment /nw:networks/nw:network/nt:link/tet:te
    /tet:information-source-entry/tet:max-resv-link-bandwidth
    /tet:te-bandwidth/tet:technology:
+---: (packet)
    +---ro packet-bandwidth? te-packet-types:bandwidth-kbps
augment /nw:networks/nw:network/nt:link/tet:te
    /tet:information-source-entry/tet:unreserved-bandwidth
    /tet:te-bandwidth/tet:technology:
+---: (packet)
    +---ro packet-bandwidth? te-packet-types:bandwidth-kbps
augment /nw:networks/tet:te/tet:templates/tet:link-template
    /tet:te-link-attributes
    /tet:interface-switching-capability/tet:max-lsp-bandwidth
    /tet:te-bandwidth/tet:technology:
+---: (packet)
    +---rw bandwidth-profile-name? string
    +---rw bandwidth-profile-type? identityref
    +---rw CIR?                          uint64
    +---rw EIR?                            uint64
    +---rw CBS?                            uint64
    +---rw EBS?                            uint64
augment /nw:networks/tet:te/tet:templates/tet:link-template
    /tet:te-link-attributes/tet:max-link-bandwidth
    /tet:te-bandwidth/tet:technology:
+---: (packet)
    +---rw packet-bandwidth? te-packet-types:bandwidth-kbps
augment /nw:networks/tet:te/tet:templates/tet:link-template
    /tet:te-link-attributes/tet:max-resv-link-bandwidth
    /tet:te-bandwidth/tet:technology:
+---: (packet)
    +---rw packet-bandwidth? te-packet-types:bandwidth-kbps
augment /nw:networks/tet:te/tet:templates/tet:link-template
    /tet:te-link-attributes/tet:unreserved-bandwidth
    /tet:te-bandwidth/tet:technology:
+---: (packet)

```

```

    +--rw packet-bandwidth?   te-packet-types:bandwidth-kbps
augment /nw:networks/nw:network/nw:network-types/tet:te-topology
    /tet-pkt:packet:
    +--rw mpls-topology!
augment /nw:networks/nw:network/nt:link/tet:te:
    +--rw load-balancing-type? mte-types:load-balancing-type
augment /nw:networks/nw:network/nw:node/nt:termination-point
    /tet:te:
    +--ro uhp-incapable?     empty

```

Figure 3 - MPLS-TE topology YANG tree

## 5.2. YANG Code

```

<CODE BEGINS>file "ietf-te-mpls-topology@2020-07-13.yang"
module iETF-te-mpls-topology {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-te-mpls-topology";

  prefix "tet-mpls";

  import iETF-network {
    prefix "nw";
  }

  import iETF-network-topology {
    prefix "nt";
  }

  import iETF-te-topology {
    prefix "tet";
  }

  import iETF-te-topology-packet {
    prefix "tet-pkt";
  }

  import iETF-mpls-te-types {
    prefix "mte-types";
  }
}

```



organization

"Internet Engineering Task Force (IETF) TEAS WG";

contact

"WG Web: <<https://datatracker.ietf.org/wg/teas/>>

WG List: <<mailto:teas@ietf.org>>

Editor: Italo Busi

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

Editor: Haomian Zheng

<<mailto:zhenghaomian@huawei.com>>

Editor: Aihua Guo

<<mailto:aihuaguo.ietf@gmail.com>>

Editor: Xufeng Liu

<<mailto:xufeng.liu.ietf@gmail.com>>"

description

"This module defines technology-specific MPLS-TE topology data model.

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 2020-07-13 {

description

"Initial Version";

reference

"draft-busizheng-teas-yang-te-mpls-topology";

}

```
/*
 * Augment TE bandwidth (to be moved to te-topology-packet?)
 */

augment "/nw:networks/nw:network/nw:node/nt:termination-point/"
  + "tet:te/"
  + "tet:interface-switching-capability/tet:max-lsp-bandwidth/"
  + "tet:te-bandwidth/tet:technology" {
when "../..../..../..../nw:network-types/tet:te-topology/"
  + "tet-pkt:packet" {
  description
    "Augmentation parameters apply only for networks with
    Packet TE topology type.";
}
description
  "Augment maximum LSP TE bandwidth for the link termination
  point (LTP).";
case packet {
  uses mte-types:te-packet-path-bandwidth;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:te-node-attributes/tet:connectivity-matrices/"
  + "tet:path-constraints/tet:te-bandwidth/tet:technology" {
when "../..../..../..../nw:network-types/tet:te-topology/"
  + "tet-pkt:packet" {
  description
    "Augmentation parameters apply only for networks with
    Packet TE topology type.";
}
description
  "Augment TE bandwidth path constraints of the TE node
  connectivity matrices.";
case packet {
  uses mte-types:te-packet-link-bandwidth;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
```

```

    + "tet:te-node-attributes/tet:connectivity-matrices/"
    + "tet:connectivity-matrix/"
    + "tet:path-constraints/tet:te-bandwidth/tet:technology" {
when "../..../..../..../..../nw:network-types/tet:te-topology/"
    + "tet-pkt:packet" {
    description
        "Augmentation parameters apply only for networks with
        Packet TE topology type.";
    }
description
    "Augment TE bandwidth path constraints of the
    connectivity matrix entry.";
case packet {
    uses mte-types:te-packet-link-bandwidth;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:information-source-entry/tet:connectivity-matrices/"
    + "tet:path-constraints/tet:te-bandwidth/tet:technology" {
when "../..../..../..../nw:network-types/tet:te-topology/"
    + "tet-pkt:packet" {
    description
        "Augmentation parameters apply only for networks with
        Packet TE topology type.";
    }
description
    "Augment TE bandwidth path constraints of the TE node
    connectivity matrices information source.";
case packet {
    uses mte-types:te-packet-link-bandwidth;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:information-source-entry/tet:connectivity-matrices/"
    + "tet:connectivity-matrix/"
    + "tet:path-constraints/tet:te-bandwidth/tet:technology" {
when "../..../..../..../nw:network-types/tet:te-topology/"
    + "tet-pkt:packet" {
    description

```

```
        "Augmentation parameters apply only for networks with
          Packet TE topology type.";
    }
    description
      "Augment TE bandwidth path constraints of the
        connectivity matrix entry information source";
    case packet {
      uses mte-types:te-packet-link-bandwidth;
    }
  }

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:tunnel-termination-point/"
  + "tet:client-layer-adaptation/tet:switching-capability/"
  + "tet:te-bandwidth/tet:technology" {
when "../..../..../..../nw:network-types/tet:te-topology/"
  + "tet-pkt:packet" {
  description
    "Augmentation parameters apply only for networks with
      Packet TE topology type.";
  }
  description
    "Augment client TE bandwidth of the tunnel termination point
      (TTP)";
  case packet {
    uses mte-types:te-packet-link-bandwidth;
  }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:tunnel-termination-point/"
  + "tet:local-link-connectivities/tet:path-constraints/"
  + "tet:te-bandwidth/tet:technology" {
when "../..../..../..../nw:network-types/tet:te-topology/"
  + "tet-pkt:packet" {
  description
    "Augmentation parameters apply only for networks with
      Packet TE topology type.";
  }
  description
    "Augment TE bandwidth path constraints for the TTP
```

```
        Local Link Connectivities.";
    case packet {
        uses mte-types:te-packet-link-bandwidth;
    }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:tunnel-termination-point/"
    + "tet:local-link-connectivities/"
    + "tet:local-link-connectivity/tet:path-constraints/"
    + "tet:te-bandwidth/tet:technology" {
    when "../../../../../../../../../../../nw:network-types/tet:te-topology/"
        + "tet-pkt:packet" {
        description
            "Augmentation parameters apply only for networks with
            Packet TE topology type.";
    }
    description
        "Augment TE bandwidth path constraints for the TTP
        Local Link Connectivity entry.";
    case packet {
        uses mte-types:te-packet-link-bandwidth;
    }
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
    + "tet:te-link-attributes/"
    + "tet:interface-switching-capability/tet:max-lsp-bandwidth/"
    + "tet:te-bandwidth/tet:technology" {
    when "../../../../../../../../../../../nw:network-types/tet:te-topology/"
        + "tet-pkt:packet" {
        description
            "Augmentation parameters apply only for networks with
            Packet TE topology type.";
    }
    description
        "Augment maximum LSP TE bandwidth for the TE link.";
    case packet {
        uses mte-types:te-packet-path-bandwidth;
    }
}
}
```

```
augment "/nw:networks/nw:network/nt:link/tet:te/"
  + "tet:te-link-attributes/"
  + "tet:max-link-bandwidth/"
  + "tet:te-bandwidth/tet:technology" {
when "../../../../../../nw:network-types/tet:te-topology/"
  + "tet-pkt:packet" {
  description
    "Augmentation parameters apply only for networks with
    Packet TE topology type.";
}
description
  "Augment maximum TE bandwidth for the TE link";
case packet {
  uses mte-types:te-packet-link-bandwidth;
}
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
  + "tet:te-link-attributes/"
  + "tet:max-resv-link-bandwidth/"
  + "tet:te-bandwidth/tet:technology" {
when "../../../../../../nw:network-types/tet:te-topology/"
  + "tet-pkt:packet" {
  description
    "Augmentation parameters apply only for networks with
    Packet TE topology type.";
}
description
  "Augment maximum reservable TE bandwidth for the TE link";
case packet {
  uses mte-types:te-packet-link-bandwidth;
}
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
  + "tet:te-link-attributes/"
  + "tet:unreserved-bandwidth/"
  + "tet:te-bandwidth/tet:technology" {
when "../../../../../../nw:network-types/tet:te-topology/"
  + "tet-pkt:packet" {
```

```
    description
      "Augmentation parameters apply only for networks with
        Packet TE topology type.";
  }
  description
    "Augment unreserved TE bandwidth for the TE Link";
  case packet {
    uses mte-types:te-packet-link-bandwidth;
  }
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
  + "tet:information-source-entry/"
  + "tet:interface-switching-capability/"
  + "tet:max-lsp-bandwidth/"
  + "tet:te-bandwidth/tet:technology" {
  when "../../../nw:network-types/tet:te-topology/"
    + "tet-pkt:packet" {
    description
      "Augmentation parameters apply only for networks with
        Packet TE topology type.";
  }
  description
    "Augment maximum LSP TE bandwidth for the TE link
      information source";
  case packet {
    uses mte-types:te-packet-path-bandwidth;
  }
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
  + "tet:information-source-entry/"
  + "tet:max-link-bandwidth/"
  + "tet:te-bandwidth/tet:technology" {
  when "../../../nw:network-types/tet:te-topology/"
    + "tet-pkt:packet" {
    description
      "Augmentation parameters apply only for networks with
        Packet TE topology type.";
  }
  description
```

```
    "Augment maximum TE bandwidth for the TE link
    information source";
  case packet {
    uses mte-types:te-packet-link-bandwidth;
  }
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
  + "tet:information-source-entry/"
  + "tet:max-resv-link-bandwidth/"
  + "tet:te-bandwidth/tet:technology" {
  when "../../../../../../../nw:network-types/tet:te-topology/"
  + "tet-pkt:packet" {
    description
      "Augmentation parameters apply only for networks with
      Packet TE topology type.";
  }
  description
    "Augment maximum reservable TE bandwidth for the TE link
    information source";
  case packet {
    uses mte-types:te-packet-link-bandwidth;
  }
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
  + "tet:information-source-entry/"
  + "tet:unreserved-bandwidth/"
  + "tet:te-bandwidth/tet:technology" {
  when "../../../../../../../nw:network-types/tet:te-topology/"
  + "tet-pkt:packet" {
    description
      "Augmentation parameters apply only for networks with
      Packet TE topology type.";
  }
  description
    "Augment unreserved TE bandwidth of the TE link
    information source";
  case packet {
    uses mte-types:te-packet-link-bandwidth;
  }
}
```



```
}

augment "/nw:networks/tet:te/tet:templates/"
  + "tet:link-template/tet:te-link-attributes/"
  + "tet:interface-switching-capability/"
  + "tet:max-lsp-bandwidth/"
  + "tet:te-bandwidth/tet:technology" {
  description
    "Augment maximum LSP TE bandwidth of the TE link
    template";
  case packet {
    uses mte-types:te-packet-path-bandwidth;
  }
}

augment "/nw:networks/tet:te/tet:templates/"
  + "tet:link-template/tet:te-link-attributes/"
  + "tet:max-link-bandwidth/"
  + "tet:te-bandwidth/tet:technology" {
  description
    "Augment maximum TE bandwidth the TE link template";
  case packet {
    uses mte-types:te-packet-link-bandwidth;
  }
}

augment "/nw:networks/tet:te/tet:templates/"
  + "tet:link-template/tet:te-link-attributes/"
  + "tet:max-resv-link-bandwidth/"
  + "tet:te-bandwidth/tet:technology" {
  description
    "Augment maximum reservable TE bandwidth for the TE link
    template.";
  case packet {
    uses mte-types:te-packet-link-bandwidth;
  }
}

augment "/nw:networks/tet:te/tet:templates/"
  + "tet:link-template/tet:te-link-attributes/"
  + "tet:unreserved-bandwidth/"
```

```
    + "tet:te-bandwidth/tet:technology" {
description
  "Augment unreserved TE bandwidth the TE link template";
case packet {
  uses mte-types:te-packet-link-bandwidth;
}
}

/*
 * Augmentations
 */

augment "/nw:networks/nw:network/nw:network-types/"
  + "tet:te-topology/tet-pkt:packet" {
description
  "Augment network types to include MPLS-TE Topology Type";
container mpls-topology {
  presence
    "Indicates an MPLS-TE Topology Type.";
description
  "Its presence indicates an MPLS-TE Topology";
}
}

augment "/nw:networks/nw:network/nt:link/tet:te" {
  when "../nw:network-types/tet:te-topology/"
  + "tet-pkt:packet/tet-mpls:mpls-topology" {
description
  "Augment MPLS-TE Topology.";
}
description
  "Augment TE Link.";

leaf load-balancing-type {
  type mte-types:load-balancing-type;
  default 'per-flow';
description
  "Indicates the type of load-balancing (per-flow or per-LSP)
  performed by the bundled TE Link.

  This leaf is not present when the TE Link is not bundled.";
}
```

```
    } // leaf load-balancing-type
  }

  augment "/nw:networks/nw:network/nw:node/nt:termination-point/"
    + "tet:te" {
    when "../.../nw:network-types/tet:te-topology/"
      + "tet-pkt:packet/tet-mpls:mpls-topology" {
      description "Augment MPLS-TE Topology.";
    }
    description "Augment LTP.";

    leaf uhp-incapable {
      type empty;
      config false;
      description
        "When present, indicates that the LTP is not capable to
        support Ultimate Hop Popping (UHP).";
    } // leaf uhp-incapable
  }
}
<CODE ENDS>
```

Figure 4 - MPLS-TE topology YANG module

## 6. Security Considerations

To be added

## 7. IANA Considerations

To be added

## 8. References

### 8.1. Normative References

[RFC6991] J. Schoenwaelder, "Common YANG Data Types", RFC6991.

[RFC8294] X. Liu, et. al., "Common YANG Data Types for the Routing Area", RFC8294.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, March 2018.

[RFC8776] Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Common YANG Data Types for Traffic Engineering", RFC8776, June 2020.

[TE-Topology] X. Liu, et. al., "YANG Data Model for TE Topologies", draft-ietf-teas-yang-te-topo, work in progress.

[L3-TE-Topology] X. Liu, et. al., "YANG Data Model for Layer 3 TE Topologies", draft-ietf-teas-yang-l3-te-topo, work in progress.

## 8.2. Informative References

[RFC5921] M. Bocci, et., al., "A Framework for MPLS in Transport Networks", RFC5921.

[TE-MPLS] T. Saad, et. al., "A YANG Data Model for MPLS Traffic Engineering Tunnels", draft-ietf-teas-yang-te-mpls, work in progress.

## Acknowledgments

The authors would also like to thank Tarek Saad, Vishnu Pavan Beeram, Rakesh Gandhi, Xufeng Liu, Igor Bryskin for their input on how to support MPLS-TP features (bidirectional LSPs, no ECMP, no PHP) using a common MPLS-TE topology model.

We thank Loa Andersson and Igor Bryskin for providing useful suggestions for this draft.

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

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

Haomian Zheng  
Huawei Technologies  
Email: zhenghaomian@huawei.com

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

Xufeng Liu  
Volta Networks  
Email: xufeng.liu.ietf@gmail.com



TEAS Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 14, 2021

Y. Lee, Ed.  
Samsung Electronics  
D. Dhody, Ed.  
S. Karunanithi  
Huawei Technologies  
R. Vilalta  
CTC  
D. King  
Lancaster University  
D. Ceccarelli  
Ericsson  
July 13, 2020

YANG models for VN/TE Performance Monitoring Telemetry and Scaling  
Intent Autonomics  
draft-ietf-teas-actn-pm-telemetry-autonomics-03

#### Abstract

This document provides YANG data models that describe performance monitoring telemetry and scaling intent mechanism for TE-tunnels and Virtual Networks (VN).

The models presented in this draft allow customers to subscribe to and monitor their key performance data of their interest on the level of TE-tunnel or VN. The models also provide customers with the ability to program autonomic scaling intent mechanism on the level of TE-tunnel as well as VN.

#### 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 14, 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 . . . . .	2
1.1. Terminology . . . . .	4
1.1.1. Requirements Language . . . . .	4
1.2. Tree diagram . . . . .	4
1.3. Prefixes in Data Node Names . . . . .	5
2. Use-Cases . . . . .	5
3. Design of the Data Models . . . . .	7
3.1. TE KPI Telemetry Model . . . . .	7
3.2. VN KPI Telemetry Model . . . . .	8
4. Autonomic Scaling Intent Mechanism . . . . .	9
5. Notification . . . . .	11
5.1. YANG Push Subscription Examples . . . . .	11
6. YANG Data Tree . . . . .	12
7. YANG Data Model . . . . .	15
7.1. ietf-te-kpi-telemetry model . . . . .	15
7.2. ietf-vn-kpi-telemetry model . . . . .	21
8. Security Considerations . . . . .	25
9. IANA Considerations . . . . .	25
10. Acknowledgements . . . . .	26
11. References . . . . .	26
11.1. Normative References . . . . .	26
11.2. Informative References . . . . .	28
Authors' Addresses . . . . .	29

## 1. Introduction

The YANG [RFC7950] model discussed in [I-D.ietf-teas-actn-vn-yang] is used to operate customer-driven Virtual Networks (VNs) during the VN instantiation, VN computation, and its life-cycle service management and operations. YANG model discussed in [I-D.ietf-teas-yang-te] is



used to operate TE-tunnels during the tunnel instantiation, and its life-cycle management and operations.

The models presented in this draft allow the applications hosted by the customers to subscribe to and monitor their key performance data of their interest on the level of VN [I-D.ietf-teas-actn-vn-yang] or TE-tunnel [I-D.ietf-teas-yang-te]. The key characteristic of the models presented in this document is a top-down programmability that allows the applications hosted by the customers to subscribe to and monitor key performance data of their interest and autonomic scaling intent mechanism on the level of VN as well as TE-tunnel.

According to the classification of [RFC8309], the YANG data models presented in this document can be classified as customer service models, which is mapped to CMI (Customer Network Controller (CNC)-Multi-Domain Service Coordinator (MSDC) interface) of ACTN [RFC8453].

[RFC8233] describes key network performance data to be considered for end-to-end path computation in TE networks. Key performance indicator (KPI) is a term that describes critical performance data that may affect VN/TE-tunnel service. The services provided can be optimized to meet the requirements (such as traffic patterns, quality, and reliability) of the applications hosted by the customers.

This document provides YANG data models generically applicable to any VN/TE-Tunnel service clients to provide an ability to program their customized performance monitoring subscription and publication data models and automatic scaling in/out intent data models. These models can be utilized by a client network controller to initiate these capability to a transport network controller communicating with the client controller via a NETCONF [RFC8341] or a RESTCONF [RFC8040] interface.

The term performance monitoring being used in this document is different from the term that has been used in transport networks for many years. Performance monitoring in this document refers to subscription and publication of streaming telemetry data. Subscription is initiated by the client (e.g., CNC) while publication is provided by the network (e.g., MDSC/PNC) based on the client's subscription. As the scope of performance monitoring in this document is telemetry data on the level of client's VN or TE-tunnel, the entity interfacing the client (e.g., MDSC) has to provide VN or TE-tunnel level information. This would require controller capability to derive VN or TE-tunnel level performance data based on lower-level data collected via PM counters in the Network Elements (NE). How the controller entity derives such customized level data (i.e., VN or TE-tunnel level) is out of the scope of this document.

The data model includes configuration and state data according to the new Network Management Datastore Architecture [RFC8342].

### 1.1. Terminology

Refer to [RFC8453], [RFC7926], and [RFC8309] for the key terms used in this document.

**Key Performance Data:** This refers to a set of data the customer is interested in monitoring for their instantiated VNs or TE-tunnels. Key performance data and key performance indicators are interchangeable in this draft.

**Scaling:** This refers to the network ability to re-shape its own resources. Scale out refers to improve network performance by increasing the allocated resources, while scale in refers to decrease the allocated resources, typically because the existing resources are unnecessary.

**Scaling Intent:** To declare scaling conditions, scaling intent is used. Specifically, scaling intent refers to the intent expressed by the client that allows the client to program/configure conditions of their key performance data either for scaling out or scaling in. Various conditions can be set for scaling intent on either VN or TE-tunnel level.

**Network Autonomics:** This refers to the network automation capability that allows client to initiate scaling intent mechanisms and provides the client with the status of the adjusted network resources based on the client's scaling intent in an automated fashion.

#### 1.1.1. Requirements Language

The key words "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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

### 1.2. Tree diagram

A simplified graphical representation of the data model is used in Section 5 of this this document. The meaning of the symbols in these diagrams is defined in [RFC8340].

### 1.3. Prefixes in Data Node Names

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

Prefix	YANG module	Reference
inet	ietf-inet-types	[RFC6991]
te	ietf-te	[I-D.ietf-teas-yang-te]
te-types	ietf-te-types	[RFC8776]
te-tel	ietf-te-kpi-telemetry	[RFCXXXX]
vn	ietf-vn	[I-D.ietf-teas-actn-vn-yang]
vn-tel	ietf-vn-kpi-telemetry	[RFCXXXX]

Table 1: Prefixes and corresponding YANG modules

Note: The RFC Editor will replace XXXX with the number assigned to the RFC once this draft becomes an RFC.

Further, the following additional documents are referenced in the model defined in this document -

- o [RFC7471] - OSPF Traffic Engineering (TE) Metric Extensions.
- o [RFC8570] - IS-IS Traffic Engineering (TE) Metric Extensions.
- o [RFC7823] - Performance-Based Path Selection for Explicitly Routed Label Switched Paths (LSPs) Using TE Metric Extensions.

## 2. Use-Cases

[I-D.xu-actn-perf-dynamic-service-control] describes use-cases relevant to this draft. It introduces the dynamic creation, modification and optimization of services based on the performance monitoring. Figure 1 shows a high-level workflows for dynamic service control based on traffic monitoring.

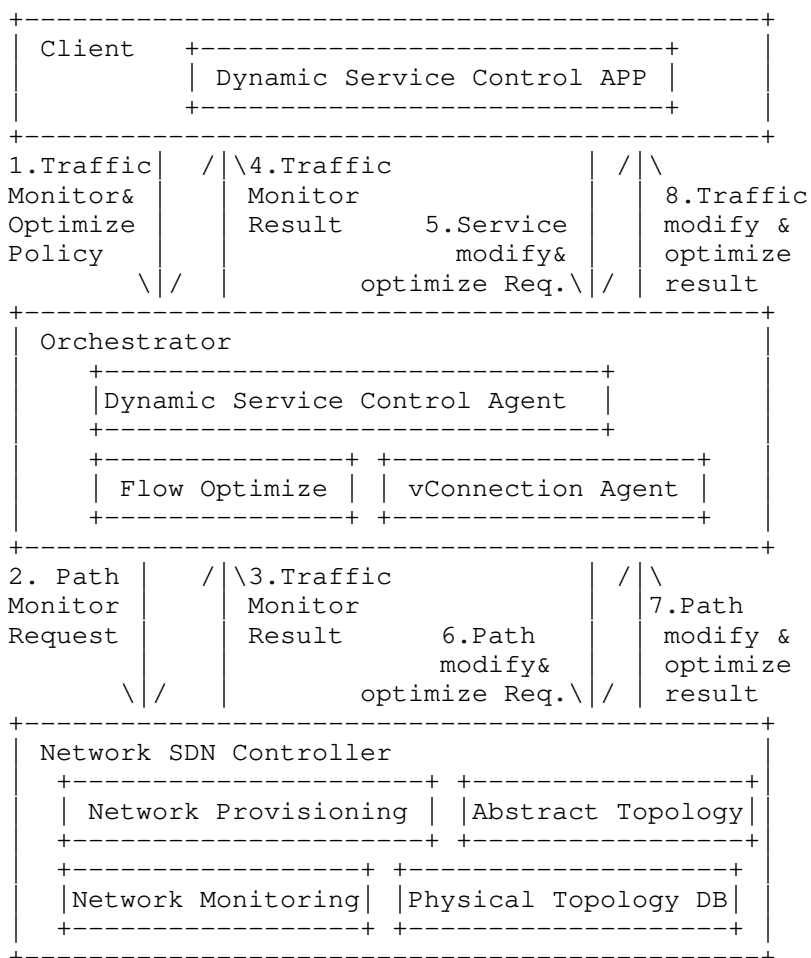


Figure 1: Workflows for dynamic service control based on traffic monitoring

Some of the key points from [I-D.xu-actn-perf-dynamic-service-control] are as follows:

- o Network traffic monitoring is important to facilitate automatic discovery of the imbalance of network traffic, and initiate the network optimization, thus helping the network operator or the virtual network service provider to use the network more efficiently and save the Capital Expense (CAPEX) and the Operating Expense (OPEX).

- o Customer services have various Service Level Agreement (SLA) requirements, such as service availability, latency, latency jitter, packet loss rate, Bit Error Rate (BER), etc. The transport network can satisfy service availability and BER requirements by providing different protection and restoration mechanisms. However, for other performance parameters, there are no such mechanisms. In order to provide high quality services according to customer SLA, one possible solution is to measure the SLA related performance parameters, and dynamically provision and optimize services based on the performance monitoring results.
- o Performance monitoring in a large scale network could generate a huge amount of performance information. Therefore, the appropriate way to deliver the information in the client and network interfaces should be carefully considered.

### 3. Design of the Data Models

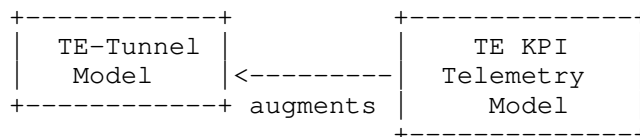
The YANG models developed in this document describe two models:

- (i) TE KPI Telemetry Model which provides the TE-Tunnel level of performance monitoring mechanism and scaling intent mechanism that allows scale in/out programming by the customer. (See Section 3.1 & Section 7.1 for details).
- (ii) VN KPI Telemetry Model which provides the VN level of the aggregated performance monitoring mechanism and scaling intent mechanism that allows scale in/out programming by the customer (See Section 3.2 & Section 7.2 for details).

#### 3.1. TE KPI Telemetry Model

This module describes performance telemetry for TE-tunnel model. The telemetry data is augmented to tunnel state. This module also allows autonomic traffic engineering scaling intent configuration mechanism on the TE-tunnel level. Various conditions can be set for auto-scaling based on the telemetry data (See Section 5 for details)

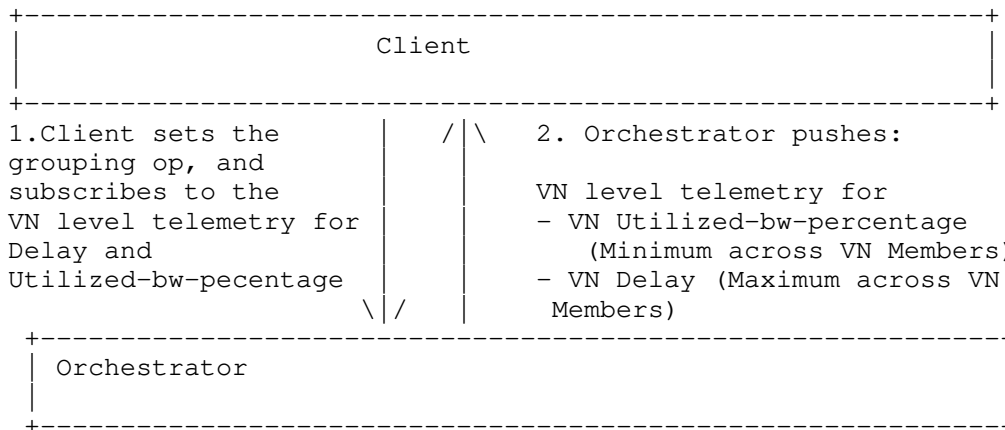
The TE KPI Telemetry Model augments the TE-Tunnel Model to enhance TE performance monitoring capability. This monitoring capability will facilitate proactive re-optimization and reconfiguration of TEs based on the performance monitoring data collected via the TE KPI Telemetry YANG model.



3.2. VN KPI Telemetry Model

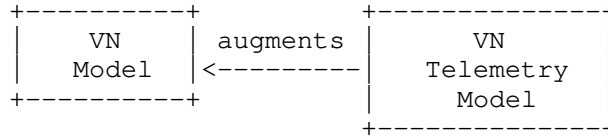
This module describes performance telemetry for VN model. The telemetry data is augmented both at the VN Level as well as individual VN member level. This module also allows autonomic traffic engineering scaling intent configuration mechanism on the VN level. Scale in/out criteria might be used for network autonomies in order the controller to react to a certain set of variations in monitored parameters (See Section 4 for illustrations).

Moreover, this module also provides mechanism to define aggregated telemetry parameters as a grouping of underlying VN level telemetry parameters. Grouping operation (such as maximum, mean) could be set at the time of configuration. For example, if maximum grouping operation is used for delay at the VN level, the VN telemetry data is reported as the maximum {delay\_vn\_member\_1, delay\_vn\_member\_2,.. delay\_vn\_member\_N}. Thus, this telemetry abstraction mechanism allows the grouping of a certain common set of telemetry values under a grouping operation. This can be done at the VN-member level to suggest how the E2E telemetry be inferred from the per domain tunnel created and monitored by PNCs. One proposed example is the following:



The VN Telemetry Model augments the basic VN model to enhance VN monitoring capability. This monitoring capability will facilitate proactive re-optimization and reconfiguration of VNs based on the

performance monitoring data collected via the VN Telemetry YANG model.



#### 4. Autonomic Scaling Intent Mechanism

Scaling intent configuration mechanism allows the client to configure automatic scale-in and scale-out mechanisms on both the TE-tunnel and the VN level. Various conditions can be set for auto-scaling based on the PM telemetry data.

There are a number of parameters involved in the mechanism:

- o scale-out-intent or scale-in-intent: whether to scale-out or scale-in.
- o performance-type: performance metric type (e.g., one-way-delay, one-way-delay-min, one-way-delay-max, two-way-delay, two-way-delay-min, two-way-delay-max, utilized bandwidth, etc.)
- o threshold-value: the threshold value for a certain performance-type that triggers scale-in or scale-out.
- o scaling-operation-type: in case where scaling condition can be set with one or more performance types, then scaling-operation-type (AND, OR, MIN, MAX, etc.) is applied to these selected performance types and its threshold values.
- o Threshold-time: the duration for which the criteria MUST hold true.
- o Cooldown-time: the duration after a scaling action has been triggered, for which there will be no further operation.

The following tree is a part of ietf-te-kpi-telemetry tree whose model is presented in full detail in Sections 6 & 7.

```

module: ietf-te-kpi-telemetry
augment /te:te/te:tunnels/te:tunnel:
  +--rw te-scaling-intent
  |   +--rw scale-in-intent
  |   |   +--rw threshold-time?      uint32
  |   |   +--rw cooldown-time?      uint32
  |   |   +--rw scaling-condition* [performance-type]
  |   |   |   +--rw performance-type      identityref
  |   |   |   +--rw threshold-value?     string
  |   |   |   +--rw scale-in-operation-type?
  |   |   |       scaling-criteria-operation
  |   +--rw scale-out-intent
  |   |   +--rw threshold-time?      uint32
  |   |   +--rw cooldown-time?      uint32
  |   |   +--rw scaling-condition* [performance-type]
  |   |   |   +--rw performance-type      identityref
  |   |   |   +--rw threshold-value?     string
  |   |   |   +--rw scale-out-operation-type?
  |   |   |       scaling-criteria-operation

```

Let say the client wants to set the scaling out operation based on two performance-types (e.g., two-way-delay and utilized-bandwidth for a te-tunnel), it can be done as follows:

- o Set Threshold-time: x (sec) (duration for which the criteria must hold true)
- o Set Cooldown-time: y (sec) (the duration after a scaling action has been triggered, for which there will be no further operation)
- o Set AND for the scale-out-operation-type

In the scaling condition's list, the following two components can be set:

List 1: Scaling Condition for Two-way-delay

- o performance type: Two-way-delay
- o threshold-value: z milli-seconds

List 2: Scaling Condition for Utilized bandwidth

- o performance type: Utilized bandwidth
- o threshold-value: w megabytes



## 5. Notification

This model does not define specific notifications. To enable notifications, the mechanism defined in [RFC8641] and [RFC8640] can be used. This mechanism currently allows the user to:

- o Subscribe to notifications on a per client basis.
- o Specify subtree filters or xpath filters so that only interested contents will be sent.
- o Specify either periodic or on-demand notifications.

### 5.1. YANG Push Subscription Examples

[RFC8641] allows subscriber applications to request a continuous, customized stream of updates from a YANG datastore.

Below example shows the way for a client to subscribe to the telemetry information for a particular tunnel (Tunnell). The telemetry parameter that the client is interested in is one-way-delay.

```
<netconf:rpc netconf:message-id="101"
  xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
  <establish-subscription
    xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push:1.0">
    <filter netconf:type="subtree">
      <te xmlns="urn:ietf:params:xml:ns:yang:ietf-te">
        <tunnels>
          <tunnel>
            <name>Tunnell</name>
            <identifier/>
            <state>
              <te-telemetry xmlns="urn:ietf:params:xml:ns:yang:
                ietf-te-kpi-telemetry">
                <one-way-delay/>
              </te-telemetry>
            </state>
          </tunnel>
        </tunnels>
      </te>
    </filter>
    <period>500</period>
    <encoding>encode-xml</encoding>
  </establish-subscription>
</netconf:rpc>
```

This example shows the way for a client to subscribe to the telemetry information for all VNs. The telemetry parameter that the client is interested in is one-way-delay and one-way-utilized-bandwidth.

```
<netconf:rpc netconf:message-id="101"
  xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
  <establish-subscription
    xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push:1.0">
    <filter netconf:type="subtree">
      <vn-state xmlns="urn:ietf:params:xml:ns:yang:ietf-vn">
        <vn>
          <vn-list>
            <vn-id/>
            <vn-name/>
            <vn-telemetry xmlns="urn:ietf:params:xml:ns:yang:
              ietf-vn-kpi-telemetry">
              <one-way-delay/>
              <one-way-utilized-bandwidth/>
            </vn-telemetry >
          </vn-list>
        </vn>
      </vn-state>
    </filter>
    <period>500</period>
  </establish-subscription>
</netconf:rpc>
```

## 6. YANG Data Tree

```
module: ietf-te-kpi-telemetry
  augment /te:te/te:tunnels/te:tunnel:
    +--rw te-scaling-intent
      +--rw scale-in-intent
        +--rw threshold-time?          uint32
        +--rw cooldown-time?          uint32
        +--rw scaling-condition* [performance-type]
          +--rw performance-type      identityref
          +--rw threshold-value?      string
          +--rw scale-in-operation-type?
            scaling-criteria-operation
        +--rw scale-out-intent
          +--rw threshold-time?        uint32
          +--rw cooldown-time?         uint32
          +--rw scaling-condition* [performance-type]
            +--rw performance-type     identityref
```

```

    |         +---rw threshold-value?          string
    |         +---rw scale-out-operation-type?
    |             scaling-criteria-operation
+---ro te-telemetry
    +---ro id?                                telemetry-id
    +---ro performance-metrics-one-way
    |     +---ro one-way-delay?                uint32
    |     +---ro one-way-delay-normality?
    |         | te-types:performance-metrics-normality
    |     +---ro one-way-residual-bandwidth?
    |         | rt-types:bandwidth-ieee-float32
    |     +---ro one-way-residual-bandwidth-normality?
    |         | te-types:performance-metrics-normality
    |     +---ro one-way-available-bandwidth?
    |         | rt-types:bandwidth-ieee-float32
    |     +---ro one-way-available-bandwidth-normality?
    |         | te-types:performance-metrics-normality
    |     +---ro one-way-utilized-bandwidth?
    |         | rt-types:bandwidth-ieee-float32
    |     +---ro one-way-utilized-bandwidth-normality?
    |         | te-types:performance-metrics-normality
+---ro performance-metrics-two-way
    +---ro two-way-delay?                    uint32
    +---ro two-way-delay-normality?
        | te-types:performance-metrics-normality

```

```
module: ietf-vn-kpi-telemetry
```

```
augment /vn:vn/vn:vn-list:
```

```

+---rw vn-scaling-intent
    |     +---rw scale-in-intent
    |         | +---rw threshold-time?        uint32
    |         | +---rw cooldown-time?        uint32
    |         | +---rw scaling-condition* [performance-type]
    |         |     +---rw performance-type    identityref
    |         |     +---rw threshold-value?   string
    |         |     +---rw scale-in-operation-type?
    |         |         scaling-criteria-operation
    |     +---rw scale-out-intent
    |         | +---rw threshold-time?        uint32
    |         | +---rw cooldown-time?        uint32
    |         | +---rw scaling-condition* [performance-type]
    |         |     +---rw performance-type    identityref
    |         |     +---rw threshold-value?   string
    |         |     +---rw scale-out-operation-type?
    |         |         scaling-criteria-operation

```

```

+--ro vn-telemetry
  +--ro performance-metrics-one-way
    |   +--ro one-way-delay?                               uint32
    |   +--ro one-way-delay-normality?
    |       |   te-types:performance-metrics-normality
    +--ro one-way-residual-bandwidth?
    |       |   rt-types:bandwidth-ieee-float32
    +--ro one-way-residual-bandwidth-normality?
    |       |   te-types:performance-metrics-normality
    +--ro one-way-available-bandwidth?
    |       |   rt-types:bandwidth-ieee-float32
    +--ro one-way-available-bandwidth-normality?
    |       |   te-types:performance-metrics-normality
    +--ro one-way-utilized-bandwidth?
    |       |   rt-types:bandwidth-ieee-float32
    +--ro one-way-utilized-bandwidth-normality?
    |       |   te-types:performance-metrics-normality
  +--ro performance-metrics-two-way
    |   +--ro two-way-delay?                               uint32
    |   +--ro two-way-delay-normality?
    |       |   te-types:performance-metrics-normality
  +--ro grouping-operation?                               grouping-operation
augment /vn:vn/vn:vn-list/vn:vn-member-list:
+--ro vn-member-telemetry
  +--ro performance-metrics-one-way
    |   +--ro one-way-delay?                               uint32
    |   +--ro one-way-delay-normality?
    |       |   te-types:performance-metrics-normality
    +--ro one-way-residual-bandwidth?
    |       |   rt-types:bandwidth-ieee-float32
    +--ro one-way-residual-bandwidth-normality?
    |       |   te-types:performance-metrics-normality
    +--ro one-way-available-bandwidth?
    |       |   rt-types:bandwidth-ieee-float32
    +--ro one-way-available-bandwidth-normality?
    |       |   te-types:performance-metrics-normality
    +--ro one-way-utilized-bandwidth?
    |       |   rt-types:bandwidth-ieee-float32
    +--ro one-way-utilized-bandwidth-normality?
    |       |   te-types:performance-metrics-normality
  +--ro performance-metrics-two-way
    |   +--ro two-way-delay?                               uint32
    |   +--ro two-way-delay-normality?
    |       |   te-types:performance-metrics-normality
  +--ro te-grouped-params*
    |   -> /te:te/tunnels/tunnel/te-kpi:te-telemetry/id
  +--ro grouping-operation?                               grouping-operation

```

## 7. YANG Data Model

### 7.1. ietf-te-kpi-telemetry model

The YANG code is as follows:

```
<CODE BEGINS> file "ietf-te-kpi-telemetry@2020-07-13.yang"
module ietf-te-kpi-telemetry {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-te-kpi-telemetry";
  prefix te-tel;

  /* Import inet-types */

  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Data Types";
  }

  /* Import TE */

  import ietf-te {
    prefix te;
    reference
      "I-D.ietf-teas-yang-te: A YANG Data Model for Traffic
      Engineering Tunnels and Interfaces";
  }

  /* Import TE Common types */

  import ietf-te-types {
    prefix te-types;
    reference
      "I-D.ietf-teas-yang-te-types: Traffic Engineering Common
      YANG 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: Young Lee <leeyoung@huawei.com>
    Dhruv Dhody <dhruv.ietf@gmail.com>";
  description
    "This module describes YANG data model for performance
```

monitoring telemetry for te tunnels.

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 (<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.

The key words '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 BCP 14 (RFC 2119) (RFC 8174) when, and only when, they appear in all capitals, as shown here.";

/\* Note: The RFC Editor will replace XXXX with the number assigned to the RFC once draft-ietf-teas-pm-telemetry-autonomics becomes an RFC.\*/

```
revision 2020-03-08 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: YANG models for VN/TE Performance Monitoring
    Telemetry and Scaling Intent Autonomics";
}

identity telemetry-param-type {
  description
    "Base identity for telemetry param types";
}

identity one-way-delay {
  base telemetry-param-type;
  description
    "To specify average Delay in one (forward)
    direction";
  reference
    "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
    RFC8570: IS-IS Traffic Engineering (TE) Metric Extensions.
    RFC7823: Performance-Based Path Selection for Explicitly
    Routed Label Switched Paths (LSPs) Using TE Metric
```

```
    Extensions";
}

identity two-way-delay {
  base telemetry-param-type;
  description
    "To specify average Delay in both (forward and reverse)
    directions";
  reference
    "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
    RFC8570: IS-IS Traffic Engineering (TE) Metric Extensions.
    RFC7823: Performance-Based Path Selection for Explicitly
    Routed Label Switched Paths (LSPs) Using TE Metric
    Extensions";
}

identity one-way-delay-variation {
  base telemetry-param-type;
  description
    "To specify average Delay Variation in one (forward) direction";
  reference
    "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
    RFC8570: IS-IS Traffic Engineering (TE) Metric Extensions.
    RFC7823: Performance-Based Path Selection for Explicitly
    Routed Label Switched Paths (LSPs) Using TE Metric
    Extensions";
}

identity two-way-delay-variation {
  base telemetry-param-type;
  description
    "To specify average Delay Variation in both (forward and reverse)
    directions";
  reference
    "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
    RFC8570: IS-IS Traffic Engineering (TE) Metric Extensions.
    RFC7823: Performance-Based Path Selection for Explicitly
    Routed Label Switched Paths (LSPs) Using TE Metric
    Extensions";
}

identity utilized-bandwidth {
  base telemetry-param-type;
  description
    "To specify utilized bandwidth over the specified source
    and destination.";
  reference
    "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
```

```
    RFC8570: IS-IS Traffic Engineering (TE) Metric Extensions.
    RFC7823: Performance-Based Path Selection for Explicitly
    Routed Label Switched Paths (LSPs) Using TE Metric
    Extensions";
}

identity utilized-percentage {
    base telemetry-param-type;
    description
        "To specify utilization percentage of the entity
        (e.g., tunnel, link, etc.)";
}

/* Typedef */

typedef telemetry-id {
    type inet:uri;
    description
        "Identifier for telemetry data. The precise
        structure of the telemetry-id will be up to the
        implementation. The identifier SHOULD be chosen
        such that the same telemetry data will always be
        identified through the same identifier, even if
        the data model is instantiated in separate
        datastores.";
}

typedef scaling-criteria-operation {
    type enumeration {
        enum AND {
            description
                "AND operation";
        }
        enum OR {
            description
                "OR operation";
        }
    }
    description
        "Operations to analyze list of scaling criterias";
}

grouping scaling-duration {
    description
        "Base scaling criteria durations";
    leaf threshold-time {
        type uint32;
        units "seconds";
    }
}
```



```
        description
            "The duration for which the criteria must hold true";
    }
    leaf cooldown-time {
        type uint32;
        units "seconds";
        description
            "The duration after a scaling-in/scaling-out action has been
            triggered, for which there will be no further operation";
    }
}

grouping scaling-criteria {
    description
        "Grouping for scaling criteria";
    leaf performance-type {
        type identityref {
            base telemetry-param-type;
        }
        description
            "Reference to the tunnel level telemetry type";
    }
    leaf threshold-value {
        type string;
        description
            "Scaling threshold for the telemetry parameter type";
    }
}

grouping scaling-in-intent {
    description
        "Basic scaling in intent";
    uses scaling-duration;
    list scaling-condition {
        key "performance-type";
        description
            "Scaling conditions";
        uses scaling-criteria;
        leaf scale-in-operation-type {
            type scaling-criteria-operation;
            default "AND";
            description
                "Operation to be applied to check between scaling criterias
                to check if the scale in threshold condition has been met.
                Defaults to AND";
        }
    }
}
}
```

```
grouping scaling-out-intent {
  description
    "Basic scaling out intent";
  uses scaling-duration;
  list scaling-condition {
    key "performance-type";
    description
      "Scaling conditions";
    uses scaling-criteria;
    leaf scale-out-operation-type {
      type scaling-criteria-operation;
      default "OR";
      description
        "Operation to be applied to check between scaling criterias
        to check if the scale out threshold condition has been met.
        Defaults to OR";
    }
  }
}

augment "/te:te/te:tunnels/te:tunnel" {
  description
    "Augmentation parameters for config scaling-criteria TE
    tunnel topologies. Scale in/out criteria might be used
    for network autonomics in order the controller to react
    to a certain set of monitored params.";
  container te-scaling-intent {
    description
      "The scaling intent";
    container scale-in-intent {
      description
        "scale-in";
      uses scaling-in-intent;
    }
    container scale-out-intent {
      description
        "scale-out";
      uses scaling-out-intent;
    }
  }
  container te-telemetry {
    config false;
    description
      "Telemetry Data";
    leaf id {
      type telemetry-id;
      description
        "ID of telemetry data used for easy reference";
    }
  }
}
```

```
    }
    uses te-types:performance-metrics-attributes;
  }
}
```

<CODE ENDS>

## 7.2. ietf-vn-kpi-telemetry model

The YANG code is as follows:

```
<CODE BEGINS> file "ietf-vn-kpi-telemetry@2020-07-13.yang"
module ietf-vn-kpi-telemetry {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-vn-kpi-telemetry";
  prefix vn-kpi;

  /* Import VN */

  import ietf-vn {
    prefix vn;
    reference
      "I-D.ietf-teas-actn-vn-yang: A YANG Data Model for VN
      Operation";
  }

  /* Import TE */

  import ietf-te {
    prefix te;
    reference
      "I-D.ietf-teas-yang-te: A YANG Data Model for Traffic
      Engineering Tunnels and Interfaces";
  }

  /* Import TE Common types */

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

  /* Import TE KPI */

  import ietf-te-kpi-telemetry {
    prefix te-kpi;
```

```
reference
  "RFC XXXX: YANG models for VN/TE Performance Monitoring
  Telemetry and Scaling Intent Autonomics";
}

/* Note: The RFC Editor will replace XXXX with the number
  assigned to this draft.*/

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: Young Lee <younglee.tx@gmail.com>
  Dhruv Dhody <dhruv.ietf@gmail.com>";
description
  "This module describes YANG data models for performance
  monitoring telemetry for Virtual Network (VN).

  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
  (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.

  The key words '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 BCP 14 (RFC 2119) (RFC 8174) when, and only when,
  they appear in all capitals, as shown here.";

/* Note: The RFC Editor will replace XXXX with the number
  assigned to the RFC once draft-lee-teas-pm-telemetry-
  autonomics becomes an RFC.*/

revision 2020-07-13 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: YANG models for VN/TE Performance Monitoring
```

```
        Telemetry and Scaling Intent Autonomics";
    }

typedef grouping-operation {
    type enumeration {
        enum MINIMUM {
            description
                "Select the minimum param";
        }
        enum MAXIMUM {
            description
                "Select the maximum param";
        }
        enum MEAN {
            description
                "Select the MEAN of the params";
        }
        enum STD_DEV {
            description
                "Select the standard deviation of the monitored params";
        }
        enum AND {
            description
                "Select the AND of the params";
        }
        enum OR {
            description
                "Select the OR of the params";
        }
    }
    description
        "Operations to analyze list of monitored params";
}

grouping vn-telemetry-param {
    description
        "augment of te-kpi:telemetry-param for VN specific params";
    leaf-list te-grouped-params {
        type leafref {
            path
                "/te:te/te:tunnels/te:tunnel/te-kpi:te-telemetry/te-kpi:id";
        }
        description
            "Allows the definition of a vn-telemetry param
            as a grouping of underlying TE params";
    }
    leaf grouping-operation {
        type grouping-operation;
    }
}
```

```
        description
            "describes the operation to apply to
            te-grouped-params";
    }
}

augment "/vn:vn/vn:vn-list" {
    description
        "Augmentation parameters for state TE VN topologies.";
    container vn-scaling-intent {
        description
            "scaling intent";
        container scale-in-intent {
            description
                "VN scale-in";
            uses te-kpi:scaling-in-intent;
        }
        container scale-out-intent {
            description
                "VN scale-out";
            uses te-kpi:scaling-out-intent;
        }
    }
    container vn-telemetry {
        config false;
        description
            "VN telemetry params";
        uses te-types:performance-metrics-attributes;
        leaf grouping-operation {
            type grouping-operation;
            description
                "describes the operation to apply to the VN-members";
        }
    }
}

augment "/vn:vn/vn:vn-list/vn:vn-member-list" {
    description
        "Augmentation parameters for state TE vn member topologies.";
    container vn-member-telemetry {
        config false;
        description
            "VN member telemetry params";
        uses te-types:performance-metrics-attributes;
        uses vn-telemetry-param;
    }
}
}
```

<CODE ENDS>

## 8. Security Considerations

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

The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF users to a preconfigured subset of all available NETCONF protocol operations and content. The NETCONF Protocol over Secure Shell (SSH) [RFC6242] describes a method for invoking and running NETCONF within a Secure Shell (SSH) session as an SSH subsystem. The Network Configuration Access Control Model (NACM) [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.

A number of configuration data nodes defined in this document are writable/deletable (i.e., "config true"). These data nodes may be considered sensitive or vulnerable in some network environments.

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. These are the subtrees and data nodes and their sensitivity/vulnerability:

- o /te:te/te:tunnels/te:tunnel/te-scaling-intent/scale-in-intent
- o /te:te/te:tunnels/te:tunnel/te-scaling-intent/scale-out-intent
- o /vn:vn/vn:vn-list/vn-scaling-intent/scale-in-intent
- o /vn:vn/vn:vn-list/vn-scaling-intent/scale-out-intent

## 9. IANA Considerations

This document registers the following namespace URIs in the IETF XML registry [RFC3688]:

---

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

---

---

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

---

This document registers the following YANG modules in the YANG Module.

Names registry [RFC7950]:

---

name: ietf-te-kpi-telemetry  
namespace: urn:ietf:params:xml:ns:yang:ietf-te-kpi-telemetry  
prefix: te-tel  
reference: RFC XXXX (TDB)

---

---

name: ietf-vn-kpi-telemetry  
namespace: urn:ietf:params:xml:ns:yang:ietf-vn-kpi-telemetry  
prefix: vn-tel  
reference: RFC XXXX (TDB)

---

## 10. Acknowledgements

We thank Rakesh Gandhi, Tarek Saad, Igor Bryskin and Kenichi Ogaki for useful discussions and their suggestions for this work.

## 11. References

### 11.1. Normative References

[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.ietf-teas-yang-te]  
Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin,  
"A YANG Data Model for Traffic Engineering Tunnels and  
Interfaces", draft-ietf-teas-yang-te-23 (work in  
progress), March 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>>.
- [RFC7926] Farrel, A., Ed., Drake, J., Bitar, N., Swallow, G.,  
Ceccarelli, D., and X. Zhang, "Problem Statement and  
Architecture for Information Exchange between  
Interconnected Traffic-Engineered Networks", BCP 206,  
RFC 7926, DOI 10.17487/RFC7926, July 2016,  
<<https://www.rfc-editor.org/info/rfc7926>>.
- [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>>.

- [RFC8233] Dhody, D., Wu, Q., Manral, V., Ali, Z., and K. Kumaki, "Extensions to the Path Computation Element Communication Protocol (PCEP) to Compute Service-Aware Label Switched Paths (LSPs)", RFC 8233, DOI 10.17487/RFC8233, September 2017, <<https://www.rfc-editor.org/info/rfc8233>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8640] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Dynamic Subscription to YANG Events and Datastores over NETCONF", RFC 8640, DOI 10.17487/RFC8640, September 2019, <<https://www.rfc-editor.org/info/rfc8640>>.
- [RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <<https://www.rfc-editor.org/info/rfc8641>>.
- [RFC8776] Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Common YANG Data Types for Traffic Engineering", RFC 8776, DOI 10.17487/RFC8776, June 2020, <<https://www.rfc-editor.org/info/rfc8776>>.

## 11.2. Informative References

- [I-D.xu-actn-perf-dynamic-service-control]  
Xu, Y., Zhang, G., Cheng, W., and z. zhenghaomian@huawei.com, "Use Cases and Requirements of Dynamic Service Control based on Performance Monitoring in ACTN Architecture", draft-xu-actn-perf-dynamic-service-control-03 (work in progress), April 2015.

- [RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015, <<https://www.rfc-editor.org/info/rfc7471>>.
- [RFC7823] Atlas, A., Drake, J., Giacalone, S., and S. Previdi, "Performance-Based Path Selection for Explicitly Routed Label Switched Paths (LSPs) Using TE Metric Extensions", RFC 7823, DOI 10.17487/RFC7823, May 2016, <<https://www.rfc-editor.org/info/rfc7823>>.
- [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>>.
- [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>>.
- [RFC8570] Ginsberg, L., Ed., Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 8570, DOI 10.17487/RFC8570, March 2019, <<https://www.rfc-editor.org/info/rfc8570>>.

#### Authors' Addresses

Young Lee (editor)  
Samsung Electronics

Email: [younglee.tx@gmail.com](mailto:younglee.tx@gmail.com)

Dhruv Dhody (editor)  
Huawei Technologies  
Divyashree Techno Park, Whitefield  
Bangalore, Karnataka 560066  
India

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

Satish Karunanithi  
Huawei Technologies  
Divyashree Techno Park, Whitefield  
Bangalore, Karnataka 560066  
India

Email: satish.karunanithi@gmail.com

Ricard Vilalta  
CTTC  
Centre Tecnologic de Telecomunicacions de Catalunya (CTTC/CERCA)  
Barcelona  
Spain

Email: ricard.vilalta@cttc.es

Daniel King  
Lancaster University

Email: d.king@lancaster.ac.uk

Daniele Ceccarelli  
Ericsson  
Torshamnsgatan, 48  
Stockholm, Sweden

Email: daniele.ceccarelli@ericsson.com

TEAS Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 14, 2021

Y. Lee, Ed.  
Samsung Electronics  
D. Dhody, Ed.  
Huawei Technologies  
D. Ceccarelli  
Ericsson  
I. Bryskin  
Individual  
B. Yoon  
ETRI  
July 13, 2020

A YANG Data Model for VN Operation  
draft-ietf-teas-actn-vn-yang-09

Abstract

This document provides a YANG data model generally applicable to any mode of Virtual Network (VN) operation.

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 14, 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 . . . . .	2
1.1.	Terminology . . . . .	4
1.1.1.	Requirements Language . . . . .	4
1.2.	Tree diagram . . . . .	4
1.3.	Prefixes in Data Node Names . . . . .	4
2.	Use-case of VN YANG Model in the ACTN context . . . . .	5
2.1.	Type 1 VN . . . . .	5
2.2.	Type 2 VN . . . . .	6
3.	High-Level Control Flows with Examples . . . . .	7
3.1.	Type 1 VN Illustration . . . . .	7
3.2.	Type 2 VN Illustration . . . . .	8
3.2.1.	VN and AP Usage . . . . .	11
4.	VN Model Usage . . . . .	12
4.1.	Customer view of VN . . . . .	12
4.2.	Auto-creation of VN by MDSC . . . . .	12
4.3.	Innovative Services . . . . .	12
4.3.1.	VN Compute . . . . .	12
4.3.2.	Multi-sources and Multi-destinations . . . . .	12
4.3.3.	Others . . . . .	13
4.3.4.	Summary . . . . .	14
5.	VN YANG Model (Tree Structure) . . . . .	14
6.	VN YANG Model . . . . .	16
7.	JSON Example . . . . .	27
7.1.	VN JSON . . . . .	27
7.2.	TE-topology JSON . . . . .	33
8.	Security Considerations . . . . .	49
9.	IANA Considerations . . . . .	51
10.	Acknowledgments . . . . .	51
11.	References . . . . .	51
11.1.	Normative References . . . . .	51
11.2.	Informative References . . . . .	53
Appendix A.	Performance Constraints . . . . .	54
Appendix B.	Contributors Addresses . . . . .	54
Authors' Addresses	. . . . .	55

## 1. Introduction

This document provides a YANG [RFC7950] data model generally applicable to any mode of Virtual Network (VN) operation.

The VN model defined in this document is applicable in generic sense as an independent model in and of itself. The VN model defined in this document can also work together with other customer service models such as L3SM [RFC8299], L2SM [RFC8466] and L1CSM [I-D.ietf-ccamp-llcsm-yang] to provide a complete life-cycle service management and operations.

The YANG model discussed in this document basically provides the following:

- o Characteristics of Access Points (APs) that describe customer's end point characteristics;
- o Characteristics of Virtual Network Access Points (VNAP) that describe how an AP is partitioned for multiple VNs sharing the AP and its reference to a Link Termination Point (LTP) of the Provider Edge (PE) Node;
- o Characteristics of Virtual Networks (VNs) that describe the customer's VN in terms of multiple VN Members comprising a VN, multi-source and/or multi-destination characteristics of the VN Member, the VN's reference to TE-topology's Abstract Node;

The actual VN instantiation and computation is performed with Connectivity Matrices sub-module of TE-Topology Model [I-D.ietf-teas-yang-te-topo] which provides TE network topology abstraction and management operation. Once TE-topology Model is used in triggering VN instantiation over the networks, TE-tunnel [I-D.ietf-teas-yang-te] Model will inevitably interact with TE-Topology model for setting up actual tunnels and LSPs under the tunnels.

Abstraction and Control of Traffic Engineered Networks (ACTN) describes a set of management and control functions used to operate one or more TE networks to construct virtual networks that can be represented to customers and that are built from abstractions of the underlying TE networks [RFC8453]. ACTN is the primary example of the usage of the VN YANG model.

Sections 2 and 3 provide the discussion of how the VN YANG model is applicable to the ACTN context where Virtual Network Service (VNS) operation is implemented for the Customer Network Controller (CNC)-Multi-Domain Service Coordinator (MSDC) interface (CMI).

The YANG model on the CMI is also known as customer service model in [RFC8309]. The YANG model discussed in this document is used to operate customer-driven VNs during the VN instantiation, VN computation, and its life-cycle service management and operations.

The VN operational state is included in the same tree as the configuration consistent with Network Management Datastore Architecture (NMDA) [RFC8342]. The origin of the data is indicated as per the origin metadata annotation.

## 1.1. Terminology

Refer to [RFC8453], [RFC7926], and [RFC8309] for the key terms used in this document.

### 1.1.1. Requirements Language

The key words "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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

## 1.2. Tree diagram

A simplified graphical representation of the data model is used in Section 5 of this this document. The meaning of the symbols in these diagrams is defined in [RFC8340].

## 1.3. Prefixes in Data Node Names

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

Prefix	YANG module	Reference
vn	ietf-vn	[RFCXXXX]
inet	ietf-inet-types	[RFC6991]
nw	ietf-network	[RFC8345]
nt	ietf-network-topology	[RFC8345]
te-types	ietf-te-types	[RFC8776]
te-topo	ietf-te-topology	[I-D.ietf-teas-yang-te-topo]

Table 1: Prefixes and corresponding YANG modules

Note: The RFC Editor will replace XXXX with the number assigned to the RFC once this draft becomes an RFC.



2. Use-case of VN YANG Model in the ACTN context

In this section, ACTN is being used to illustrate the general usage of the VN YANG model. The model presented in this section has the following ACTN context.

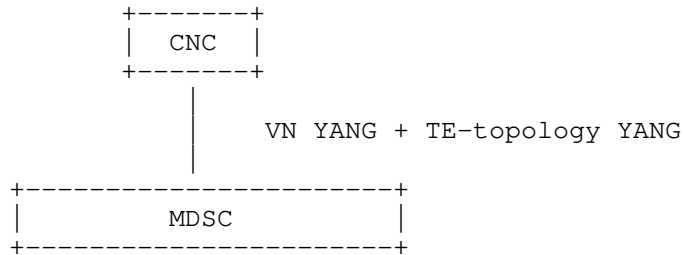


Figure 1: ACTN CMI

Both ACTN VN YANG and TE-topology models are used over the CMI to establish a VN over TE networks.

2.1. Type 1 VN

As defined in [RFC8453], a Virtual Network is a customer view of the TE network. To recapitulate VN types from [RFC8453], Type 1 VN is defined as follows:

The VN can be seen as a set of edge-to-edge abstract links (a Type 1 VN). Each abstract link is referred to as a VN member and is formed as an end-to-end tunnel across the underlying networks. Such tunnels may be constructed by recursive slicing or abstraction of paths in the underlying networks and can encompass edge points of the customer's network, access links, intra-domain paths, and inter-domain links.

If we were to create a VN where we have four VN-members as follows:

VN-Member 1	L1-L4
VN-Member 2	L1-L7
VN-Member 3	L2-L4
VN-Member 4	L3-L8

Where L1, L2, L3, L4, L7 and L8 correspond to a Customer End-Point, respectively.

This VN can be modeled as one abstract node representation as follows in Figure 2:

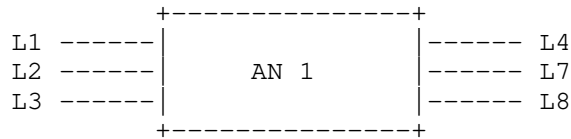


Figure 2: Abstract Node (One node topology)

Modeling a VN as one abstract node is the easiest way for customers to express their end-to-end connectivity; however, customers are not limited to express their VN only with one abstract node.

2.2. Type 2 VN

For some VN members of a VN, the customers are allowed to configure the actual path (i.e., detailed virtual nodes and virtual links) over the VN/abstract topology agreed mutually between CNC and MDSC prior to or a topology created by the MDSC as part of VN instantiation. Type 1 VN is a higher abstraction of a Type 2 VN.

If a Type 2 VN is desired for some or all of VN members of a type 1 VN (see the example in Section 2.1), the TE-topology model can provide the following abstract topology (that consists of virtual nodes and virtual links) which is built under the Type 1 VN.

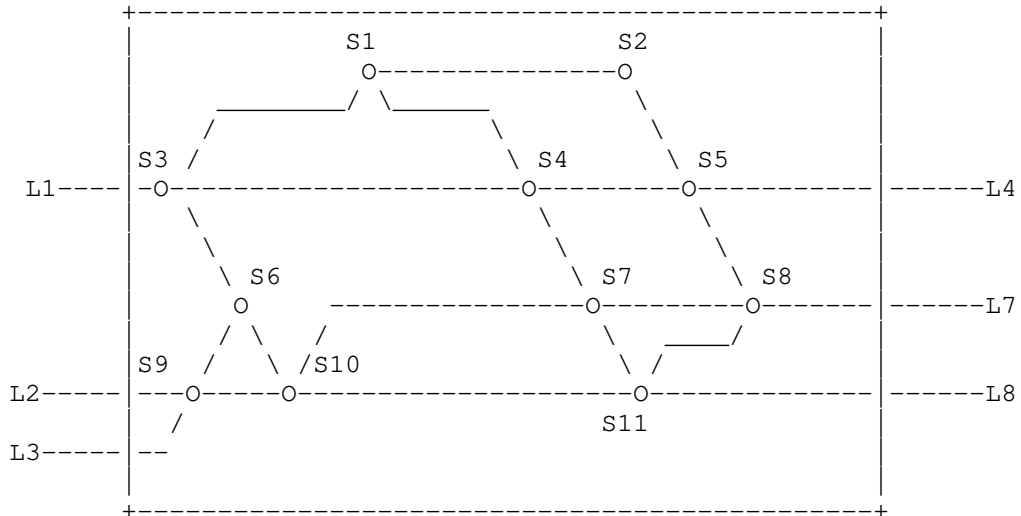


Figure 3: Type 2 topology

As you see from Figure 3, the Type 1 abstract node is depicted as a Type 1 abstract topology comprising of detailed virtual nodes and virtual links.

As an example, if VN-member 1 (L1-L4) is chosen to configure its own path over Type 2 topology, it can select, say, a path that consists of the ERO {S3,S4,S5} based on the topology and its service requirement. This capability is enacted via TE-topology configuration by the customer.

### 3. High-Level Control Flows with Examples

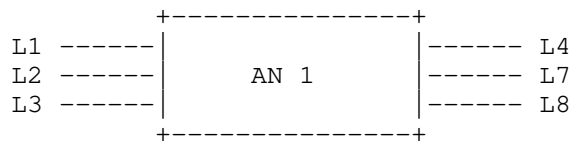
#### 3.1. Type 1 VN Illustration

If we were to create a VN where we have four VN-members as follows:

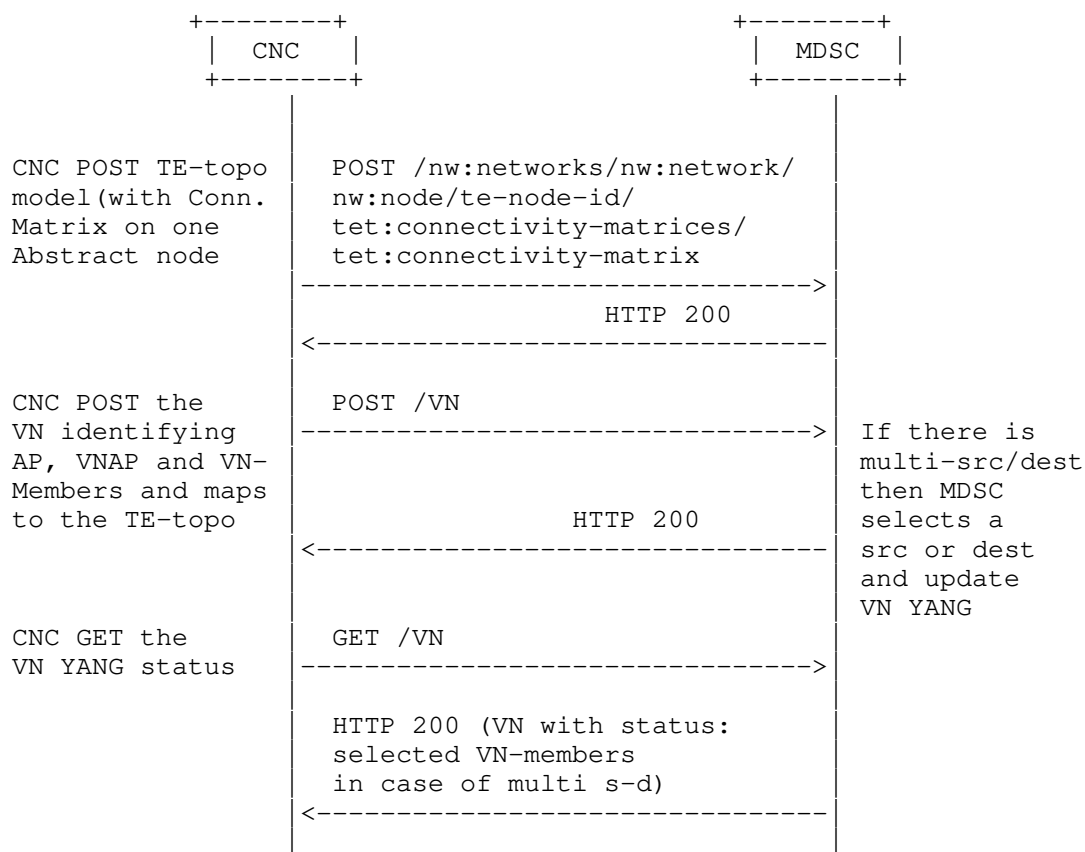
VN-Member 1	L1-L4
VN-Member 2	L1-L7
VN-Member 3	L2-L4
VN-Member 4	L3-L8

Where L1, L2, L3, L4, L7 and L8 correspond to Access Points.

This VN can be modeled as one abstract node representation as follows:



If this VN is Type 1, the following diagram shows the message flow between CNC and MDSC to instantiate this VN using VN and TE-Topology Models.

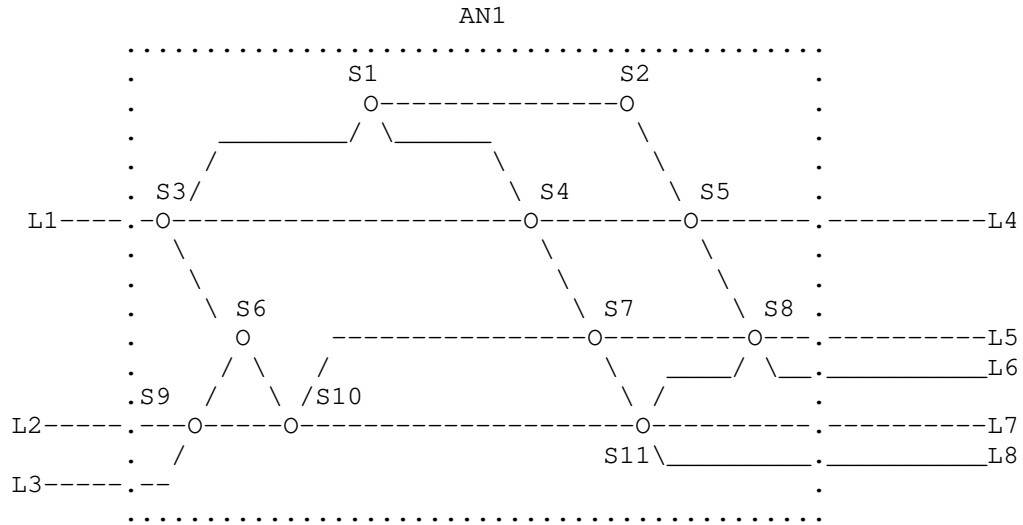


### 3.2. Type 2 VN Illustration

For some VN members, the customer may want to "configure" explicit routes over the path that connects its two end-points. Let us consider the following example.

- VN-Member 1 L1-L4 (via S3, S4, and S5)
- VN-Member 2 L1-L7 (via S3, S4, S7 and S8)
- VN-Member 3 L2-L7 (via S9, S10, and S11)
- VN-Member 4 L3-L8 (via S9, S10 and S11)

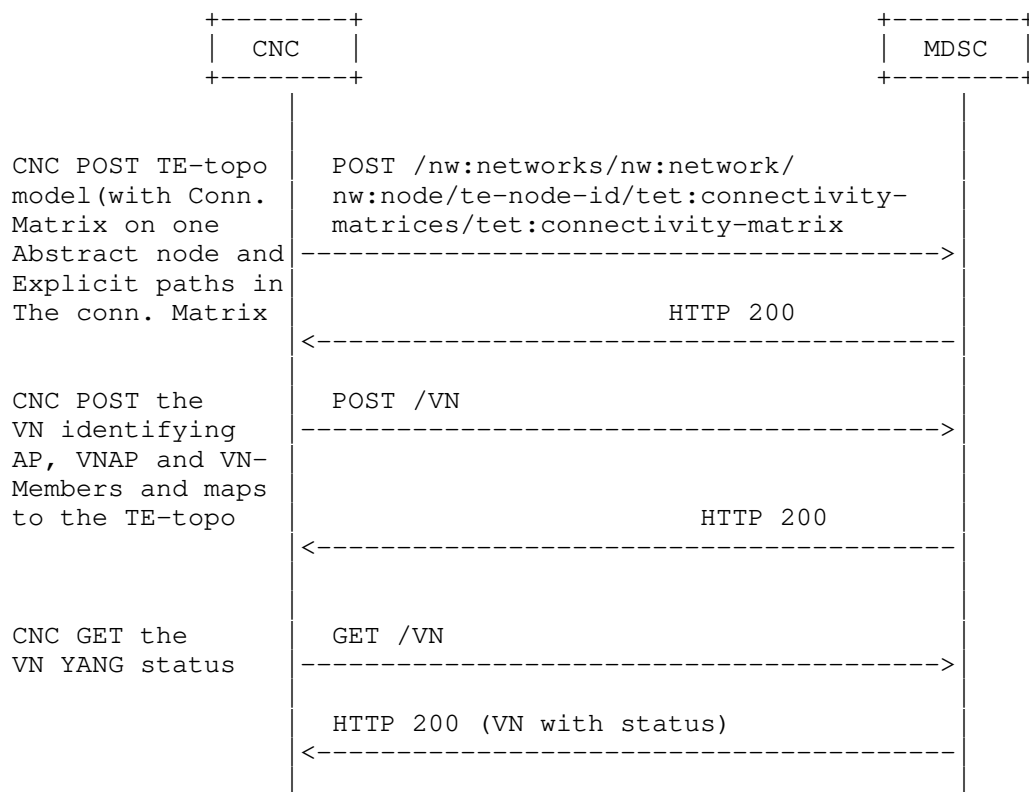
Where the following topology is the underlay for Abstraction Node 1 (AN1).



There are two options depending on whether CNC or MDSC creates the single abstract node topology.

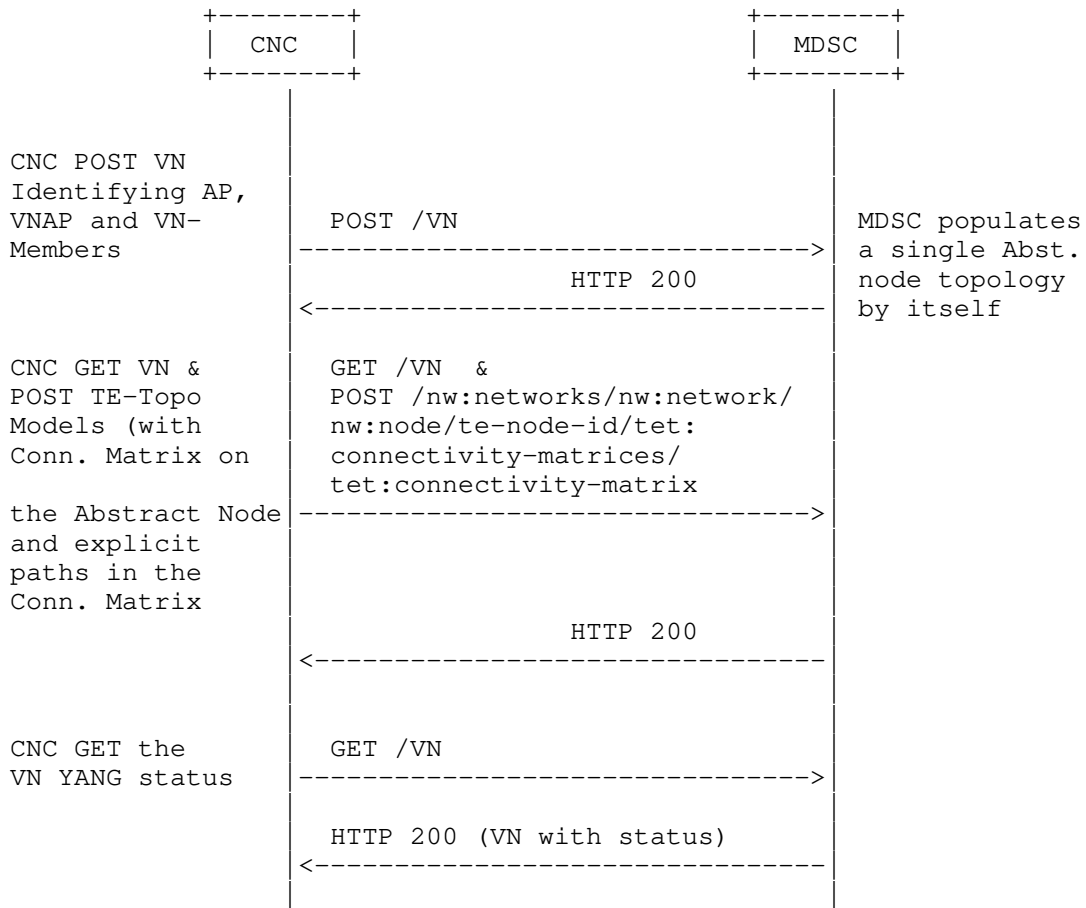
Case 1:

If CNC creates the single abstract node topology, the following diagram shows the message flow between CNC and MDSC to instantiate this VN using VN and TE-Topology Model.



Case 2:

On the other hand, if MDSC create the single abstract node topology based VN YANG posted by the CNC, the following diagram shows the message flow between CNC and MDSC to instantiate this VN using VN and TE-Topology Models.



Section 7 provides JSON examples for both VN model and TE-topology Connectivity Matrix sub-model to illustrate how a VN can be created by the CNC making use of the VN module as well as the TE-topology Connectivity Matrix module.

### 3.2.1. VN and AP Usage

The customer access information may be known at the time of VN creation. A shared logical AP identifier is used between the customer and the operator to identify the access link between Customer Edge (CE) and Provider Edge (PE) . This is described in Section 6 of [RFC8453].

In some VN operations, the customer access may not be known at the initial VN creation. The VN operation allow a creation of VN with

only PE identifier as well. The customer access information could be added later.

To achieve this the 'ap' container has a leaf for 'pe' node that allows AP to be created with PE information. The vn-member (and vn) could use APs that only have PE information initially.

#### 4. VN Model Usage

##### 4.1. Customer view of VN

The VN-YANG model allows to define a customer view, and allows the customer to communicate using the VN constructs as described in the [RFC8454]. It also allows to group the set of edge-to-edge links (i.e., VN members) under a common umbrella of VN. This allows the customer to instantiate and view the VN as one entity, making it easier for some customers to work on VN without worrying about the details of the provider based YANG models.

This is similar to the benefits of having a separate YANG model for the customer services as described in [RFC8309], which states that service models do not make any assumption of how a service is actually engineered and delivered for a customer.

##### 4.2. Auto-creation of VN by MDSC

The VN could be configured at the MDSC explicitly by the CNC using the VN YANG model. In some other cases, the VN is not explicitly configured, but created automatically by the MDSC based on the customer service model and local policy, even in these case the VN YANG model can be used by the CNC to learn details of the underlying VN created to meet the requirements of customer service model.

##### 4.3. Innovative Services

###### 4.3.1. VN Compute

VN Model supports VN compute (pre-instantiation mode) to view the full VN as a single entity before instantiation. Achieving this via path computation or "compute only" tunnel setup does not provide the same functionality.

###### 4.3.2. Multi-sources and Multi-destinations

In creating a virtual network, the list of sources or destinations or both may not be pre-determined by the customer. For instance, for a given source, there may be a list of multiple-destinations to which the optimal destination may be chosen depending on the network



resource situations. Likewise, for a given destination, there may also be multiple-sources from which the optimal source may be chosen. In some cases, there may be a pool of multiple sources and destinations from which the optimal source-destination may be chosen. The following YANG module is shown for describing source container and destination container. The following YANG tree shows how to model multi-sources and multi-destinations.

```

+--rw vn
  +--rw vn-list* [vn-id]
    +--rw vn-id          vn-id
    +--rw vn-topology-id?  te-types:te-topology-id
    +--rw abstract-node?
      |   -> /nw:networks/network/node/tet:te-node-id
    +--rw vn-member-list* [vn-member-id]
      +--rw vn-member-id          vn-member-id
      +--rw src
        +--rw src?
          |   -> /ap/access-point-list/access-point-id
        +--rw src-vn-ap-id?
          |   -> /ap/access-point-list/vn-ap/vn-ap-id
        +--rw multi-src?          boolean {multi-src-dest}?
      +--rw dest
        +--rw dest?
          |   -> /ap/access-point-list/access-point-id
        +--rw dest-vn-ap-id?
          |   -> /ap/access-point-list/vn-ap/vn-ap-id
        +--rw multi-dest?          boolean {multi-src-dest}?
      +--rw connectivity-matrix-id?  leafref
      +--ro oper-status?             identityref
    +--ro if-selected?              boolean {multi-src-dest}?
    +--rw admin-status?             identityref
    +--ro oper-status?             identityref
    +--rw vn-level-diversity?       te-types:te-path-disjointness

```

#### 4.3.3. Others

The VN YANG model can be easily augmented to support the mapping of VN to the Services such as L3SM and L2SM as described in [I-D.ietf-teas-te-service-mapping-yang].

The VN YANG model can be extended to support telemetry, performance monitoring and network autonomics as described in [I-D.ietf-teas-actn-pm-telemetry-autonomics].

## 4.3.4. Summary

This section summarizes the innovative service features of the VN YANG.

- o Maintenance of AP and VNAP along with VN
- o VN construct to group of edge-to-edge links
- o VN Compute (pre-instantiate)
- o Multi-Source / Multi-Destination
- o Ability to support various VN and VNS Types
  - \* VN Type 1: Customer configures the VN as a set of VN Members. No other details need to be set by customer, making for a simplified operations for the customer.
  - \* VN Type 2: Along with VN Members, the customer could also provide an abstract topology, this topology is provided by the Abstract TE Topology YANG Model.

## 5. VN YANG Model (Tree Structure)

```

module: ietf-vn
+--rw ap
|
|  +--rw access-point-list* [access-point-id]
|  |   +--rw access-point-id    access-point-id
|  |   +--rw pe?
|  |   |       -> /nw:networks/network/node/tet:te-node-id
|  |   +--rw max-bandwidth?     te-types:te-bandwidth
|  |   +--rw avl-bandwidth?     te-types:te-bandwidth
|  |   +--rw vn-ap* [vn-ap-id]
|  |   |   +--rw vn-ap-id        access-point-id
|  |   |   +--rw vn?            -> /vn/vn-list/vn-id
|  |   |   +--rw abstract-node?
|  |   |   |       -> /nw:networks/network/node/tet:te-node-id
|  |   |   +--rw ltp?           leafref
|  |   |   +--ro max-bandwidth? te-types:te-bandwidth
|  +--rw vn
|  |   +--rw vn-list* [vn-id]
|  |   |   +--rw vn-id            vn-id
|  |   |   +--rw vn-topology-id? te-types:te-topology-id
|  |   |   +--rw abstract-node?
|  |   |   |       -> /nw:networks/network/node/tet:te-node-id
|  |   |   +--rw vn-member-list* [vn-member-id]
|  |   |   |   +--rw vn-member-id    vn-member-id

```

```

+--rw src
|   +--rw src?
|   |       -> /ap/access-point-list/access-point-id
+--rw src-vn-ap-id?
|   -> /ap/access-point-list/vn-ap/vn-ap-id
+--rw multi-src?      boolean {multi-src-dest}?
+--rw dest
|   +--rw dest?
|   |       -> /ap/access-point-list/access-point-id
+--rw dest-vn-ap-id?
|   -> /ap/access-point-list/vn-ap/vn-ap-id
+--rw multi-dest?    boolean {multi-src-dest}?
+--rw connectivity-matrix-id? leafref
+--ro oper-status?   identityref
+--ro if-selected?   boolean {multi-src-dest}?
+--rw admin-status? identityref
+--ro oper-status?   identityref
+--rw vn-level-diversity? te-types:te-path-disjointness

```

## rpcs:

```

+---x vn-compute
+---w input
|   +---w abstract-node?
|   |       -> /nw:networks/network/node/tet:te-node-id
+---w vn-member-list* [vn-member-id]
|   +---w vn-member-id          vn-member-id
|   +---w src
|   |   +---w src?
|   |   |       -> /ap/access-point-list/access-point-id
|   |   +---w src-vn-ap-id?
|   |   |       -> /ap/access-point-list/vn-ap/vn-ap-id
|   |   +---w multi-src?      boolean {multi-src-dest}?
|   +---w dest
|   |   +---w dest?
|   |   |       -> /ap/access-point-list/access-point-id
|   |   +---w dest-vn-ap-id?
|   |   |       -> /ap/access-point-list/vn-ap/vn-ap-id
|   |   +---w multi-dest?    boolean {multi-src-dest}?
|   +---w connectivity-matrix-id? leafref
+---w vn-level-diversity? te-types:te-path-disjointness
+--ro output
+--ro vn-member-list* [vn-member-id]
+--ro vn-member-id          vn-member-id
+--ro src
|   +--ro src?
|   |       -> /ap/access-point-list/access-point-id
+--ro src-vn-ap-id?
|   -> /ap/access-point-list/vn-ap/vn-ap-id

```

```

| +--ro multi-src?          boolean {multi-src-dest}?
+--ro dest
| +--ro dest?
| |           -> /ap/access-point-list/access-point-id
| +--ro dest-vn-ap-id?
| |           -> /ap/access-point-list/vn-ap/vn-ap-id
| +--ro multi-dest?        boolean {multi-src-dest}?
+--ro connectivity-matrix-id? leafref
+--ro if-selected?         boolean
| {multi-src-dest}?
+--ro compute-status?      identityref

```

## 6. VN YANG Model

The YANG model is as follows:

```

<CODE BEGINS> file "ietf-vn@2020-07-13.yang"
module ietf-vn {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-vn";
  prefix vn;

  /* Import inet-types */

  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Data Types";
  }

  /* Import network */

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

  /* Import network topology */

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

  /* Import TE Common types */

```

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

/* Import TE Topology */

import ietf-te-topology {
  prefix tet;
  reference
    "I-D.ietf-teas-yang-te-topo: YANG Data Model for Traffic
    Engineering (TE) Topologies";
}

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: Young Lee <younglee.tx@gmail.com>
        : Dhruv Dhody <dhruv.ietf@gmail.com>";
description
  "This module contains a YANG module for the VN. It describes a
  VN operation module that takes place in the context of the
  CNC-MDSC Interface (CMI) of the ACTN architecture where the
  CNC is the actor of a VN Instantiation/modification/deletion
  as per RFC 8453.

  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
  (https://trustee.ietf.org/license-info).
```

```
revision 2020-07-13 {
  description
    "initial version.";
  reference
    "RFC XXXX: A YANG Data Model for VN Operation";
}

/* Features */

feature multi-src-dest {
  description
    "Support for selection of one src or destination
    among multiple.";
  reference
    "RFC 8453: Framework for Abstraction and Control of TE
    Networks (ACTN)";
}

/* Identity VN State*/

identity vn-state-type {
  description
    "Base identity for VN state";
}

identity vn-state-up {
  base vn-state-type;
  description
    "VN state up";
}

identity vn-state-down {
  base vn-state-type;
  description
    "VN state down";
}

/* Identity VN Admin State*/

identity vn-admin-state-type {
  description
    "Base identity for VN admin states";
}

identity vn-admin-state-up {
  base vn-admin-state-type;
  description
    "VN administratively state up";
}
```

```
    }

    identity vn-admin-state-down {
      base vn-admin-state-type;
      description
        "VN administratively state down";
    }

    /* Identity VN Compute State*/

    identity vn-compute-state-type {
      description
        "Base identity for compute states";
    }

    identity vn-compute-state-computing {
      base vn-compute-state-type;
      description
        "State path compute in progress";
    }

    identity vn-compute-state-computation-ok {
      base vn-compute-state-type;
      description
        "State path compute successful";
    }

    identity vn-compute-state-computation-failed {
      base vn-compute-state-type;
      description
        "State path compute failed";
    }

    /* Typedef */

    typedef vn-id {
      type inet:uri;
      description
        "Identifier for a VN. The precise structure of the
        vn-id will be up to the implementation. The
        identifier SHOULD be chosen such that the same VN
        will always be identified through the same
        identifier, even if the data model is instantiated
        in separate datastores.";
    }

    typedef access-point-id {
      type inet:uri;
    }
```

```
description
  "Identifier for an AP. The precise structure of the
  access-point-id will be up to the implementation.
  The identifier SHOULD be chosen such that the same AP
  will always be identified through the same
  identifier, even if the data model is instantiated
  in separate datastores. This type is used for both AP
  and VNAP";
}

typedef vn-member-id {
  type inet:uri;
  description
    "Identifier for a VN member. The precise structure of
    the vn-member-id will be up to the implementation.
    The identifier SHOULD be chosen such that the same VN
    member will always be identified through the same
    identifier, even if the data model is instantiated
    in separate datastores. ";
}

/* Groupings */

grouping vn-ap {
  description
    "VNAP related information";
  leaf vn-ap-id {
    type access-point-id;
    description
      "A unique identifier for the referred VNAP";
  }
  leaf vn {
    type leafref {
      path "/vn/vn-list/vn-id";
    }
    description
      "A reference to the VN";
  }
  leaf abstract-node {
    type leafref {
      path "/nw:networks/nw:network/nw:node/tet:te-node-id";
    }
    description
      "A reference to the abstract node in TE Topology that
      represent the VN";
  }
  leaf ltp {
    type leafref {
```



```
        path "/nw:networks/nw:network/nw:node/"
            + "nt:termination-point/tet:te-tp-id";
    }
    description
        "A reference LTP in the TE-topology";
    reference
        "I-D.ietf-teas-yang-te-topo: YANG Data Model for Traffic
        Engineering (TE) Topologies";
    }
    leaf max-bandwidth {
        type te-types:te-bandwidth;
        config false;
        description
            "The max bandwidth of the VNAP";
    }
    reference
        "RFC 8453: Framework for Abstraction and Control of TE
        Networks (ACTN)";
} //vn-ap

grouping access-point {
    description
        "AP related information";
    leaf access-point-id {
        type access-point-id;
        description
            "A unique identifier for the referred access point";
    }
    leaf pe {
        type leafref {
            path "/nw:networks/nw:network/nw:node/tet:te-node-id";
        }
        description
            "A reference to the PE node in the native TE Topology";
    }
    leaf max-bandwidth {
        type te-types:te-bandwidth;
        description
            "The max bandwidth of the AP";
    }
    leaf avl-bandwidth {
        type te-types:te-bandwidth;
        description
            "The available bandwidth of the AP";
    }
}
/*add details and any other properties of AP,
not associated by a VN
CE port, PE port etc.
```

```
    */
    list vn-ap {
        key "vn-ap-id";
        uses vn-ap;
        description
            "List of VNAP in this AP";
    }
    reference
        "RFC 8453: Framework for Abstraction and Control of TE
        Networks (ACTN)";
} //access-point

grouping vn-member {
    description
        "The vn-member is described by this grouping";
    leaf vn-member-id {
        type vn-member-id;
        description
            "A vn-member identifier";
    }
    container src {
        description
            "The source of VN Member";
        leaf src {
            type leafref {
                path "/ap/access-point-list/access-point-id";
            }
            description
                "A reference to source AP";
        }
        leaf src-vn-ap-id {
            type leafref {
                path "/ap/access-point-list/vn-ap/vn-ap-id";
            }
            description
                "A reference to source VNAP";
        }
        leaf multi-src {
            if-feature "multi-src-dest";
            type boolean;
            description
                "Is the source part of multi-source, where
                only one of the source is enabled";
        }
    }
}
container dest {
    description
        "the destination of VN Member";
}
```

```
leaf dest {
  type leafref {
    path "/ap/access-point-list/access-point-id";
  }
  description
    "A reference to destination AP";
}
leaf dest-vn-ap-id {
  type leafref {
    path "/ap/access-point-list/vn-ap/vn-ap-id";
  }
  description
    "A reference to dest VNAP";
}
leaf multi-dest {
  if-feature "multi-src-dest";
  type boolean;
  description
    "Is destination part of multi-destination, where only one
    of the destination is enabled";
}
}
leaf connectivity-matrix-id {
  type leafref {
    path "/nw/networks/nw:network/nw:node/tet:te/"
      + "tet:te-node-attributes/"
      + "tet:connectivity-matrices/"
      + "tet:connectivity-matrix/tet:id";
  }
  description
    "A reference to connectivity-matrix";
  reference
    "I-D.ietf-teas-yang-te-topo: YANG Data Model for Traffic
    Engineering (TE) Topologies";
}
reference
  "RFC 8454: Information Model for Abstraction and Control of TE
  Networks (ACTN)";
} //vn-member

grouping vn-policy {
  description
    "policy for VN-level diverisity";
  leaf vn-level-diversity {
    type te-types:te-path-disjointness;
    description
      "The type of disjointness on the VN level (i.e., across all
      VN members)";
  }
}
```

```
    }
  }

  /* Configuration data nodes */

  container ap {
    description
      "AP configurations";
    list access-point-list {
      key "access-point-id";
      description
        "access-point identifier";
      uses access-point {
        description
          "The access-point information";
      }
    }
  }
  reference
    "RFC 8453: Framework for Abstraction and Control of TE
    Networks (ACTN)";
}
container vn {
  description
    "VN configurations";
  list vn-list {
    key "vn-id";
    description
      "A virtual network is identified by a vn-id";
    leaf vn-id {
      type vn-id;
      description
        "A unique VN identifier";
    }
    leaf vn-topology-id {
      type te-types:te-topology-id;
      description
        "An optional identifier to the TE Topology Model where the
        abstract nodes and links of the Topology can be found for
        Type 2 VNS";
    }
    leaf abstract-node {
      type leafref {
        path "/nw:networks/nw:network/nw:node/tet:te-node-id";
      }
      description
        "A reference to the abstract node in TE Topology";
    }
  }
  list vn-member-list {
```

```
    key "vn-member-id";
    description
      "List of vn-members in a VN";
    uses vn-member;
    leaf oper-status {
      type identityref {
        base vn-state-type;
      }
      config false;
      description
        "The vn-member operational state.";
    }
  }
  leaf if-selected {
    if-feature "multi-src-dest";
    type boolean;
    default "false";
    config false;
    description
      "Is the vn-member is selected among the multi-src/dest
      options";
  }
  leaf admin-status {
    type identityref {
      base vn-admin-state-type;
    }
    default "vn-admin-state-up";
    description
      "VN administrative state.";
  }
  leaf oper-status {
    type identityref {
      base vn-state-type;
    }
    config false;
    description
      "VN operational state.";
  }
  uses vn-policy;
} //vn-list
reference
  "RFC 8453: Framework for Abstraction and Control of TE
  Networks (ACTN)";
} //vn

/* RPC */

rpc vn-compute {
```

```
description
  "The VN computation without actual instantiation";
input {
  leaf abstract-node {
    type leafref {
      path "/nw:networks/nw:network/nw:node/tet:te-node-id";
    }
    description
      "A reference to the abstract node in TE Topology";
  }
  list vn-member-list {
    key "vn-member-id";
    description
      "List of VN-members in a VN";
    uses vn-member;
  }
  uses vn-policy;
}
output {
  list vn-member-list {
    key "vn-member-id";
    description
      "List of VN-members in a VN";
    uses vn-member;
    leaf if-selected {
      if-feature "multi-src-dest";
      type boolean;
      default "false";
      description
        "Is the vn-member is selected among the multi-src/dest
        options";
    }
    leaf compute-status {
      type identityref {
        base vn-compute-state-type;
      }
      description
        "The VN-member compute state.";
    }
  }
}
} //vn-compute
}

<CODE ENDS>
```

## 7. JSON Example

This section provides json implementation examples as to how VN YANG model and TE topology model are used together to instantiate virtual networks.

The example in this section includes following VN

- o VN1 (Type 1): Which maps to the single node topology abstract1 (node D1) and consist of VN Members 104 (L1 to L4), 107 (L1 to L7), 204 (L2 to L4), 308 (L3 to L8) and 108 (L1 to L8). We also show how disjointness (node, link, srlg) is supported in the example on the global level (i.e., connectivity matrices level).
- o VN2 (Type 2): Which maps to the single node topology abstract2 (node D2), this topology has an underlay topology (absolute) (see figure in section 3.2). This VN has a single VN member 105 (L1 to L5) and an underlay path (S4 and S7) has been set in the connectivity matrix of abstract2 topology;
- o VN3 (Type 1): This VN has a multi-source, multi-destination feature enable for VN Member 104 (L1 to L4)/107 (L1 to L7) {multi-src} and VN Member 204 (L2 to L4)/304 (L3 to L4) {multi-dest} usecase. The selected VN-member is known via the field "if-selected" and the corresponding connectivity-matrix-id.

Note that the VN YANG model also include the AP and VNAP which shows various VN using the same AP.

### 7.1. VN JSON

```
{
  "ap":{
    "access-point-list": [
      {
        "access-point-id": 101,
        "access-point-name": "101",
        "vn-ap": [
          {
            "vn-ap-id": 10101,
            "vn": 1,
            "abstract-node": "D1",
            "ltp": "1-0-1"
          },
          {
            "vn-ap-id": 10102,
            "vn": 2,
            "abstract-node": "D2",
```

```
        "ltp": "1-0-1"
      },
      {
        "vn-ap-id": 10103,
        "vn": 3,
        "abstract-node": "D3",
        "ltp": "1-0-1"
      },
    ]
  },
  {
    "access-point-id": 202,
    "access-point-name": "202",
    "vn-ap": [
      {
        "vn-ap-id": 20201,
        "vn": 1,
        "abstract-node": "D1",
        "ltp": "2-0-2"
      }
    ]
  },
  {
    "access-point-id": 303,
    "access-point-name": "303",
    "vn-ap": [
      {
        "vn-ap-id": 30301,
        "vn": 1,
        "abstract-node": "D1",
        "ltp": "3-0-3"
      },
      {
        "vn-ap-id": 30303,
        "vn": 3,
        "abstract-node": "D3",
        "ltp": "3-0-3"
      }
    ]
  },
  {
    "access-point-id": 440,
    "access-point-name": "440",
    "vn-ap": [
      {
        "vn-ap-id": 44001,
        "vn": 1,
        "abstract-node": "D1",
```



```
        "ltp": "4-4-0"
      }
    ]
  },
  {
    "access-point-id": 550,
    "access-point-name": "550",
    "vn-ap": [
      {
        "vn-ap-id": 55002,
        "vn": 2,
        "abstract-node": "D2",
        "ltp": "5-5-0"
      }
    ]
  },
  {
    "access-point-id": 770,
    "access-point-name": "770",
    "vn-ap": [
      {
        "vn-ap-id": 77001,
        "vn": 1,
        "abstract-node": "D1",
        "ltp": "7-7-0"
      },
      {
        "vn-ap-id": 77003,
        "vn": 3,
        "abstract-node": "D3",
        "ltp": "7-7-0"
      }
    ]
  },
  {
    "access-point-id": 880,
    "access-point-name": "880",
    "vn-ap": [
      {
        "vn-ap-id": 88001,
        "vn": 1,
        "abstract-node": "D1",
        "ltp": "8-8-0"
      },
      {
        "vn-ap-id": 88003,
        "vn": 3,
        "abstract-node": "D3",
```

```

        "ltp": "8-8-0"
      }
    ]
  },
  "vn":{
    "vn-list": [
      {
        "vn-id": 1,
        "vn-name": "vn1",
        "vn-topology-id": "te-topology:abstract1",
        "abstract-node": "D1",
        "vn-member-list": [
          {
            "vn-member-id": 104,
            "src": {
              "src": 101,
              "src-vn-ap-id": 10101,
            },
            "dest": {
              "dest": 440,
              "dest-vn-ap-id": 44001,
            },
            "connectivity-matrix-id": 104
          },
          {
            "vn-member-id": 107,
            "src": {
              "src": 101,
              "src-vn-ap-id": 10101,
            },
            "dest": {
              "dest": 770,
              "dest-vn-ap-id": 77001,
            },
            "connectivity-matrix-id": 107
          },
          {
            "vn-member-id": 204,
            "src": {
              "src": 202,
              "dest-vn-ap-id": 20401,
            },
            "dest": {
              "dest": 440,
              "dest-vn-ap-id": 44001,
            },
          },
        ]
      }
    ]
  }
}

```

```

        "connectivity-matrix-id": 204
    },
    {
        "vn-member-id": 308,
        "src": {
            "src": 303,
            "src-vn-ap-id": 30301,
        },
        "dest": {
            "dest": 880,
            "src-vn-ap-id": 88001,
        },
        "connectivity-matrix-id": 308
    },
    {
        "vn-member-id": 108,
        "src": {
            "src": 101,
            "src-vn-ap-id": 10101,
        },
        "dest": {
            "dest": 880,
            "dest-vn-ap-id": 88001,
        },
        "connectivity-matrix-id": 108
    }
]
},
{
    "vn-id": 2,
    "vn-name": "vn2",
    "vn-topology-id": "te-topology:abstract2",
    "abstract-node": "D2",
    "vn-member-list": [
        {
            "vn-member-id": 105,
            "src": {
                "src": 101,
                "src-vn-ap-id": 10102,
            },
            "dest": {
                "dest": 550,
                "dest-vn-ap-id": 55002,
            },
            "connectivity-matrix-id": 105
        }
    ]
},

```

```
{
  "vn-id": 3,
  "vn-name": "vn3",
  "vn-topology-id": "te-topology:abstract3",
  "abstract-node": "D3",
  "vn-member-list": [
    {
      "vn-member-id": 104,
      "src": {
        "src": 101,
      },
      "dest": {
        "dest": 440,
        "multi-dest": true
      }
    },
    {
      "vn-member-id": 107,
      "src": {
        "src": 101,
        "src-vn-ap-id": 10103,
      },
      "dest": {
        "dest": 770,
        "dest-vn-ap-id": 77003,
        "multi-dest": true
      },
      "connectivity-matrix-id": 107,
      "if-selected": true,
    },
    {
      "vn-member-id": 204,
      "src": {
        "src": 202,
        "multi-src": true,
      },
      "dest": {
        "dest": 440,
      },
    },
    {
      "vn-member-id": 304,
      "src": {
        "src": 303,
        "src-vn-ap-id": 30303,
        "multi-src": true,
      },
      "dest": {
```

```
        "dest": 440,
        "src-vn-ap-id": 44003,
      },
      "connectivity-matrix-id": 304,
      "if-selected":true,
    },
  ],
},
]
}
}
}
```

## 7.2. TE-topology JSON

```
{
  "networks": {
    "network": [
      {
        "network-types": {
          "te-topology": {}
        },
        "network-id": "abstract1",
        "provider-id": 201,
        "client-id": 600,
        "te-topology-id": "te-topology:abstract1",
        "node": [
          {
            "node-id": "D1",
            "te-node-id": "2.0.1.1",
            "te": {
              "te-node-attributes": {
                "domain-id" : 1,
                "is-abstract": [null],
                "connectivity-matrices": {
                  "is-allowed": true,
                  "path-constraints": {
                    "bandwidth-generic": {
                      "te-bandwidth": {
                        "generic": [
                          {
                            "generic": "0x1p10",
                          }
                        ]
                      }
                    }
                  }
                }
              }
            }
          }
        ]
      }
    ]
  }
}
```

```
    }
  }
  "disjointness": "node link srlg",
},
"connectivity-matrix": [
  {
    "id": 104,
    "from": "1-0-1",
    "to": "4-4-0"
  },
  {
    "id": 107,
    "from": "1-0-1",
    "to": "7-7-0"
  },
  {
    "id": 204,
    "from": "2-0-2",
    "to": "4-4-0"
  },
  {
    "id": 308,
    "from": "3-0-3",
    "to": "8-8-0"
  },
  {
    "id": 108,
    "from": "1-0-1",
    "to": "8-8-0"
  },
]
}
},
"termination-point": [
  {
    "tp-id": "1-0-1",
    "te-tp-id": 10001,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  }
]
```

```
    },
    {
      "tp-id": "1-1-0",
      "te-tp-id": 10100,
      "te": {
        "interface-switching-capability": [
          {
            "switching-capability": "switching-otn",
            "encoding": "lsp-encoding-oduk"
          }
        ]
      }
    },
    {
      "tp-id": "2-0-2",
      "te-tp-id": 20002,
      "te": {
        "interface-switching-capability": [
          {
            "switching-capability": "switching-otn",
            "encoding": "lsp-encoding-oduk"
          }
        ]
      }
    },
    {
      "tp-id": "2-2-0",
      "te-tp-id": 20200,
      "te": {
        "interface-switching-capability": [
          {
            "switching-capability": "switching-otn",
            "encoding": "lsp-encoding-oduk"
          }
        ]
      }
    },
    {
      "tp-id": "3-0-3",
      "te-tp-id": 30003,
      "te": {
        "interface-switching-capability": [
          {
            "switching-capability": "switching-otn",
            "encoding": "lsp-encoding-oduk"
          }
        ]
      }
    }
  ]
}
```

```
    }
  },
  {
    "tp-id": "3-3-0",
    "te-tp-id": 30300,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  },
  {
    "tp-id": "4-0-4",
    "te-tp-id": 40004,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  },
  {
    "tp-id": "4-4-0",
    "te-tp-id": 40400,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  },
  {
    "tp-id": "5-0-5",
    "te-tp-id": 50005,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  }
]
```



```
    }
  },
  {
    "tp-id": "5-5-0",
    "te-tp-id": 50500,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  },
  {
    "tp-id": "6-0-6",
    "te-tp-id": 60006,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  },
  {
    "tp-id": "6-6-0",
    "te-tp-id": 60600,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  },
  {
    "tp-id": "7-0-7",
    "te-tp-id": 70007,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  }
]
```

```

    }
  },
  {
    "tp-id": "7-7-0",
    "te-tp-id": 70700,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  },
  {
    "tp-id": "8-0-8",
    "te-tp-id": 80008,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  },
  {
    "tp-id": "8-8-0",
    "te-tp-id": 80800,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  }
]
},
{
  "network-types": {
    "te-topology": {}
  },
  "network-id": "abstract2",
  "provider-id": 201,

```

```
"client-id": 600,
"te-topology-id": "te-topology:abstract2",
"node": [
  {
    "node-id": "D2",
    "te-node-id": "2.0.1.2",
    "te": {
      "te-node-attributes": {
        "domain-id" : 1,
        "is-abstract": [null],
        "connectivity-matrices": {
          "is-allowed": true,
          "underlay": {
            "enabled": true
          },
        },
        "path-constraints": {
          "bandwidth-generic": {
            "te-bandwidth": {
              "generic": [
                {
                  "generic": "0x1p10"
                }
              ]
            }
          }
        },
        "optimizations": {
          "objective-function": {
            "objective-function-type":
              "of-maximize-residual-bandwidth"
          }
        },
        "connectivity-matrix": [
          {
            "id": 105,
            "from": "1-0-1",
            "to": "5-5-0",
            "underlay": {
              "enabled": true,
              "primary-path": {
                "network-ref": "absolute",
                "path-element": [
                  {
                    "path-element-id": 1,
                    "index": 1,
                    "numbered-hop": {
                      "address": "4.4.4.4",
```

```

        "hop-type": "STRICT"
      }
    },
    {
      "path-element-id": 2,
      "index": 2,
      "numbered-hop": {
        "address": "7.7.7.7",
        "hop-type": "STRICT"
      }
    }
  ]
}
}
}
},
"termination-point": [
  {
    "tp-id": "1-0-1",
    "te-tp-id": 10001,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  },
  {
    "tp-id": "1-1-0",
    "te-tp-id": 10100,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  },
  {
    "tp-id": "2-0-2",
    "te-tp-id": 20002,

```

```
"te": {
  "interface-switching-capability": [
    {
      "switching-capability": "switching-otn",
      "encoding": "lsp-encoding-oduk"
    }
  ]
},
{
  "tp-id": "2-2-0",
  "te-tp-id": 20200,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "3-0-3",
  "te-tp-id": 30003,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "3-3-0",
  "te-tp-id": 30300,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "4-0-4",
  "te-tp-id": 40004,
```

```
"te": {
  "interface-switching-capability": [
    {
      "switching-capability": "switching-otn",
      "encoding": "lsp-encoding-oduk"
    }
  ]
},
{
  "tp-id": "4-4-0",
  "te-tp-id": 40400,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "5-0-5",
  "te-tp-id": 50005,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "5-5-0",
  "te-tp-id": 50500,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "6-0-6",
  "te-tp-id": 60006,
```

```
"te": {
  "interface-switching-capability": [
    {
      "switching-capability": "switching-otn",
      "encoding": "lsp-encoding-oduk"
    }
  ]
},
{
  "tp-id": "6-6-0",
  "te-tp-id": 60600,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "7-0-7",
  "te-tp-id": 70007,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "7-7-0",
  "te-tp-id": 70700,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "8-0-8",
  "te-tp-id": 80008,
```

```

    "te": {
        "interface-switching-capability": [
            {
                "switching-capability": "switching-otn",
                "encoding": "lsp-encoding-oduk"
            }
        ]
    },
    {
        "tp-id": "8-8-0",
        "te-tp-id": 80800,
        "te": {
            "interface-switching-capability": [
                {
                    "switching-capability": "switching-otn",
                    "encoding": "lsp-encoding-oduk"
                }
            ]
        }
    }
]
},
{
    "network-types": {
        "te-topology": {}
    },
    "network-id": "abstract3",
    "provider-id": 201,
    "client-id": 600,
    "te-topology-id": "te-topology:abstract3",
    "node": [
        {
            "node-id": "D3",
            "te-node-id": "3.0.1.1",
            "te": {
                "te-node-attributes": {
                    "domain-id" : 3,
                    "is-abstract": [null],
                    "connectivity-matrices": {
                        "is-allowed": true,
                        "path-constraints": {
                            "bandwidth-generic": {
                                "te-bandwidth": {
                                    "generic": [

```



```

        {
          "generic": "0x1p10",
        }
      ]
    }
  },
  "connectivity-matrix": [
    {
      "id": 107,
      "from": "1-0-1",
      "to": "7-7-0"
    },
    {
      "id": 308,
      "from": "3-0-3",
      "to": "8-8-0"
    },
  ],
]
}
},
"termination-point": [
  {
    "tp-id": "1-0-1",
    "te-tp-id": 10001,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  },
  {
    "tp-id": "1-1-0",
    "te-tp-id": 10100,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  },
]
},
},

```

```
{
  "tp-id": "2-0-2",
  "te-tp-id": 20002,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "2-2-0",
  "te-tp-id": 20200,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "3-0-3",
  "te-tp-id": 30003,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "3-3-0",
  "te-tp-id": 30300,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
},
```

```
{
  "tp-id": "4-0-4",
  "te-tp-id": 40004,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "4-4-0",
  "te-tp-id": 40400,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "5-0-5",
  "te-tp-id": 50005,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "5-5-0",
  "te-tp-id": 50500,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
}
```

```
    },
    {
      "tp-id": "6-0-6",
      "te-tp-id": 60006,
      "te": {
        "interface-switching-capability": [
          {
            "switching-capability": "switching-otn",
            "encoding": "lsp-encoding-oduk"
          }
        ]
      }
    },
    {
      "tp-id": "6-6-0",
      "te-tp-id": 60600,
      "te": {
        "interface-switching-capability": [
          {
            "switching-capability": "switching-otn",
            "encoding": "lsp-encoding-oduk"
          }
        ]
      }
    },
    {
      "tp-id": "7-0-7",
      "te-tp-id": 70007,
      "te": {
        "interface-switching-capability": [
          {
            "switching-capability": "switching-otn",
            "encoding": "lsp-encoding-oduk"
          }
        ]
      }
    },
    {
      "tp-id": "7-7-0",
      "te-tp-id": 70700,
      "te": {
        "interface-switching-capability": [
          {
            "switching-capability": "switching-otn",
            "encoding": "lsp-encoding-oduk"
          }
        ]
      }
    }
  ]
}
```

```

    },
    {
      "tp-id": "8-0-8",
      "te-tp-id": 80008,
      "te": {
        "interface-switching-capability": [
          {
            "switching-capability": "switching-otn",
            "encoding": "lsp-encoding-oduk"
          }
        ]
      }
    },
    {
      "tp-id": "8-8-0",
      "te-tp-id": 80800,
      "te": {
        "interface-switching-capability": [
          {
            "switching-capability": "switching-otn",
            "encoding": "lsp-encoding-oduk"
          }
        ]
      }
    }
  ]
},
]
}
}

```

## 8. Security Considerations

The configuration, state, and action data defined in this document are designed to be accessed via a management protocol with a secure transport layer, 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 users to a preconfigured subset of all available NETCONF protocol operations and content.

The model presented in this document is used in the interface between the Customer Network Controller (CNC) and Multi-Domain Service Coordinator (MDSC), which is referred to as CNC-MDSC Interface (CMI). Therefore, many security risks such as malicious attack and rogue elements attempting to connect to various ACTN components. Furthermore, some ACTN components (e.g., MSDC) represent a single point of failure and threat vector and must also manage policy conflicts and eavesdropping of communication between different ACTN components.

A number of configuration data nodes defined in this document are writable/deletable (i.e., "config true") These data nodes may be considered sensitive or vulnerable in some network environments.

These are the subtrees and data nodes and their sensitivity/vulnerability:

- o access-point-list:
  - \* access-point-id
  - \* max-bandwidth
  - \* avl-bandwidth
- o vn-ap:
  - \* vn-ap-id
  - \* vn
  - \* abstract-node
  - \* ltp
- o vn-list
  - \* vn-id
  - \* vn-topology-id
  - \* abstract-node
- o vn-member-id
  - \* src
  - \* src-vn-ap-id

- \* dest
- \* dest-vn-ap-id
- \* connectivity-matrix-id

## 9. IANA Considerations

This document registers the following namespace URIs in the IETF XML registry [RFC3688]:

---

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

---

This document registers the following YANG modules in the YANG Module Names registry [RFC6020]:

---

name: ietf-vn  
namespace: urn:ietf:params:xml:ns:yang:ietf-vn  
prefix: vn  
reference: RFC XXXX (TDB)

---

## 10. Acknowledgments

The authors would like to thank Xufeng Liu, Adrian Farrel, Tom Petch, and Kenichi Ogaki for their helpful comments and valuable suggestions.

## 11. References

### 11.1. Normative References

- [I-D.ietf-teas-yang-te]  
Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin,  
"A YANG Data Model for Traffic Engineering Tunnels and  
Interfaces", draft-ietf-teas-yang-te-23 (work in  
progress), March 2020.

- [I-D.ietf-teas-yang-te-topo]  
Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", draft-ietf-teas-yang-te-topo-22 (work in progress), June 2019.
- [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>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [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>>.
- [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <<https://www.rfc-editor.org/info/rfc8345>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8776] Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Common YANG Data Types for Traffic Engineering", RFC 8776, DOI 10.17487/RFC8776, June 2020, <<https://www.rfc-editor.org/info/rfc8776>>.

## 11.2. Informative References

- [I-D.ietf-ccamp-llcsm-yang]  
Lee, Y., Lee, K., Zheng, H., Dhody, D., Dios, O., and D. Ceccarelli, "A YANG Data Model for L1 Connectivity Service Model (L1CSM)", draft-ietf-ccamp-llcsm-yang-11 (work in progress), March 2020.
- [I-D.ietf-teas-actn-pm-telemetry-autonomics]  
Lee, Y., Dhody, D., Karunanithi, S., Vilata, R., King, D., and D. Ceccarelli, "YANG models for VN/TE Performance Monitoring Telemetry and Scaling Intent Autonomics", draft-ietf-teas-actn-pm-telemetry-autonomics-02 (work in progress), March 2020.
- [I-D.ietf-teas-te-service-mapping-yang]  
Lee, Y., Dhody, D., Fioccola, G., WU, Q., Ceccarelli, D., and J. Tantsura, "Traffic Engineering (TE) and Service Mapping Yang Model", draft-ietf-teas-te-service-mapping-yang-03 (work in progress), March 2020.

- [RFC7926] Farrel, A., Ed., Drake, J., Bitar, N., Swallow, G., Ceccarelli, D., and X. Zhang, "Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks", BCP 206, RFC 7926, DOI 10.17487/RFC7926, July 2016, <<https://www.rfc-editor.org/info/rfc7926>>.
- [RFC8299] Wu, Q., Ed., Litkowski, S., Tomotaki, L., and K. Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC 8299, DOI 10.17487/RFC8299, January 2018, <<https://www.rfc-editor.org/info/rfc8299>>.
- [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>>.
- [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>>.
- [RFC8454] Lee, Y., Belotti, S., Dhody, D., Ceccarelli, D., and B. Yoon, "Information Model for Abstraction and Control of TE Networks (ACTN)", RFC 8454, DOI 10.17487/RFC8454, September 2018, <<https://www.rfc-editor.org/info/rfc8454>>.
- [RFC8466] Wen, B., Fioccola, G., Ed., Xie, C., and L. Jalil, "A YANG Data Model for Layer 2 Virtual Private Network (L2VPN) Service Delivery", RFC 8466, DOI 10.17487/RFC8466, October 2018, <<https://www.rfc-editor.org/info/rfc8466>>.

#### Appendix A. Performance Constraints

At the time of creation of VN, it is natural to provide VN level constraints and optimization criteria. It should be noted that this YANG model rely on the TE-Topology Model [I-D.ietf-teas-yang-te-topo] by using a reference to an abstract node to achieve this. Further, connectivity-matrix structure is used to assign the constraints and optimization criteria include delay, jitter etc. [RFC8776] define some of the metric-types already and future documents are meant to augment it.

#### Appendix B. Contributors Addresses

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

Peter Park  
KT  
Email: peter.park@kt.com

Haomian Zheng  
Huawei Technologies  
Email: zhenghaomian@huawei.com

Xian Zhang  
Huawei Technologies  
Email: zhang.xian@huawei.com

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

Takuya Miyasaka  
KDDI  
Email: ta-miyasaka@kddi.com

#### Authors' Addresses

Young Lee (editor)  
Samsung Electronics  
Email: younglee.tx@gmail.com

Dhruv Dhody (editor)  
Huawei Technologies  
Divyashree Techno Park, Whitefield  
Bangalore, Karnataka 560066  
India  
Email: dhruv.ietf@gmail.com

Daniele Ceccarelli  
Ericsson  
Torshamnsgatan, 48  
Stockholm, Sweden  
Email: daniele.ceccarelli@ericsson.com

Igor Bryskin  
Individual

Email: [i\\_bryskin@yahoo.com](mailto:i_bryskin@yahoo.com)

Bin Yeong Yoon  
ETRI

Email: [byyun@etri.re.kr](mailto:byyun@etri.re.kr)

TEAS Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 14, 2021

Y. Lee, Ed.  
Samsung Electronics  
D. Dhody, Ed.  
G. Fioccola  
Q. Wu, Ed.  
Huawei Technologies  
D. Ceccarelli  
Ericsson  
J. Tantsura  
Apstra  
July 13, 2020

Traffic Engineering (TE) and Service Mapping Yang Model  
draft-ietf-teas-te-service-mapping-yang-04

Abstract

This document provides a YANG data model to map customer service models (e.g., the L3VPN Service Model (L3SM)) to Traffic Engineering (TE) models (e.g., the TE Tunnel or the Virtual Network (VN) model). This model is referred to as TE Service Mapping Model and is applicable generically to the operator's need for seamless control and management of their VPN services with TE tunnel support.

The model is principally used to allow monitoring and diagnostics of the management systems to show how the service requests are mapped onto underlying network resource and TE models.

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 14, 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	3
1.1.	Terminology	4
1.1.1.	Requirements Language	5
1.2.	Tree diagram	5
1.3.	Prefixes in Data Node Names	5
2.	TE and Service Related Parameters	6
2.1.	VN/Tunnel Selection Requirements	7
2.2.	Availability Requirement	8
3.	YANG Modeling Approach	8
3.1.	Forward Compatibility	9
3.2.	TE and Network Models	9
4.	L3VPN Architecture in the ACTN Context	10
4.1.	Service Mapping	13
4.2.	Site Mapping	13
5.	Applicability of TE-Service Mapping in Generic context	14
6.	YANG Data Trees	14
6.1.	Service Mapping Types	14
6.2.	Service Models	16
6.2.1.	L3SM	16
6.2.2.	L2SM	16
6.2.3.	L1CSM	17
6.3.	Network Models	18
6.3.1.	L3NM	18
6.3.2.	L2NM	19
7.	YANG Data Models	20
7.1.	ietf-te-service-mapping-types	20
7.2.	Service Models	29
7.2.1.	ietf-l3sm-te-service-mapping	29
7.2.2.	ietf-l2sm-te-service-mapping	31
7.2.3.	ietf-l1csm-te-service-mapping	33
7.3.	Network Models	35

7.3.1. ietf-l3nm-te-service-mapping . . . . .	35
7.3.2. ietf-l2nm-te-service-mapping . . . . .	37
8. Security Considerations . . . . .	39
9. IANA Considerations . . . . .	41
10. Acknowledgements . . . . .	42
11. References . . . . .	42
11.1. Normative References . . . . .	42
11.2. Informative References . . . . .	45
Appendix A. Contributor Addresses . . . . .	46
Authors' Addresses . . . . .	46

## 1. Introduction

Data models are a representation of objects that can be configured or monitored within a system. Within the IETF, YANG [RFC7950] is the language of choice for documenting data models, and YANG models have been produced to allow configuration or modelling of a variety of network devices, protocol instances, and network services. YANG data models have been classified in [RFC8199] and [RFC8309].

Framework for Abstraction and Control of Traffic Engineered Networks (ACTN) [RFC8453] introduces an architecture to support virtual network services and connectivity services.

[I-D.ietf-teas-actn-vn-yang] defines a YANG model and describes how customers or end-to-end orchestrator can request and/or instantiate a generic virtual network service. [I-D.ietf-teas-actn-yang] describes the way IETF YANG models of different classifications can be applied to the ACTN interfaces. In particular, it describes how customer service models can be mapped into the CNC-MDSC Interface (CMI) of the ACTN architecture.

The models presented in this document are also applicable in generic context [RFC8309] as part of Customer Service Model used between Service Orchestrator and Customer.

[RFC8299] provides a L3VPN service delivery YANG model for PE-based VPNs. The scope of that draft is limited to a set of domains under control of the same network operator to deliver services requiring TE tunnels.

[RFC8466] provides a L2VPN service delivery YANG model for PE-based VPNs. The scope of that draft is limited to a set of domains under control of the same network operator to deliver services requiring TE tunnels.

[I-D.ietf-ccamp-llcsm-yang] provides a L1 connectivity service delivery YANG model for PE-based VPNs. The scope of that draft is

limited to a set of domains under control of the same network operator to deliver services requiring TE tunnels.

While the IP/MPLS Provisioning Network Controller (PNC) is responsible for provisioning the VPN service on the Provider Edge (PE) nodes, the Multi-Domain Service Coordinator (MDSC) can coordinate how to map the VPN services onto Traffic Engineering (TE) tunnels. This is consistent with the two of the core functions of the MDSC specified in [RFC8453]:

- o Customer mapping/translation function: This function is to map customer requests/commands into network provisioning requests that can be sent to the PNC according to the business policies that have been provisioned statically or dynamically. Specifically, it provides mapping and translation of a customer's service request into a set of parameters that are specific to a network type and technology such that the network configuration process is made possible.
- o Virtual service coordination function: This function translates customer service-related information into virtual network service operations in order to seamlessly operate virtual networks while meeting a customer's service requirements. In the context of ACTN, service/virtual service coordination includes a number of service orchestration functions such as multi-destination load balancing, guarantees of service quality, bandwidth and throughput. It also includes notifications for service fault and performance degradation and so forth.

Section 2 describes a set of TE and service related parameters that this document addresses as "new and advanced parameters" that are not included in generic service models. Section 3 discusses YANG modelling approach.

Apart from the service model, the TE mapping is equally applicable to the Network Models (L3 VPN Service Network Model (L3NM) [I-D.ietf-opsawg-l3sm-l3nm], L2 VPN Service Network Model (L2NM) [I-D.ietf-opsawg-l2nm] etc.). See Section 3.2 for details.

### 1.1. Terminology

Refer to [RFC8453], [RFC7926], and [RFC8309] for the key terms used in this document.

The terminology for describing YANG data models is found in [RFC7950].



### 1.1.1. Requirements Language

The key words "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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

### 1.2. Tree diagram

A simplified graphical representation of the data model is used in Section 5 of this this document. The meaning of the symbols in these diagrams is defined in [RFC8340].

### 1.3. Prefixes in Data Node Names

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

Prefix	YANG module	Reference
inet	ietf-inet-types	[RFC6991]
tsm- types	ietf-te-service-mapping- types	[RFCXXXX]
l1csm	ietf-l1csm	[I-D.ietf-ccamp-l1csm-yang]
l2vpn- svc	ietf-l2vpn-svc	[RFC8466]
l3vpn- svc	ietf-l3vpn-svc	[RFC8299]
l1-tsm	ietf-l1csm-te-service- mapping	[RFCXXXX]
l2-tsm	ietf-l2sm-te-service- mapping	[RFCXXXX]
l3-tsm	ietf-l3sm-te-service- mapping	[RFCXXXX]
vn	ietf-vn	[I-D.ietf-teas-actn-vn-yang ]
nw	ietf-network	[RFC8345]
te- types	ietf-te-types	[RFC8776]
te	ietf-te	[I-D.ietf-teas-yang-te]
l2vpn- ntw	ietf-l2vpn-ntw	[I-D.ietf-opsawg-l2nm]
l3vpn- ntw	ietf-l3vpn-ntw	[I-D.ietf-opsawg-l3sm-l3nm]
rt	ietf-routing	[RFC8349]
sr- policy	ietf-sr-policy	[I-D.raza-spring-sr-policy- yang]

Table 1: Prefixes and corresponding YANG modules

Note: The RFC Editor should replace XXXX with the number assigned to the RFC once this draft becomes an RFC.

## 2. TE and Service Related Parameters

While L1/L2/L3 service models (L1CSM, L2SM, L3SM) are intended to provide service-specific parameters for VPN service instances, there are a number of TE Service related parameters that are not included in these service models.

Additional 'service parameters and policies' that are not included in the aforementioned service models are addressed in the YANG models defined in this document.

## 2.1. VN/Tunnel Selection Requirements

In some cases, the service requirements may need addition TE tunnels to be established. This may occur when there are no suitable existing TE tunnels that can support the service requirements, or when the operator would like to dynamically create and bind tunnels to the VPN such that they are not shared by other VPNs, for example, for network slicing. The establishment of TE tunnels is subject to the network operator's policies.

To summarize, there are three modes of VN/Tunnel selection operations to be supported as follows. Additional modes may be defined in the future.

- o New VN/Tunnel Binding - A customer could request a VPN service based on VN/Tunnels that are not shared with other existing or future services. This might be to meet VPN isolation requirements. Further, the YANG model described in Section 5 of this document can be used to describe the mapping between the VPN service and the ACTN VN. The VN (and TE tunnels) could be bound to the VPN and not used for any other VPN. Under this mode, the following sub-categories can be supported:
  1. Hard Isolation with deterministic characteristics: A customer could request a VPN service using a set of TE Tunnels with deterministic characteristics requirements (e.g., no latency variation) and where that set of TE Tunnels must not be shared with other VPN services and must not compete for bandwidth or other network resources with other TE Tunnels.
  2. Hard Isolation: This is similar to the above case but without the deterministic characteristics requirements.
  3. Soft Isolation: The customer requests a VPN service using a set of TE tunnels which can be shared with other VPN services.
- o VN/Tunnel Sharing - A customer could request a VPN service where new tunnels (or a VN) do not need to be created for each VPN and can be shared across multiple VPNs. Further, the mapping YANG model described in Section 5 of this document can be used to describe the mapping between the VPN service and the tunnels in use. No modification of the properties of a tunnel (or VN) is allowed in this mode: an existing tunnel can only be selected.
- o VN/Tunnel Modify - This mode allows the modification of the properties of the existing VN/tunnel (e.g., bandwidth).

- o TE Mapping Template - This mode allows a VPN service to be bound to a mapping template containing constraints and optimization criteria. This allows mapping with the underlay TE characteristics without first creating a VN or tunnels to map. The VPN service could be mapped to a template first. Once the VN/Tunnels are actually created/selected for the VPN service, this mode is no longer used and replaced with the above modes.

2.2. Availability Requirement

Availability is another service requirement or intent that may influence the selection or provisioning of TE tunnels or a VN to support the requested service. Availability is a probabilistic measure of the length of time that a VPN/VN instance functions without a network failure.

The availability level will need to be translated into network specific policies such as the protection/reroute policy associated with a VN or Tunnel. The means by which this is achieved is not in the scope of this document.

3. YANG Modeling Approach

This section provides how the TE and Service mapping parameters are supported using augmentation of the existing service models (i.e., [I-D.ietf-ccamp-llcsm-yang], [RFC8466], and [RFC8299]). Figure 1 shows the scope of the Augmented LxSM Model.

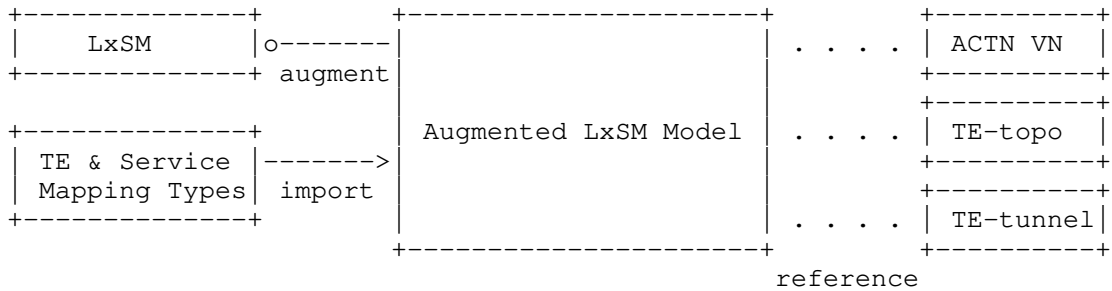


Figure 1: Augmented LxSM Model

The Augmented LxSM model (where x=1,2,3) augments the basic LxSM model while importing the common TE and Service related parameters (defined in Section 2) grouping information from TE and Service Mapping Types. The TE and Service Mapping Types (ietf-te-service-mapping-types) module is the repository of all common groupings imported by each augmented LxSM model. Any future service models would import this mapping-type common model.

The role of the augmented LxSm service model is to expose the mapping relationship between service models and TE models so that VN/VPN service instantiations provided by the underlying TE networks can be viewed outside of the MDSC, for example by an operator who is diagnosing the behaviour of the network. It also allows for the customers to access operational state information about how their services are instantiated with the underlying VN, TE topology or TE tunnels provided that the MDSC operator is willing to share that information. This mapping will facilitate a seamless service management operation with underlay-TE network visibility.

As seen in Figure 1, the augmented LxSM service model records a mapping between the customer service models and the ACTN VN YANG model. Thus, when the MDSC receives a service request it creates a VN that meets the customer's service objectives with various constraints via TE-topology model [I-D.ietf-teas-yang-te-topo], and this relationship is recorded by the Augmented LxSM Model. The model also supports a mapping between a service model and TE-topology or a TE-tunnel.

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

### 3.1. Forward Compatibility

The YANG module defined in this document supports three existing service models via augmenting while sharing the common TE and Service Mapping Types.

It is possible that new service models will be defined at some future time and that it will be desirable to map them to underlying TE constructs in the same way as the three existing models are augmented.

### 3.2. TE and Network Models

The L2/L3 network models (L2NM, L3NM) are intended to describe a VPN Service in the Service Provider Network. It contains information of the Service Provider network and might include allocated resources. It can be used by network controllers to manage and control the VPN Service configuration in the Service Provider network.

Similar to service model, the existing network models (i.e., [I-D.ietf-opsawg-l3sm-l3nm], and [I-D.ietf-opsawg-l2nm]) are augmented to include the TE and Service mapping parameters. Figure 2 shows the scope of the Augmented LxNM Model.

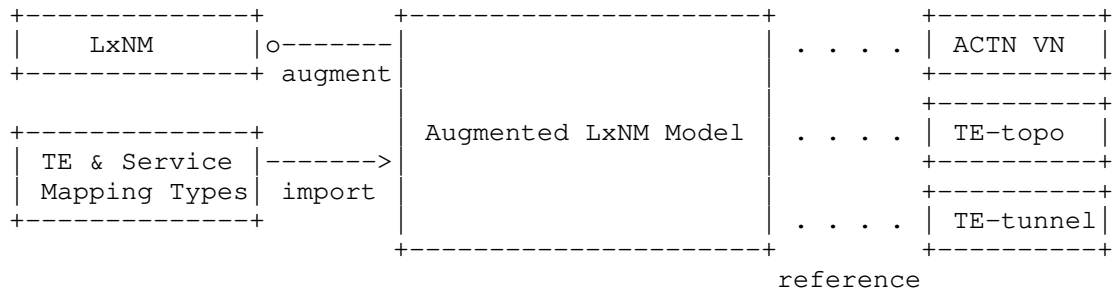
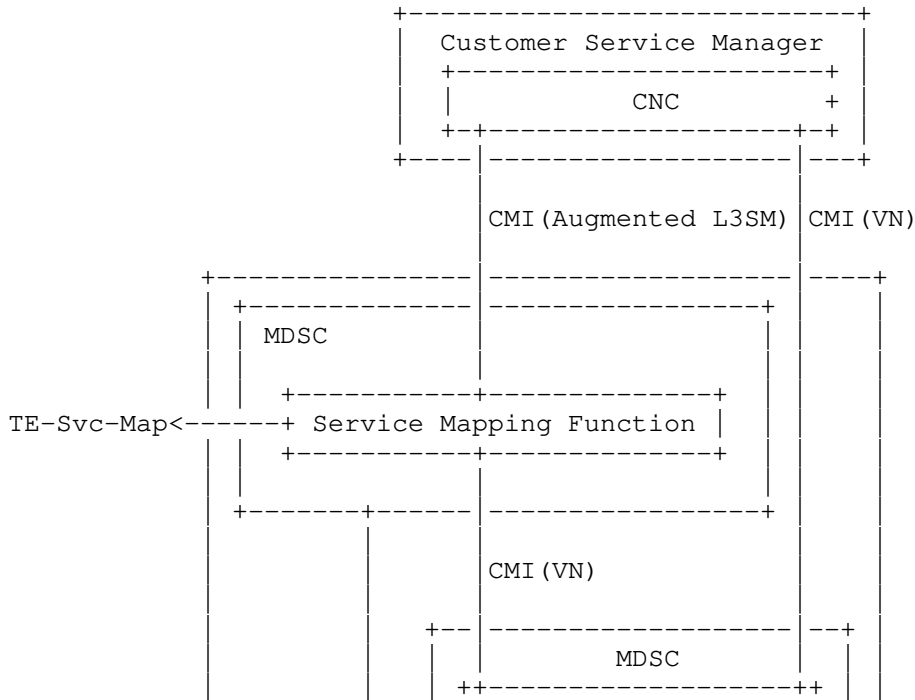


Figure 2: Augmented LxNM Model

The Augmented LxNM model (where x=2,3) augments the basic LxNM model while importing the common TE mapping related parameters (defined in Section 2) grouping information from TE and Service Mapping Types. The role of the augmented LxNM network model is to expose the mapping relationship between network models and TE models.

4. L3VPN Architecture in the ACTN Context

Figure 3 shows the architectural context of this document referencing the ACTN components and interfaces.



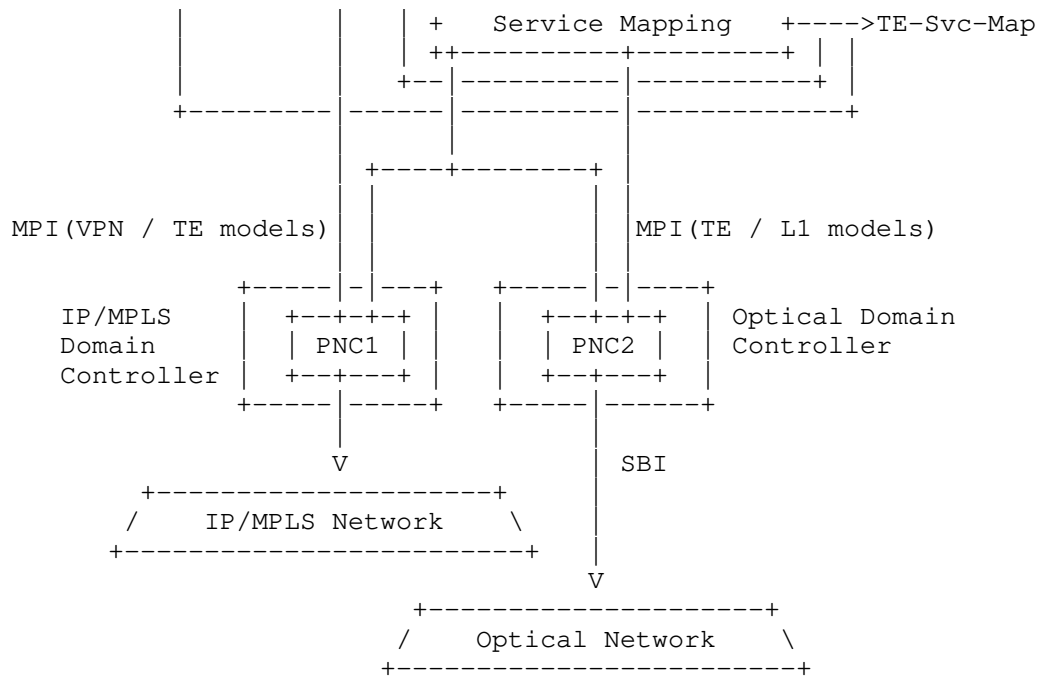


Figure 3: L3VPN Architecture from the IP+Optical Network Perspective

There are three main entities in the ACTN architecture and shown in Figure 3.

- o **CNC:** The Customer Network Controller is responsible for generating service requests. In the context of an L3VPN, the CNC uses the Augmented L3SM to express the service request and communicate it to the network operator.
- o **MDSC:** This entity is responsible for coordinating a L3VPN service request (expressed via the Augmented L3SM) with the IP/MPLS PNC and the Transport PNC. For TE services, one of the key responsibilities of the MDSC is to coordinate with both the IP PNC and the Transport PNC for the mapping of the Augmented L3VPN Service Model to the ACTN VN model. In the VN/TE-tunnel binding case, the MDSC will need to coordinate with the Transport PNC to dynamically create the TE-tunnels in the transport network as needed. These tunnels are added as links in the IP/MPLS Layer topology. The MDSC coordinates with IP/MPLS PNC to create the TE-tunnels in the IP/MPLS layer, as part of the ACTN VN creation.

- o PNC: The Provisioning Network Controller is responsible for configuring and operating the network devices. Figure 2 shows two distinct PNCs.
- \* IP/MPLS PNC (PNC1): This entity is responsible for device configuration to create PE-PE L3VPN tunnels for the VPN customer and for the configuration of the L3VPN VRF on the PE nodes. Each network element would select a tunnel based on the configuration.
- \* Transport PNC (PNC2): This entity is responsible for device configuration for TE tunnels in the transport networks.

There are four main interfaces shown in Figure 2.

- o CMI: The CNC-MDSC Interface is used to communicate service requests from the customer to the operator. The requests may be expressed as Augmented VPN service requests (L2SM, L3SM), as connectivity requests (L1CSM), or as virtual network requests (ACTN VN).
- o MPI: The MDSC-PNC Interface is used by the MDSC to orchestrate networks under the control of PNCs. The requests on this interface may use TE tunnel models, TE topology models, VPN network configuration models or layer one connectivity models.
- o SBI: The Southbound Interface is used by the PNC to control network devices and is out of scope for this document.

The TE Service Mapping Model as described in this document can be used to see the mapping between service models and VN models and TE Tunnel/Topology models. That mapping may occur in the CNC if a service request is mapped to a VN request. Or it may occur in the MDSC where a service request is mapped to a TE tunnel, TE topology, or VPN network configuration model. The TE Service Mapping Model may be read from the CNC or MDSC to understand how the mapping has been made and to see the purpose for which network resources are used.

As shown in Figure 2, the MDSC may be used recursively. For example, the CNC might map a L3SM request to a VN request that it sends to a recursive MDSC.

The high-level control flows for one example are as follows:

1. A customer asks for an L3VPN between CE1 and CE2 using the Augmented L3SM model.



2. The MDSC considers the service request and local policy to determine if it needs to create a new VN or any TE Topology, and if that is the case, ACTN VN YANG [I-D.ietf-teas-actn-vn-yang] is used to configure a new VN based on this VPN and map the VPN service to the ACTN VN. In case an existing tunnel is to be used, each device will select which tunnel to use and populate this mapping information.
3. The MDSC interacts with both the IP/MPLS PNC and the Transport PNC to create a PE-PE tunnel in the IP network mapped to a TE tunnel in the transport network by providing the inter-layer access points and tunnel requirements. The specific service information is passed to the IP/MPLS PNC for the actual VPN configuration and activation.
  - A. The Transport PNC creates the corresponding TE tunnel matching with the access point and egress point.
  - B. The IP/MPLS PNC maps the VPN ID with the corresponding TE tunnel ID to bind these two IDs.
4. The IP/MPLS PNC creates/updates a VRF instance for this VPN customer. This is not in the scope of this document.

#### 4.1. Service Mapping

Augmented L3SM and L2SM can be used to request VPN service creation including the creation of sites and corresponding site network access connection between CE and PE. A VPN-ID is used to identify each VPN service ordered by the customer. The ACTN VN can be used further to establish PE-to-PE connectivity between VPN sites belonging to the same VPN service. A VN-ID is used to identify each virtual network established between VPN sites.

Once the ACTN VN has been established over the TE network (maybe a new VN, maybe modification of an existing VN, or maybe the use of an unmodified existing VN), the mapping between the VPN service and the ACTN VN service can be created.

#### 4.2. Site Mapping

The elements in Augmented L3SM and L2SM define site location parameters and constraints such as distance and access diversity that can influence the placement of network attachment points (i.e, virtual network access points (VNAP)). To achieve this, a central directory can be set up to establish the mapping between location parameters and constraints and network attachment point location. Suppose multiple attachment points are matched, the management system

can use constraints or other local policy to select the best candidate network attachment points.

After a network attachment point is selected, the mapping between VPN site and VNAP can be established as shown in Table 1.

Site	Site Network Access	Location (Address, Postal Code, State, City, Country Code)	Access Diversity (Constraint-Type, Group-id, Target Group-id)	PE
SITE1	ACCESS1	(,,US,NewYork,)	(10,PE-Diverse,10)	PE1
SITE2	ACCESS2	(,,CN,Beijing,)	(10,PE-Diverse,10)	PE2
SITE3	ACCESS3	(,,UK,London, )	(12,same-PE,12)	PE4
SITE4	ACCESS4	(,,FR,Paris,)	(20,Bearer-Diverse,20)	PE7

Table 2: : Mapping Between VPN Site and VNAP

## 5. Applicability of TE-Service Mapping in Generic context

As discussed in the Introduction Section, the models presented in this document are also applicable generically outside of the ACTN architecture. [RFC8309] defines Customer Service Model between Customer and Service Orchestrator and Service Delivery Model between Service Orchestrator and Network Orchestrator(s). TE-Service mapping models defined in this document can be regarded primarily as Customer Service Model and secondarily as Service Deliver Model.

## 6. YANG Data Trees

### 6.1. Service Mapping Types

```

module: ietf-te-service-mapping-types
  +--rw te-mapping-templates
    +--rw te-mapping-template* [id]
      +--rw id                te-mapping-template-id
      +--rw description?     string
      +--rw map-type?        identityref
      +--rw path-constraints
        | +--rw te-bandwidth
        | | +--rw (technology)?
        | | +--:(generic)
  
```

```

|         +--rw generic?    te-bandwidth
+--rw link-protection?      identityref
+--rw setup-priority?       uint8
+--rw hold-priority?        uint8
+--rw signaling-type?       identityref
+--rw path-metric-bounds
|   +--rw path-metric-bound* [metric-type]
|   +--rw metric-type       identityref
|   +--rw upper-bound?     uint64
+--rw path-affinities-values
|   +--rw path-affinities-value* [usage]
|   +--rw usage             identityref
|   +--rw value?           admin-groups
+--rw path-affinity-names
|   +--rw path-affinity-name* [usage]
|   +--rw usage             identityref
|   +--rw affinity-name* [name]
|   +--rw name              string
+--rw path-srlgs-lists
|   +--rw path-srlgs-list* [usage]
|   +--rw usage             identityref
|   +--rw values*          srlg
+--rw path-srlgs-names
|   +--rw path-srlgs-name* [usage]
|   +--rw usage             identityref
|   +--rw names*           string
+--rw disjointness?        te-path-disjointness
+--rw optimizations
+--rw (algorithm)?
+--:(metric) {path-optimization-metric}?
|   +--rw optimization-metric* [metric-type]
|   |   +--rw metric-type
|   |   |   identityref
|   |   +--rw weight?                               uint8
|   |   +--rw explicit-route-exclude-objects
|   |   |   +--rw route-object-exclude-object* [index]
|   |   |   ...
|   |   +--rw explicit-route-include-objects
|   |   |   +--rw route-object-include-object* [index]
|   |   |   ...
|   +--rw tiebreakers
|   |   +--rw tiebreaker* [tiebreaker-type]
|   |   +--rw tiebreaker-type identityref
+--:(objective-function)
|   {path-optimization-objective-function}?
+--rw objective-function
+--rw objective-function-type? identityref

```

## 6.2. Service Models

## 6.2.1. L3SM

```

module: ietf-l3sm-te-service-mapping
augment /l3vpn-svc:l3vpn-svc/l3vpn-svc:vpn-services
  /l3vpn-svc:vpn-service:
  +--rw te-service-mapping!
    +--rw te-mapping
      +--rw map-type?                               identityref
      +--rw availability-type?                       identityref
      +--rw (te)?
        +--:(vn)
          | +--rw vn-ref?
          |   -> /vn:vn/vn-list/vn-id
        +--:(te-topo)
          | +--rw vn-topology-id?
          |   | te-types:te-topology-id
          | +--rw abstract-node?
          |   -> /nw:networks/network/node/node-id
        +--:(te-tunnel)
          | +--rw te-tunnel-list*                     te:tunnel-ref
          | +--rw sr-policy*
          |   [policy-color-ref policy-endpoint-ref]
          |   {sr-policy}?
          |   +--rw policy-color-ref                 leafref
          |   +--rw policy-endpoint-ref             leafref
        +--:(te-mapping-template) {template}?
          +--rw te-mapping-template-ref?           leafref
augment /l3vpn-svc:l3vpn-svc/l3vpn-svc:sites/l3vpn-svc:site
  /l3vpn-svc:site-network-accesses
  /l3vpn-svc:site-network-access:
  +--rw (te)?
    +--:(vn)
      | +--rw vn-ref?
      |   -> /vn:ap/access-point-list/access-point-id
    +--:(te)
      +--rw ltp?                                   te-types:te-tp-id

```

## 6.2.2. L2SM

```

module: ietf-l2sm-te-service-mapping
augment /l2vpn-svc:l2vpn-svc/l2vpn-svc:vpn-services
  /l2vpn-svc:vpn-service:
  +--rw te-service-mapping!
    +--rw te-mapping
      +--rw map-type? identityref
      +--rw availability-type? identityref
      +--rw (te)?
        +--:(vn)
          | +--rw vn-ref?
          |   -> /vn:vn/vn-list/vn-id
        +--:(te-topo)
          | +--rw vn-topology-id?
          |   | te-types:te-topology-id
          | +--rw abstract-node?
          |   -> /nw:networks/network/node/node-id
        +--:(te-tunnel)
          | +--rw te-tunnel-list* te:tunnel-ref
          | +--rw sr-policy*
          |   [policy-color-ref policy-endpoint-ref]
          |   {sr-policy}?
          |   +--rw policy-color-ref leafref
          |   +--rw policy-endpoint-ref leafref
        +--:(te-mapping-template) {template}?
          +--rw te-mapping-template-ref? leafref
augment /l2vpn-svc:l2vpn-svc/l2vpn-svc:sites/l2vpn-svc:site
  /l2vpn-svc:site-network-accesses
  /l2vpn-svc:site-network-access:
  +--rw (te)?
    +--:(vn)
      | +--rw vn-ref?
      |   -> /vn:ap/access-point-list/access-point-id
    +--:(te)
      +--rw ltp? te-types:te-tp-id

```

### 6.2.3. L1CSM

```

module: ietf-llcsm-te-service-mapping
augment /llcsm:ll-connectivity/llcsm:services/llcsm:service:
  +--rw te-service-mapping!
    +--rw te-mapping
      +--rw map-type?                identityref
      +--rw availability-type?       identityref
      +--rw (te)?
        +--:(vn)
          | +--rw vn-ref?
          |   -> /vn:vn/vn-list/vn-id
        +--:(te-topo)
          | +--rw vn-topology-id?
          | |   te-types:te-topology-id
          | +--rw abstract-node?
          |   -> /nw:networks/network/node/node-id
        +--:(te-tunnel)
          | +--rw te-tunnel-list*      te:tunnel-ref
          | +--rw sr-policy*
          |   [policy-color-ref policy-endpoint-ref]
          |   {sr-policy}?
          |   +--rw policy-color-ref    leafref
          |   +--rw policy-endpoint-ref leafref
          +--:(te-mapping-template) {template}?
            +--rw te-mapping-template-ref? leafref
augment /llcsm:ll-connectivity/llcsm:access/llcsm:unis/llcsm:uni:
  +--rw (te)?
    +--:(vn)
      | +--rw vn-ref?
      |   -> /vn:ap/access-point-list/access-point-id
    +--:(te)
      +--rw ltp?          te-types:te-tp-id

```

### 6.3. Network Models

#### 6.3.1. L3NM

```

module: ietf-l3nm-te-service-mapping
augment /l3vpn-ntw:l3vpn-ntw/l3vpn-ntw:vpn-services
  /l3vpn-ntw:vpn-service:
  +---rw te-service-mapping!
  +---rw te-mapping
  +---rw map-type?                identityref
  +---rw availability-type?       identityref
  +---rw (te)?
  +---:(vn)
  |   +---rw vn-ref?
  |       -> /vn:vn/vn-list/vn-id
  +---:(te-topo)
  |   +---rw vn-topology-id?
  |       |   te-types:te-topology-id
  |   +---rw abstract-node?
  |       -> /nw:networks/network/node/node-id
  +---:(te-tunnel)
  |   +---rw te-tunnel-list*      te:tunnel-ref
  |   +---rw sr-policy*
  |       [policy-color-ref policy-endpoint-ref]
  |       {sr-policy}?
  |       +---rw policy-color-ref    leafref
  |       +---rw policy-endpoint-ref leafref
  +---:(te-mapping-template) {template}?
  +---rw te-mapping-template-ref? leafref
augment /l3vpn-ntw:l3vpn-ntw/l3vpn-ntw:vpn-services
  /l3vpn-ntw:vpn-service/l3vpn-ntw:vpn-nodes
  /l3vpn-ntw:vpn-node/l3vpn-ntw:vpn-network-accesses
  /l3vpn-ntw:vpn-network-access:
  +---rw (te)?
  +---:(vn)
  |   +---rw vn-ref?
  |       -> /vn:ap/access-point-list/access-point-id
  +---:(te)
  +---rw ltp?          te-types:te-tp-id

```

### 6.3.2. L2NM

```

module: ietf-l2nm-te-service-mapping
augment /l2vpn-ntw:l2vpn-ntw/l2vpn-ntw:vpn-services
  /l2vpn-ntw:vpn-service:
  +--rw te-service-mapping!
    +--rw te-mapping
      +--rw map-type?                               identityref
      +--rw availability-type?                       identityref
      +--rw (te)?
        +--:(vn)
          | +--rw vn-ref?
          |   -> /vn:vn/vn-list/vn-id
        +--:(te-topo)
          | +--rw vn-topology-id?
          |   |   te-types:te-topology-id
          |   +--rw abstract-node?
          |     -> /nw:networks/network/node/node-id
        +--:(te-tunnel)
          | +--rw te-tunnel-list*                     te:tunnel-ref
          | +--rw sr-policy*
          |   [policy-color-ref policy-endpoint-ref]
          |   {sr-policy}?
          |   +--rw policy-color-ref                   leafref
          |   +--rw policy-endpoint-ref                 leafref
        +--:(te-mapping-template) {template}?
          +--rw te-mapping-template-ref?             leafref
augment /l2vpn-ntw:l2vpn-ntw/l2vpn-ntw:vpn-services
  /l2vpn-ntw:vpn-service/l2vpn-ntw:vpn-nodes
  /l2vpn-ntw:vpn-node/l2vpn-ntw:vpn-network-accesses
  /l2vpn-ntw:vpn-network-access:
  +--rw (te)?
    +--:(vn)
      | +--rw vn-ref?
      |   -> /vn:ap/access-point-list/access-point-id
    +--:(te)
      +--rw ltp?                                     te-types:te-tp-id

```

## 7. YANG Data Models

The YANG codes are as follows:

### 7.1. ietf-te-service-mapping-types

```

<CODE BEGINS> file "ietf-te-service-mapping-types@2020-07-13.yang"
module ietf-te-service-mapping-types {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-te-service-mapping-types";

```



```
prefix tsm-types;

/* Import inet-types */

import ietf-inet-types {
  prefix inet;
  reference
    "RFC 6991: Common YANG Data Types";
}

/* Import inet-types */

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

/* Import network model */

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

/* Import TE model */

import ietf-te {
  prefix te;
  reference
    "I-D.ietf-teas-yang-te: A YANG Data Model for Traffic
    Engineering Tunnels and Interfaces";
}

/* Import VN model */

import ietf-vn {
  prefix vn;
  reference
    "I-D.ietf-teas-actn-vn-yang: A Yang Data Model for VN Operation";
}

/* Import Routing */

import ietf-routing {
  prefix rt;
  reference
```

```
    "RFC 8349: A YANG Data Model for Routing Management";
}

/* Import SR Policy */

import ietf-sr-policy {
  prefix sr-policy;
  reference
    "I-D.raza-spring-sr-policy-yang: YANG Data Model for Segment
    Routing Policy";
}

organization
  "IETF Traffic Engineering Architecture and Signaling (TEAS)
  Working Group";
contact
  "WG Web:  <http://tools.ietf.org/wg/teas/>
  WG List:  <mailto:teas@ietf.org>

  Editor:   Young Lee
            <mailto:younglee.tx@gmail.com>
  Editor:   Dhruv Dhody
            <mailto:dhruv.ietf@gmail.com>
  Editor:   Qin Wu
            <mailto:bill.wu@huawei.com>";

description
  "This module contains a YANG module for TE & Service mapping
  parameters and policies as a common grouping applicable to
  various service models (e.g., L1CSM, L2SM, L3SM, etc.)

  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
  (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.

  The key words '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 BCP 14 (RFC 2119) (RFC 8174) when, and only when,
  they appear in all capitals, as shown here.";
```

```
revision 2020-07-13 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: Traffic Engineering and Service Mapping Yang Model";
}

/*
 * Features
 */

feature template {
  description
    "Support TE mapping templates.";
}

feature sr-policy {
  description
    "Support SR Policy.";
}

/*
 * Identity for map-type
 */

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

identity new {
  base map-type;
  description
    "The new VN/tunnels are binded to the service.";
}

identity hard-isolation {
  base new;
  description
    "Hard isolation.";
}

identity detnet-hard-isolation {
  base hard-isolation;
  description
    "Hard isolation with deterministic characteristics.";
}
```

```
identity soft-isolation {
  base new;
  description
    "Soft-isolation.";
}

identity select {
  base map-type;
  description
    "The VPN service selects an existing tunnel with no
    modification.";
}

identity modify {
  base map-type;
  description
    "The VPN service selects an existing tunnel and allows to modify
    the properties of the tunnel (e.g., b/w)";
}

identity template {
  base map-type;
  description
    "The VPN service selects an TE mapping template with path
    constraints and optimization criteria";
}

/*
 * 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 te-mapping-template-id {
  type inet:uri;
  description
    "Identifier for a TE mapping template. The precise
     structure of the te-mapping-template-id will be up
     to the implementation. The identifier SHOULD be
     chosen such that the same template will always be
     identified through the same identifier, even if the
     data model is instantiated in separate datastores.";
}

/*
 * Groupings
 */

grouping te-ref {
  description
    "The reference to TE.";
  choice te {
    description
      "The TE";
    case vn {
      leaf vn-ref {
        type leafref {
          path "/vn:vn/vn:vn-list/vn:vn-id";
        }
      }
    }
  }
}
```

```
        description
            "The reference to VN";
        reference
            "RFC 8453: Framework for Abstraction and Control of TE
            Networks (ACTN)";
    }
}
case te-topo {
    leaf vn-topology-id {
        type te-types:te-topology-id;
        description
            "An identifier to the TE Topology Model where the abstract
            nodes and links of the Topology can be found for Type 2
            VNS";
        reference
            "I-D.ietf-teas-yang-te-topo: YANG Data Model for Traffic
            Engineering (TE) Topologies";
    }
    leaf abstract-node {
        type leafref {
            path "/nw:networks/nw:network/nw:node/nw:node-id";
        }
        description
            "A reference to the abstract node in TE Topology";
        reference
            "I-D.ietf-teas-yang-te-topo: YANG Data Model for Traffic
            Engineering (TE) Topologies";
    }
}
case te-tunnel {
    leaf-list te-tunnel-list {
        type te:tunnel-ref;
        description
            "Reference to TE Tunnels";
        reference
            "I-D.ietf-teas-yang-te: A YANG Data Model for Traffic
            Engineering Tunnels and Interfaces";
    }
    list sr-policy {
        if-feature "sr-policy";
        key "policy-color-ref policy-endpoint-ref";
        description
            "SR Policy";
        leaf policy-color-ref {
            type leafref {
                path
                    "/rt:routing/sr-policy:segment-routing"
                    + "/sr-policy:traffic-engineering/sr-policy:policies"
            }
        }
    }
}
```

```
        + "/sr-policy:policy/sr-policy:color";
    }
    description
        "Reference to sr-policy color";
}
leaf policy-endpoint-ref {
    type leafref {
        path
            "/rt:routing/sr-policy:segment-routing"
            + "/sr-policy:traffic-engineering/sr-policy:policies"
            + "/sr-policy:policy/sr-policy:endpoint";
    }
    description
        "Reference to sr-policy endpoint";
}
}
}
case te-mapping-template {
    if-feature "template";
    leaf te-mapping-template-ref {
        type leafref {
            path "/te-mapping-templates/te-mapping-template/id";
        }
        description
            "An identifier to the TE Mapping Template where the TE
            constraints and optimization criteria are specified.";
    }
}
}
}
}
//grouping

grouping te-endpoint-ref {
    description
        "The reference to TE endpoints.";
    choice te {
        description
            "The TE";
        case vn {
            leaf vn-ref {
                type leafref {
                    path "/vn:ap/vn:access-point-list/vn:access-point-id";
                }
                description
                    "The reference to VN AP";
                reference
                    "RFC 8453: Framework for Abstraction and Control of TE
```

```
        Networks (ACTN)";
    }
}
case te {
  leaf ltp {
    type te-types:te-tp-id;
    description
      "Reference LTP in the TE-topology";
    reference
      "I-D.ietf-teas-yang-te-topo: YANG Data Model for Traffic
      Engineering (TE) Topologies";
  }
}
}
}
//grouping

grouping te-mapping {
  description
    "Mapping between Services and TE";
  container te-mapping {
    description
      "Mapping between Services and TE";
    leaf map-type {
      type identityref {
        base map-type;
      }
      description
        "Isolation Requirements, Tunnel Bind or
        Tunnel Selection";
    }
    leaf availability-type {
      type identityref {
        base availability-type;
      }
      description
        "Availability Requirement for the Service";
    }
    uses te-ref;
  }
}
//grouping

container te-mapping-templates {
  description
    "The TE constraints and optimization criteria";
```



```
list te-mapping-template {
  key "id";
  leaf id {
    type te-mapping-template-id;
    description
      "Identification of the Template to be used.";
  }
  leaf description {
    type string;
    description
      "Description of the template.";
  }
  leaf map-type {
    type identityref {
      base map-type;
    }
    must '(. != "template")' {
      error-message "The map-type must be other than "
        + "TE mapping template";
    }
    description
      "Map type for the VN/Tunnel creation/
        selection.";
  }
  uses te-types:generic-path-constraints;
  uses te-types:generic-path-optimization;
  description
    "List for templates.";
}
}
```

<CODE ENDS>

## 7.2. Service Models

### 7.2.1. ietf-l3sm-te-service-mapping

```
<CODE BEGINS> file "ietf-l3sm-te-service-mapping@2020-07-13.yang"
module ietf-l3sm-te-service-mapping {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-l3sm-te-service-mapping";
  prefix l3-tsm;

  import ietf-te-service-mapping-types {
    prefix tsm-types;
  }
}
```

```
reference
  "RFC XXXX: Traffic Engineering and Service Mapping Yang Model";
}
import ietf-l3vpn-svc {
  prefix l3vpn-svc;
  reference
    "RFC 8299: YANG Data Model for L3VPN Service Delivery";
}

organization
  "IETF Traffic Engineering Architecture and Signaling (TEAS)
  Working Group";
contact
  "WG Web: <http://tools.ietf.org/wg/teas/>
  WG List: <mailto:teas@ietf.org>

  Editor: Young Lee
         <mailto:younglee.tx@gmail.com>
  Editor: Dhruv Dhody
         <mailto:dhruv.ietf@gmail.com>
  Editor: Qin Wu
         <mailto:bill.wu@huawei.com>";
description
  "This module contains a YANG module for the mapping of Layer 3
  Service Model (L3SM) to the TE and VN.

  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
  (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.

  The key words '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 BCP 14 (RFC 2119) (RFC 8174) when, and only when,
  they appear in all capitals, as shown here.";

revision 2020-03-08 {
  description
    "Initial revision.";
```

```

    reference
      "RFC XXXX: Traffic Engineering and Service Mapping Yang Model";
  }

  /*
  * Augmentation to L3SM
  */

  augment "/l3vpn-svc:l3vpn-svc/l3vpn-svc:vpn-services"
    + "/l3vpn-svc:vpn-service" {
    description
      "L3SM augmented to include TE parameters and mapping";
    container te-service-mapping {
      presence "Indicates L3 service to TE mapping";
      description
        "Container to augment l3sm to TE parameters and mapping";
      uses tsm-types:te-mapping;
    }
  }

  //augment

  augment "/l3vpn-svc:l3vpn-svc/l3vpn-svc:sites/l3vpn-svc:site"
    + "/l3vpn-svc:site-network-accesses"
    + "/l3vpn-svc:site-network-access" {
    description
      "This augment is only valid for TE mapping of L3SM network-access
      to TE endpoints";
    uses tsm-types:te-endpoint-ref;
  }

  //augment
}

<CODE ENDS>

```

### 7.2.2. ietf-l2sm-te-service-mapping

```

<CODE BEGINS> file "ietf-l2sm-te-service-mapping@2020-07-13.yang"
module ietf-l2sm-te-service-mapping {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-l2sm-te-service-mapping";
  prefix l2-tsm;

  import ietf-te-service-mapping-types {
    prefix tsm-types;
    reference

```

```
    "RFC XXXX: Traffic Engineering and Service Mapping Yang Model";
}
import ietf-l2vpn-svc {
  prefix l2vpn-svc;
  reference
    "RFC 8466: A YANG Data Model for Layer 2 Virtual Private Network
      (L2VPN) Service Delivery";
}

organization
  "IETF Traffic Engineering Architecture and Signaling (TEAS)
  Working Group";
contact
  "WG Web: <http://tools.ietf.org/wg/teas/>
  WG List: <mailto:teas@ietf.org>

  Editor: Young Lee
          <mailto:younglee.tx@gmail.com>
  Editor: Dhruv Dhody
          <mailto:dhruv.ietf@gmail.com>
  Editor: Qin Wu
          <mailto:bill.wu@huawei.com>";
description
  "This module contains a YANG module for the mapping of Layer 2
  Service Model (L2SM) to the TE and VN.

  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
  (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.

  The key words '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 BCP 14 (RFC 2119) (RFC 8174) when, and only when,
  they appear in all capitals, as shown here.";

revision 2020-07-13 {
  description
    "Initial revision.";
```

```
    reference
      "RFC XXXX: Traffic Engineering and Service Mapping Yang Model";
  }
/*
 * Augmentation to L2SM
 */
augment "/l2vpn-svc:l2vpn-svc/l2vpn-svc:vpn-services/"
  + "l2vpn-svc:vpn-service" {
  description
    "L2SM augmented to include TE parameters and mapping";
  container te-service-mapping {
    presence "indicates L2 service to te mapping";
    description
      "Container to augment L2SM to TE parameters and mapping";
    uses tsm-types:te-mapping;
  }
}
//augment

augment "/l2vpn-svc:l2vpn-svc/l2vpn-svc:sites/l2vpn-svc:site"
  + "l2vpn-svc:site-network-accesses"
  + "l2vpn-svc:site-network-access" {
  description
    "This augment is only valid for TE mapping of L2SM network-access
    to TE endpoints";
  uses tsm-types:te-endpoint-ref;
}
//augment
}
```

<CODE ENDS>

### 7.2.3. ietf-llcsm-te-service-mapping

```
<CODE BEGINS> file "ietf-llcsm-te-service-mapping@2020-07-13.yang"
module ietf-llcsm-te-service-mapping {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-llcsm-te-service-mapping";
  prefix ll-tsm;

  import ietf-te-service-mapping-types {
    prefix tsm-types;
    reference

```

```
    "RFC XXXX: Traffic Engineering and Service Mapping Yang Model";
}
import ietf-llcsm {
  prefix llcsm;
  reference
    "I-D.ietf-ccamp-llcsm-yang: A YANG Data Model for L1 Connectivity
    Service Model (L1CSM)";
}

organization
  "IETF Traffic Engineering Architecture and Signaling (TEAS)
  Working Group";
contact
  "WG Web: <http://tools.ietf.org/wg/teas/>
  WG List: <mailto:teas@ietf.org>

  Editor: Young Lee
         <mailto:younglee.tx@gmail.com>
  Editor: Dhruv Dhody
         <mailto:dhruv.ietf@gmail.com>
  Editor: Qin Wu
         <mailto:bill.wu@huawei.com>";
description
  "This module contains a YANG module for the mapping of
  Layer 1 Connectivity Service Module (L1CSM) to the TE and VN

  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
  (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.

  The key words '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 BCP 14 (RFC 2119) (RFC 8174) when, and only when,
  they appear in all capitals, as shown here.";

revision 2020-07-13 {
  description
    "Initial revision.";
```

```
    reference
      "RFC XXXX: Traffic Engineering and Service Mapping Yang Model";
  }
/*
 * Augmentation to L1CSM
 */
augment "/l1csm:l1-connectivity/l1csm:services/l1csm:service" {
  description
    "L1CSM augmented to include TE parameters and mapping";
  container te-service-mapping {
    presence "Indicates L1 service to TE mapping";
    description
      "Container to augment L1CSM to TE parameters and mapping";
    uses tsm-types:te-mapping;
  }
}
//augment

augment "/l1csm:l1-connectivity/l1csm:access/l1csm:unis/"
  + "l1csm:uni" {
  description
    "This augment is only valid for TE mapping of L1CSM UNI to TE
    endpoints";
  uses tsm-types:te-endpoint-ref;
}
//augment
}

<CODE ENDS>
```

### 7.3. Network Models

#### 7.3.1. ietf-l3nm-te-service-mapping

```
<CODE BEGINS> file "ietf-l3nm-te-service-mapping@2020-07-13.yang"
module ietf-l3nm-te-service-mapping {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-l3nm-te-service-mapping";
  prefix l3nm-tsm;

  import ietf-te-service-mapping-types {
    prefix tsm-types;
    reference
```

```
    "RFC XXXX: Traffic Engineering and Service Mapping Yang Model";
}
import ietf-l3vpn-ntw {
  prefix l3vpn-ntw;
  reference
    "I-D.ietf-opsawg-l3sm-l3nm: A Layer 3 VPN Network YANG Model";
}

organization
  "IETF Traffic Engineering Architecture and Signaling (TEAS)
  Working Group";
contact
  "WG Web: <http://tools.ietf.org/wg/teas/>
  WG List: <mailto:teas@ietf.org>

  Editor: Young Lee
         <mailto:younglee.tx@gmail.com>
  Editor: Dhruv Dhody
         <mailto:dhruv.ietf@gmail.com>
  Editor: Qin Wu
         <mailto:bill.wu@huawei.com>";
description
  "This module contains a YANG module for the mapping of Layer 3
  Network Model (L3NM) to the TE and VN.

  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
  (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.

  The key words '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 BCP 14 (RFC 2119) (RFC 8174) when, and only when,
  they appear in all capitals, as shown here.";

revision 2020-07-13 {
  description
    "Initial revision.";
  reference
```



```

    "RFC XXXX: Traffic Engineering and Service Mapping Yang Model";
}

/*
 * Augmentation to L3NM
 */

augment "/l3vpn-ntw:l3vpn-ntw/l3vpn-ntw:vpn-services"
  + "/l3vpn-ntw:vpn-service" {
  description
    "L3SM augmented to include TE parameters and mapping";
  container te-service-mapping {
    presence "Indicates L3 network to TE mapping";
    description
      "Container to augment l3nm to TE parameters and mapping";
    uses tsm-types:te-mapping;
  }
}

//augment

augment "/l3vpn-ntw:l3vpn-ntw/l3vpn-ntw:vpn-services"
  + "/l3vpn-ntw:vpn-service"
  + "/l3vpn-ntw:vpn-nodes/l3vpn-ntw:vpn-node"
  + "/l3vpn-ntw:vpn-network-accesses"
  + "/l3vpn-ntw:vpn-network-access" {
  description
    "This augment is only valid for TE mapping of L3NM network-access
    to TE endpoints";
  uses tsm-types:te-endpoint-ref;
}

//augment
}

<CODE ENDS>

```

### 7.3.2. ietf-l2nm-te-service-mapping

```

<CODE BEGINS> file "ietf-l2nm-te-service-mapping@2020-07-13.yang"
module ietf-l2nm-te-service-mapping {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-l2nm-te-service-mapping";
  prefix l2nm-tsm;

  import ietf-te-service-mapping-types {
    prefix tsm-types;
  }
}

```

```
reference
  "RFC XXXX: Traffic Engineering and Service Mapping Yang Model";
}
import ietf-l2vpn-ntw {
  prefix l2vpn-ntw;
  reference
    "I-D.ietf-l2nm: A Layer 2 VPN Network YANG Model";
}

organization
  "IETF Traffic Engineering Architecture and Signaling (TEAS)
  Working Group";
contact
  "WG Web: <http://tools.ietf.org/wg/teas/>
  WG List: <mailto:teas@ietf.org>

  Editor: Young Lee
          <mailto:younglee.tx@gmail.com>
  Editor: Dhruv Dhody
          <mailto:dhruv.ietf@gmail.com>
  Editor: Qin Wu
          <mailto:bill.wu@huawei.com>";
description
  "This module contains a YANG module for the mapping of Layer 2
  Network Model (L2NM) to the TE and VN.

  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
  (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.

  The key words '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 BCP 14 (RFC 2119) (RFC 8174) when, and only when,
  they appear in all capitals, as shown here.";

revision 2020-07-13 {
  description
    "Initial revision.";
```

```

    reference
      "RFC XXXX: Traffic Engineering and Service Mapping Yang Model";
  }

/*
 * Augmentation to L2NM
 */

augment "/l2vpn-ntw:l2vpn-ntw/l2vpn-ntw:vpn-services"
  + "/l2vpn-ntw:vpn-service" {
  description
    "L2SM augmented to include TE parameters and mapping";
  container te-service-mapping {
    presence "Indicates L2 network to TE mapping";
    description
      "Container to augment l2nm to TE parameters and mapping";
    uses tsm-types:te-mapping;
  }
}

//augment

augment "/l2vpn-ntw:l2vpn-ntw/l2vpn-ntw:vpn-services"
  + "/l2vpn-ntw:vpn-service"
  + "/l2vpn-ntw:vpn-nodes/l2vpn-ntw:vpn-node"
  + "/l2vpn-ntw:vpn-network-accesses"
  + "/l2vpn-ntw:vpn-network-access" {
  description
    "This augment is only valid for TE mapping of L2NM network-access
    to TE endpoints";
  uses tsm-types:te-endpoint-ref;
}

//augment
}

<CODE ENDS>

```

## 8. Security Considerations

The YANG modules defined in this document is designed to be accessed via network management protocol such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is 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 pre-configured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in the YANG modules which 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. These are the subtrees and data nodes and their sensitivity/vulnerability:

- o /l3vpn-svc/vpn-services/vpn-service/te-service-mapping/te-mapping/  
- configure TE Service mapping.
- o /l3vpn-svc/sites/site/site-network-accesses/site-network-access/  
te/ - configure TE Endpoint mapping.
- o /l2vpn-svc/vpn-services/vpn-service/te-service-mapping/te-mapping/  
- configure TE Service mapping.
- o /l2vpn-svc/sites/site/site-network-accesses/site-network-access/  
te/ - configure TE Endpoint mapping.
- o /l1-connectivity/services/service/te-service-mapping/te-mapping/ -  
configure TE Service mapping.
- o /l1-connectivity/access/unis/uni/te/ - configure TE Endpoint  
mapping.
- o /l3vpn-ntw/vpn-services/vpn-service/te-service-mapping/te-mapping/  
- configure TE Network mapping.
- o /l3vpn-ntw/vpn-services/vpn-service/vpn-nodes/vpn-node/vpn-  
network-accesses/vpn-network-access/te/ - configure TE Endpoint  
mapping.
- o /l2vpn-ntw/vpn-services/vpn-service/te-service-mapping/te-mapping/  
- configure TE Network mapping.
- o /l2vpn-ntw/vpn-services/vpn-service/vpn-nodes/vpn-node/vpn-  
network-accesses/vpn-network-access/te/ - configure TE Endpoint  
mapping.

Unauthorized access to above list can adversely affect the VPN service.

Some of the readable data nodes in the YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. The TE related parameters attached to the VPN service can leak sensitive information about the network. This is applicable to all elements in the yang models defined in this document.

This document has no RPC defined.

## 9. IANA Considerations

This document request the IANA to register four URIs in the "IETF XML Registry" [RFC3688]. Following the format in RFC 3688, the following registrations are requested -

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

URI: urn:ietf:params:xml:ns:yang:ietf-l3sm-te-service-mapping  
Registrant Contact: The IESG.  
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-l2sm-te-service-mapping  
Registrant Contact: The IESG.  
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-l1csm-te-service-mapping  
Registrant Contact: The IESG.  
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-l3nm-te-service-mapping  
Registrant Contact: The IESG.  
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-l2nm-te-service-mapping  
Registrant Contact: The IESG.  
XML: N/A, the requested URI is an XML namespace.

This document request the IANA to register four YANG modules in the "YANG Module Names" registry [RFC6020], as follows -

Name: ietf-te-service-mapping-types  
Namespace: urn:ietf:params:xml:ns:yang:ietf-te-service-mapping-types  
Prefix: tsm-types  
Reference: [This.I-D]

Name: ietf-l3sm-te-service-mapping  
Namespace: urn:ietf:params:xml:ns:yang:ietf-l3sm-te-service-mapping  
Prefix: l3-tsm  
Reference: [This.I-D]

Name: ietf-l2sm-te-service-mapping  
Namespace: urn:ietf:params:xml:ns:yang:ietf-l2sm-te-service-mapping  
Prefix: l2-tsm  
Reference: [This.I-D]

Name: ietf-l1csm-te-service-mapping  
Namespace: urn:ietf:params:xml:ns:yang:ietf-l1csm-te-service-mapping  
Prefix: l1-tsm  
Reference: [This.I-D]

Name: ietf-l3nm-te-service-mapping  
Namespace: urn:ietf:params:xml:ns:yang:ietf-l3nm-te-service-mapping  
Prefix: l3nm-tsm  
Reference: [This.I-D]

Name: ietf-l2nm-te-service-mapping  
Namespace: urn:ietf:params:xml:ns:yang:ietf-l2nm-te-service-mapping  
Prefix: l2nm-tsm  
Reference: [This.I-D]

## 10. Acknowledgements

We thank Diego Caviglia, and Igor Bryskin for useful discussions and motivation for this work.

## 11. References

### 11.1. Normative References

[I-D.ietf-ccamp-l1csm-yang]  
Lee, Y., Lee, K., Zheng, H., Dhody, D., Dios, O., and D. Ceccarelli, "A YANG Data Model for L1 Connectivity Service Model (L1CSM)", draft-ietf-ccamp-l1csm-yang-11 (work in progress), March 2020.

- [I-D.ietf-opsawg-l2nm]  
Barguil, S., Dios, O., Boucadair, M., Munoz, L., Jalil, L., and J. Ma, "A Layer 2 VPN Network YANG Model", draft-ietf-opsawg-l2nm-00 (work in progress), July 2020.
- [I-D.ietf-opsawg-l3sm-l3nm]  
Barguil, S., Dios, O., Boucadair, M., Munoz, L., and A. Aguado, "A Layer 3 VPN Network YANG Model", draft-ietf-opsawg-l3sm-l3nm-03 (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.ietf-teas-yang-te]  
Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te-23 (work in progress), March 2020.
- [I-D.ietf-teas-yang-te-topo]  
Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", draft-ietf-teas-yang-te-topo-22 (work in progress), June 2019.
- [I-D.raza-spring-sr-policy-yang]  
Raza, K., Sawaya, R., Shunwan, Z., Voyer, D., Durrani, M., Matsushima, S., and V. Beeram, "YANG Data Model for Segment Routing Policy", draft-raza-spring-sr-policy-yang-02 (work in progress), November 2019.
- [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>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.

- [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>>.
- [RFC7926] Farrel, A., Ed., Drake, J., Bitar, N., Swallow, G., Ceccarelli, D., and X. Zhang, "Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks", BCP 206, RFC 7926, DOI 10.17487/RFC7926, July 2016, <<https://www.rfc-editor.org/info/rfc7926>>.
- [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>>.
- [RFC8299] Wu, Q., Ed., Litkowski, S., Tomotaki, L., and K. Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC 8299, DOI 10.17487/RFC8299, January 2018, <<https://www.rfc-editor.org/info/rfc8299>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.



- [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <<https://www.rfc-editor.org/info/rfc8345>>.
- [RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <<https://www.rfc-editor.org/info/rfc8349>>.
- [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>>.
- [RFC8466] Wen, B., Fioccola, G., Ed., Xie, C., and L. Jalil, "A YANG Data Model for Layer 2 Virtual Private Network (L2VPN) Service Delivery", RFC 8466, DOI 10.17487/RFC8466, October 2018, <<https://www.rfc-editor.org/info/rfc8466>>.
- [RFC8776] Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Common YANG Data Types for Traffic Engineering", RFC 8776, DOI 10.17487/RFC8776, June 2020, <<https://www.rfc-editor.org/info/rfc8776>>.

## 11.2. Informative References

- [I-D.ietf-teas-actn-yang]  
Lee, Y., Zheng, H., Ceccarelli, D., Yoon, B., Dios, O., Shin, J., and S. Belotti, "Applicability of YANG models for Abstraction and Control of Traffic Engineered Networks", draft-ietf-teas-actn-yang-05 (work in progress), February 2020.
- [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>>.
- [RFC8199] Bogdanovic, D., Claise, B., and C. Moberg, "YANG Module Classification", RFC 8199, DOI 10.17487/RFC8199, July 2017, <<https://www.rfc-editor.org/info/rfc8199>>.
- [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>>.

[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>>.

#### Appendix A. Contributor Addresses

Adrian Farrel  
Old Dog Consulting

Email: [adrian@olddog.co.uk](mailto:adrian@olddog.co.uk)

Italo Busi  
Huawei Technologies

Email: [Italo.Busi@huawei.com](mailto:Italo.Busi@huawei.com)

Haomian Zheng  
Huawei Technologies

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

Anton Snitser  
Sedonasys

Email: [antons@sedonasys.com](mailto:antons@sedonasys.com)

SAMIER BARGUIL GIRALDO  
Telefonica

Email: [samier.barguilgiraldo.ext@telefonica.com](mailto:samier.barguilgiraldo.ext@telefonica.com)

Oscar Gonzalez de Dios  
Telefonica

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

Carlo Perocchio  
Ericsson

Email: [carlo.perocchio@ericsson.com](mailto:carlo.perocchio@ericsson.com)

#### Authors' Addresses

Young Lee (editor)  
Samsung Electronics

Email: [younglee.tx@gmail.com](mailto:younglee.tx@gmail.com)

Dhruv Dhody (editor)  
Huawei Technologies

Email: dhruv.ietf@gmail.com

Giuseppe Fioccola  
Huawei Technologies

Email: giuseppe.fioccola@huawei.com

Qin Wu (editor)  
Huawei Technologies

Email: bill.wu@huawei.com

Daniele Ceccarelli  
Ericsson  
Torshamnsgatan, 48  
Stockholm, Sweden

Email: daniele.ceccarelli@ericsson.com

Jeff Tantsura  
Apstra

Email: jefftant.ietf@gmail.com

TEAS Working Group  
Internet Draft  
Intended status: Standard Track  
Expires: January 2021

Italo Busi (Ed.)  
Huawei  
Sergio Belotti (Ed.)  
Nokia  
Victor Lopez  
Telefonica  
Anurag Sharma  
Google  
Yan Shi  
China Unicom

July 11, 2020

Yang model for requesting Path Computation  
draft-ietf-teas-yang-path-computation-10.txt

#### 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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

This Internet-Draft will expire on January 11, 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 (<http://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.

## Abstract

There are scenarios, typically in a hierarchical SDN context, where the topology information provided by a TE network provider may not be sufficient for its client to perform end-to-end path computation. In these cases the client would need to request the provider to calculate some (partial) feasible paths.

This document defines a YANG data model for an RPC to request path computation. This model complements the solution, defined in RFCXXXX, to configure a TE Tunnel path in "compute-only" mode.

[RFC EDITOR NOTE: Please replace RFC XXXX with the RFC number of draft-ietf-teas-yang-te once it has been published.

Moreover this document describes some use cases where a path computation request, via YANG-based protocols (e.g., NETCONF or RESTCONF), can be needed.

## Table of Contents

1. Introduction.....	3
1.1. Terminology.....	5
1.2. Tree Diagram.....	5
1.3. Prefixes in Data Node Names.....	5
2. Use Cases.....	6
2.1. Packet/Optical Integration.....	6
2.2. Multi-domain TE Networks.....	11
2.3. Data center interconnections.....	13

2.4. Backward Recursive Path Computation scenario.....	15
2.5. Hierarchical PCE scenario.....	16
3. Motivations.....	18
3.1. Motivation for a YANG Model.....	18
3.1.1. Benefits of common data models.....	18
3.1.2. Benefits of a single interface.....	19
3.1.3. Extensibility.....	20
3.2. Interactions with TE Topology.....	20
3.2.1. TE Topology Aggregation.....	21
3.2.2. TE Topology Abstraction.....	24
3.2.3. Complementary use of TE topology and path computation.....	25
3.3. Path Computation RPC.....	28
3.3.1. Temporary reporting of the computed path state.....	30
4. Path Computation and Optimization for multiple paths.....	32
5. YANG Model for requesting Path Computation.....	33
5.1. Synchronization of multiple path computation requests....	34
5.2. Returned metric values.....	37
5.3. Multiple Paths Requests for the same TE Tunnel.....	38
5.4. Multi-Layer Path Computation.....	42
6. YANG model for TE path computation.....	43
6.1. YANG Tree.....	43
6.2. YANG Module.....	57
7. Security Considerations.....	82
8. IANA Considerations.....	83
9. References.....	84
9.1. Normative References.....	84
9.2. Informative References.....	85
Appendix A. Examples of dimensioning the "detailed connectivity matrix".....	87
Acknowledgments.....	93
Contributors.....	93
Authors' Addresses.....	94

## 1. Introduction

There are scenarios, typically in a hierarchical SDN context, where the topology information provided by a TE network provider may not be sufficient for its client to perform end-to-end path computation. In these cases the client would need to request the provider to calculate some (partial) feasible paths, complementing his topology knowledge, to make his end-to-end path computation feasible.

This type of scenarios can be applied to different interfaces in different reference architectures:

- o ABNO control interface [RFC7491], in which an Application Service Coordinator can request ABNO controller to take in charge path calculation (see Figure 1 in [RFC7491]).
- o ACTN [RFC8453], where a controller hierarchy is defined, the need for path computation arises on both interfaces CMI (interface between Customer Network Controller (CNC) and Multi Domain Service Coordinator (MDSC)) and/or MPI (interface between MSDC-PNC). [RFC8454] describes an information model for the Path Computation request.

Multiple protocol solutions can be used for communication between different controller hierarchical levels. This document assumes that the controllers are communicating using YANG-based protocols (e.g., NETCONF or RESTCONF).

Path Computation Elements, Controllers and Orchestrators perform their operations based on Traffic Engineering Databases (TED). Such TEDs can be described, in a technology agnostic way, with the YANG Data Model for TE Topologies [TE-TOPO]. Furthermore, the technology specific details of the TED are modeled in the augmented TE topology models (e.g. [OTN-TOPO] for OTN ODU technologies).

The availability of such topology models allows providing the TED using YANG-based protocols (e.g., NETCONF or RESTCONF). Furthermore, it enables a PCE/Controller performing the necessary abstractions or modifications and offering this customized topology to another PCE/Controller or high level orchestrator.

The tunnels that can be provided over the networks described with the topology models can be also set-up, deleted and modified via YANG-based protocols (e.g., NETCONF or RESTCONF) using the TE-Tunnel Yang model [TE-TUNNEL].

This document defines a YANG model for an RPC to request path computation, which complements the solution defined in [TE-TUNNEL], to configure a TE Tunnel path in "compute-only" mode.

The YANG model definition does not make any assumption about whether that the client or the server implement a "PCE" functionality, as defined in [RFC4655], and the PCEP protocol, as defined in [RFC5440].

Moreover, this document describes some use cases where a path computation request, via YANG-based protocols (e.g., NETCONF or RESTCONF), can be needed.

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

### 1.1. Terminology

**TED:** The traffic engineering database is a collection of all TE information about all TE nodes and TE links in a given network.

**PCE:** A Path Computation Element (PCE) is an entity that is capable of computing a network path or route based on a network graph, and of applying computational constraints during the computation. The PCE entity is an application that can be located within a network node or component, on an out-of-network server, etc. For example, a PCE would be able to compute the path of a TE LSP by operating on the TED and considering bandwidth and other constraints applicable to the TE LSP service request. [RFC4655].

The terminology for describing YANG data models is found in [RFC7950].

### 1.2. Tree Diagram

A simplified graphical representation of the data model is used in section 6.1 of this this document. The meaning of the symbols in these diagrams is defined in [RFC8340].

### 1.3. Prefixes in Data Node Names

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



Prefix	YANG module	Reference
inet	ietf-inet-types	[RFC6991]
te-types	ietf-te-types	[RFC8776]
te	ietf-te	[TE-TUNNEL]
te-pc	ietf-te-path-computation	this document

Table 1: Prefixes and corresponding YANG modules

## 2. Use Cases

This section presents some use cases, where a client needs to request underlying SDN controllers for path computation.

The use of the YANG model defined in this document is not restricted to these use cases but can be used in any other use case when deemed useful.

The presented uses cases have been grouped, depending on the different underlying topologies: a) Packet-Optical integration; b) Multi-domain Traffic Engineered (TE) Networks; and c) Data center interconnections. Use cases d) and e) respectively present how to apply this Yang model for standard multi-domain PCE i.e. Backward Recursive Path Computation [RFC5441] and Hierarchical PCE [RFC6805].

### 2.1. Packet/Optical Integration

In this use case, an Optical network is used to provide connectivity to some nodes of a Packet network (see Figure 1).

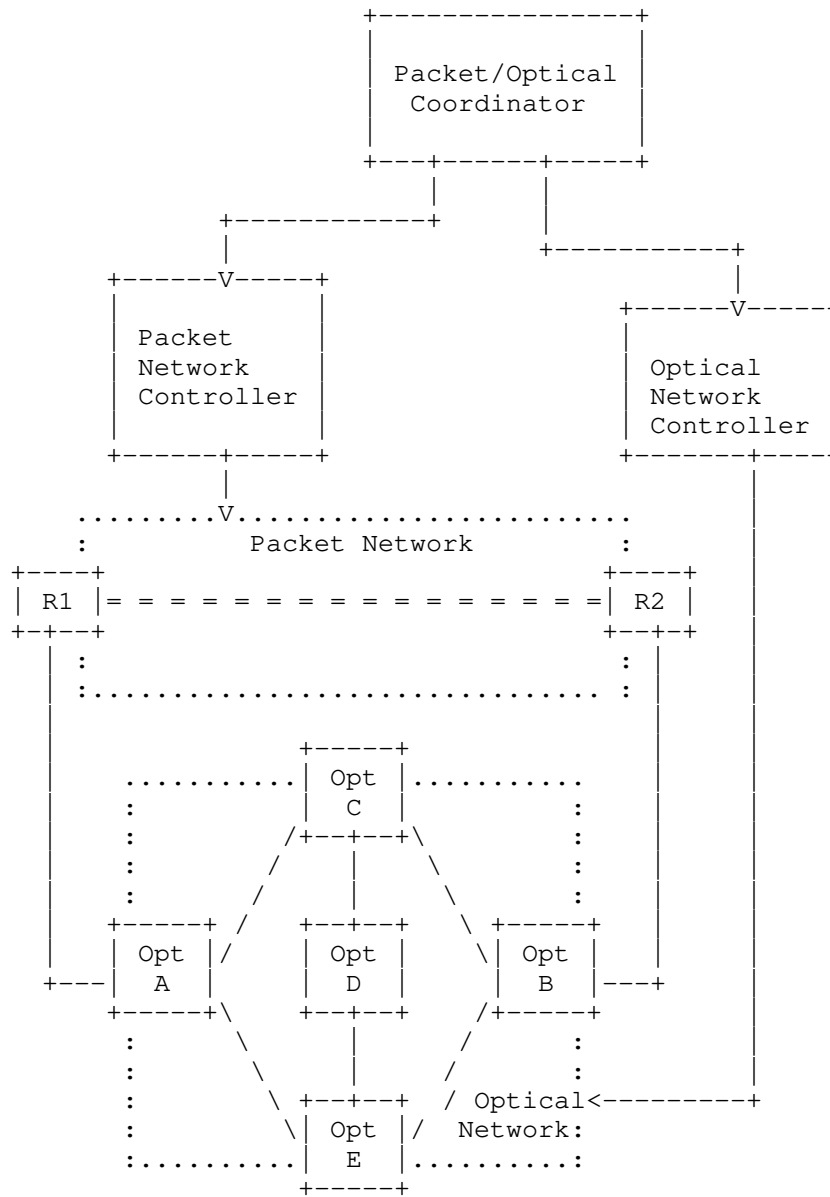


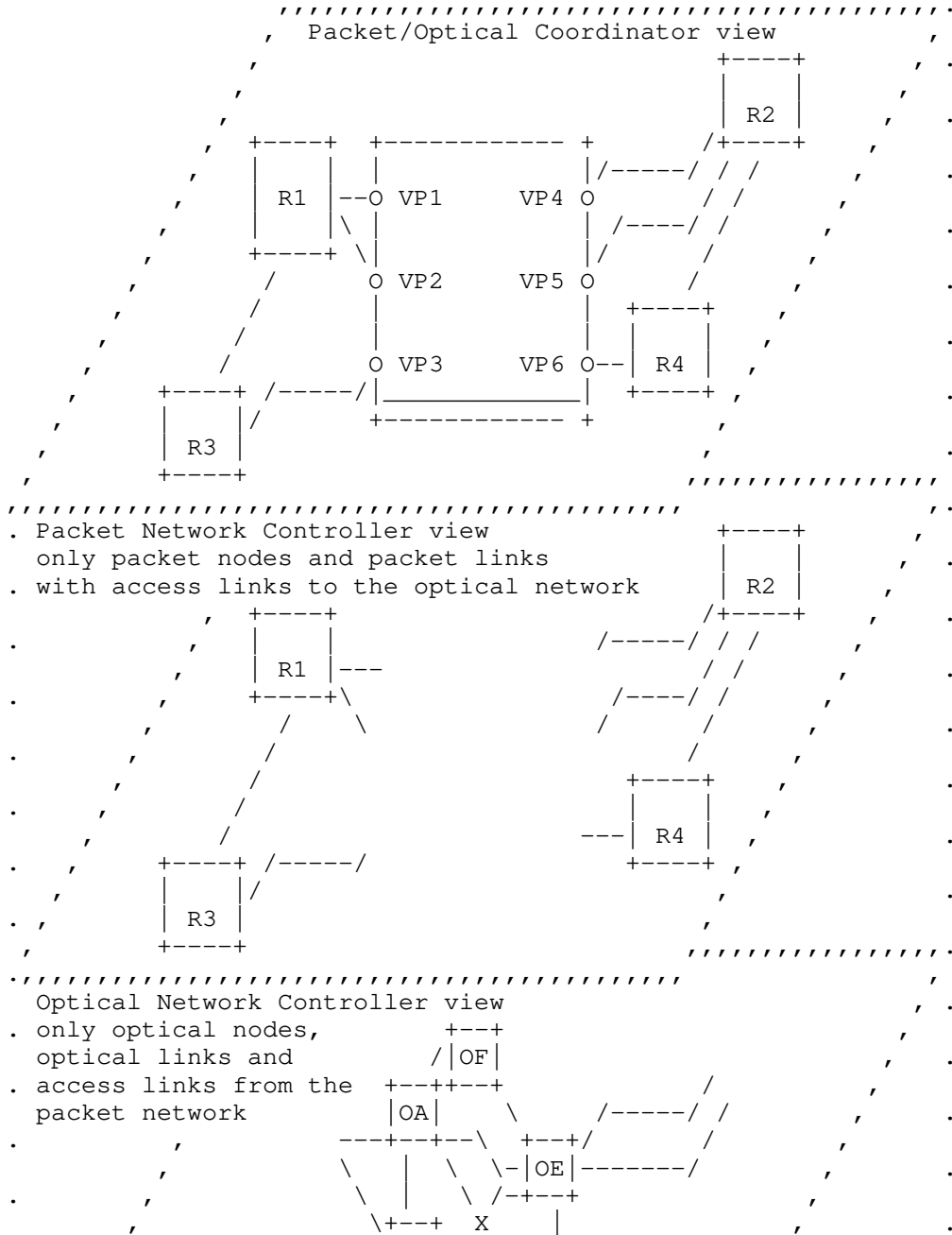
Figure 1 - Packet/Optical Integration Use Case

Figure 1 as well as Figure 2 below only show a partial view of the packet network connectivity, before additional packet connectivity is provided by the Optical network.

It is assumed that the Optical network controller provides to the packet/optical coordinator an abstracted view of the Optical network. A possible abstraction could be to represent the whole optical network as one "virtual node" with "virtual ports" connected to the access links, as shown in Figure 2.

It is also assumed that Packet network controller can provide the packet/optical coordinator the information it needs to setup connectivity between packet nodes through the Optical network (e.g., the access links).

The path computation request helps the coordinator to know the real connections that can be provided by the optical network.



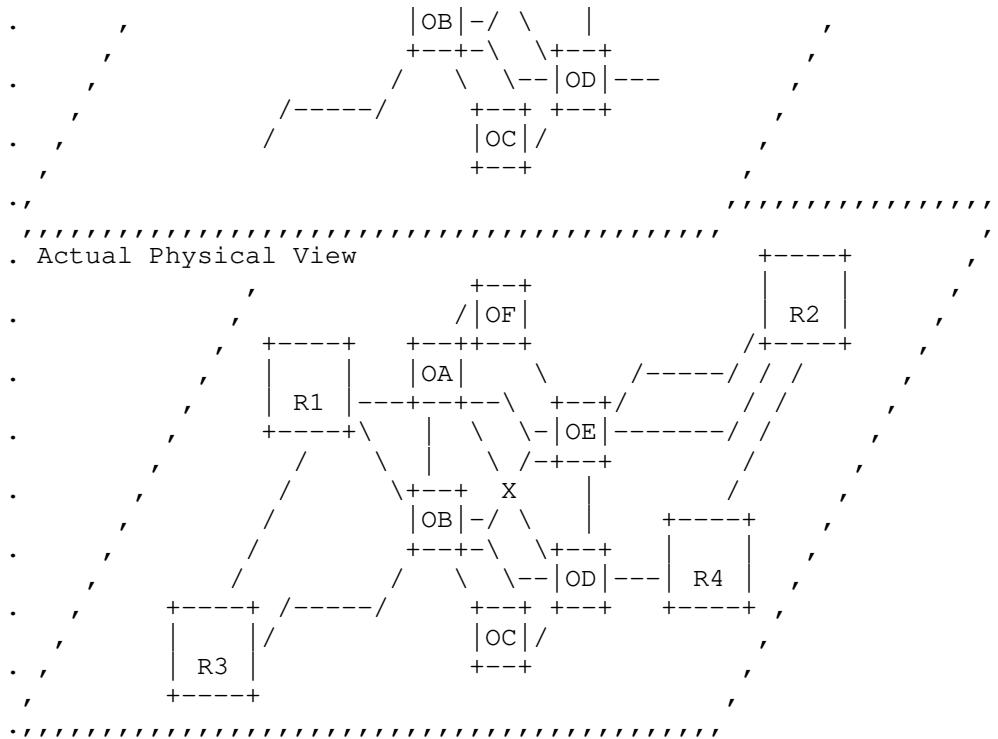


Figure 2 - Packet and Optical Topology Abstractions

In this use case, the coordinator needs to setup an optimal underlying path for an IP link between R1 and R2.

As depicted in Figure 2, the coordinator has only an "abstracted view" of the physical network, and it does not know the feasibility or the cost of the possible optical paths (e.g., VP1-VP4 and VP2-VP5), which depend from the current status of the physical resources within the optical network and on vendor-specific optical attributes.

The coordinator can request the underlying Optical domain controller to compute a set of potential optimal paths, taking into account optical constraints. Then, based on its own constraints, policy and knowledge (e.g. cost of the access links), it can choose which one of these potential paths to use to setup the optimal end-to-end path crossing optical network.

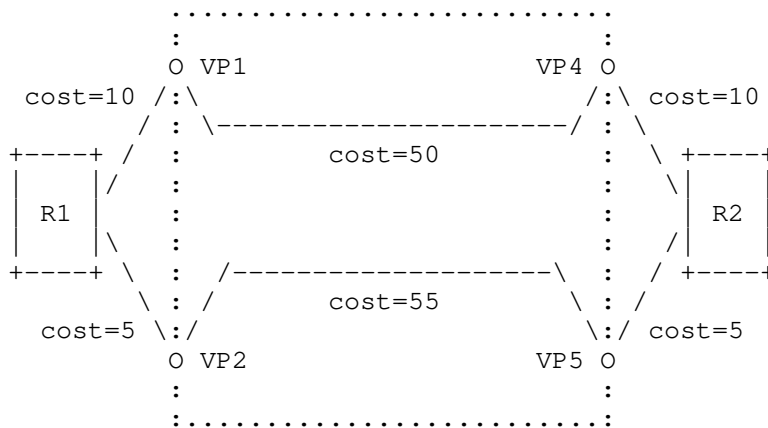


Figure 3 - Packet/Optical Path Computation Example

For example, in Figure 3, the Coordinator can request the Optical network controller to compute the paths between VP1-VP4 and VP2-VP5 and then decide to setup the optimal end-to-end path using the VP2-VP5 Optical path even this is not the optimal path from the Optical domain perspective.

Considering the dynamicity of the connectivity constraints of an Optical domain, it is possible that a path computed by the Optical network controller when requested by the Coordinator is no longer valid/available when the Coordinator requests it to be setup up. This is further discussed in section 3.3.

### 2.2. Multi-domain TE Networks

In this use case there are two TE domains which are interconnected together by multiple inter-domains links.

A possible example could be a multi-domain optical network.

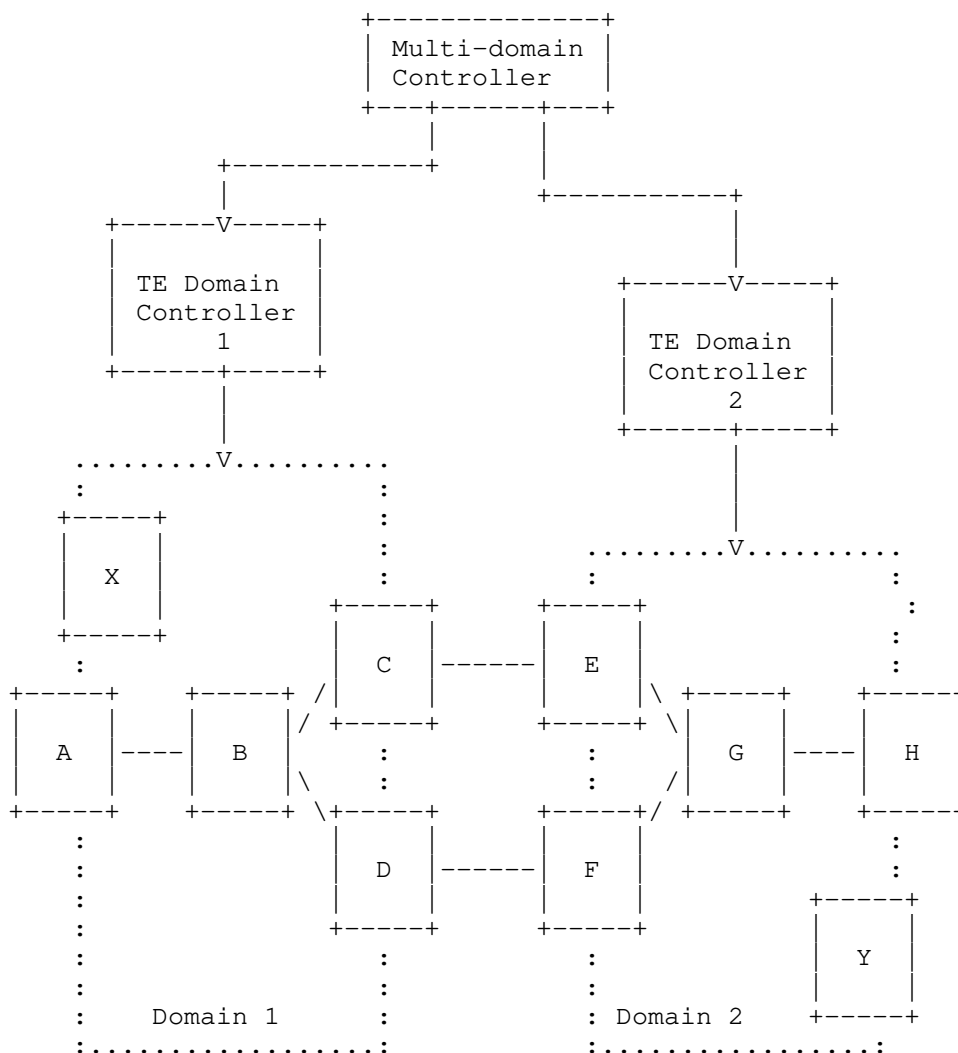


Figure 4 - Multi-domain multi-link interconnection

In order to setup an end-to-end multi-domain TE path (e.g., between nodes A and H), the multi-domain controller needs to know the feasibility or the cost of the possible TE paths within the two TE domains, which depend from the current status of the physical resources within each TE network. This is more challenging in case of optical networks because the optimal paths depend also on vendor-

specific optical attributes (which may be different in the two domains if they are provided by different vendors).

In order to setup a multi-domain TE path (e.g., between nodes A and H), the multi-domain controller can request the TE domain controllers to compute a set of intra-domain optimal paths and take decisions based on the information received. For example:

- o The multi-domain controller asks TE domain controllers to provide set of paths between A-C, A-D, E-H and F-H
- o TE domain controllers return a set of feasible paths with the associated costs: the path A-C is not part of this set (in optical networks, it is typical to have some paths not being feasible due to optical constraints that are known only by the optical domain controller)
- o The multi-domain controller will select the path A-D-F-H since it is the only feasible multi-domain path and then request the TE domain controllers to setup the A-D and F-H intra-domain paths
- o If there are multiple feasible paths, the multi-domain controller can select the optimal path knowing the cost of the intra-domain paths (provided by the TE domain controllers) and the cost of the inter-domain links (known by the multi-domain controller)

This approach may have some scalability issues when the number of TE domains is quite big (e.g. 20).

In this case, it would be worthwhile using the abstract TE topology information provided by the TE domain controllers to limit the number of potential optimal end-to-end paths and then request path computation to fewer TE domain controllers in order to decide what the optimal path within this limited set is.

For more details, see section 3.2.3.

### 2.3. Data center interconnections

In these use case, there is a TE domain which is used to provide connectivity between data centers which are connected with the TE domain using access links.





resources. It may not be able to make this decision because it has only an abstract view of the TE network (as in use case in 2.1).

The cloud network orchestrator can request to the TE network controller to compute the cost of the possible TE paths (e.g., DC1-DC2 and DC1-DC3) and to the DC controller to provide the information it needs about the required data center resources within DC2 and DC3 and then it can take the decision about the optimal solution based on this information and its policy.

#### 2.4. Backward Recursive Path Computation scenario

[RFC5441] has defined the Virtual Source Path Tree (VSPT) TLV within PCE Reply Object in order to compute inter-domain paths following a "Backward Recursive Path Computation" (BRPC) method. The main principle is to forward the PCE request message up to the destination domain. Then, each PCE involved in the computation will compute its part of the path and send it back to the requester through PCE Response message. The resulting computation is spread from destination PCE to source PCE. Each PCE is in charge of merging the path it received with the one it calculated. At the end, the source PCE merges its local part of the path with the received one to achieve the end-to-end path.

Figure 6 below show a typical BRPC scenario where 3 PCEs cooperate to compute inter-domain paths.

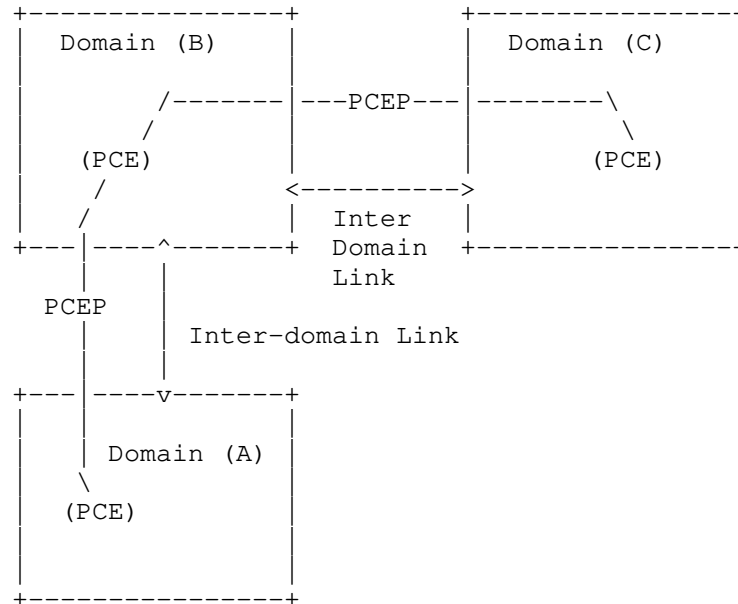


Figure 6 - BRPC Scenario

In this use case, a client can use the YANG model defined in this document to request path computation to the PCE that controls the source of the tunnel. For example, a client can request to the PCE of domain A to compute a path from a source S, within domain A, to a destination D, within domain C. Then PCE of domain A will use PCEP protocol, as per [RFC5441], to compute the path from S to D and in turn gives the final answer to the requester.

2.5. Hierarchical PCE scenario

[RFC6805] has defined an architecture and extensions to the PCE standard to compute inter-domain path following a hierarchical method. Two new roles have been defined: Parent PCE and child PCE. The parent PCE is in charge to coordinate the end-to-end path computation. For that purpose it sends to each child PCE involve in the multi-domain path computation a PCE Request message to obtain the local part of the path. Once received all answer through PCE Response message, the Parent PCE will merge the different local parts of the path to achieve the end-to-end path.

Figure 7 below shows a typical hierarchical scenario where a Parent PCE request end-to-end path to the different child PCE. Note that a

PCE could take independently the role of Child or Parent PCE depending of which PCE will request the path.

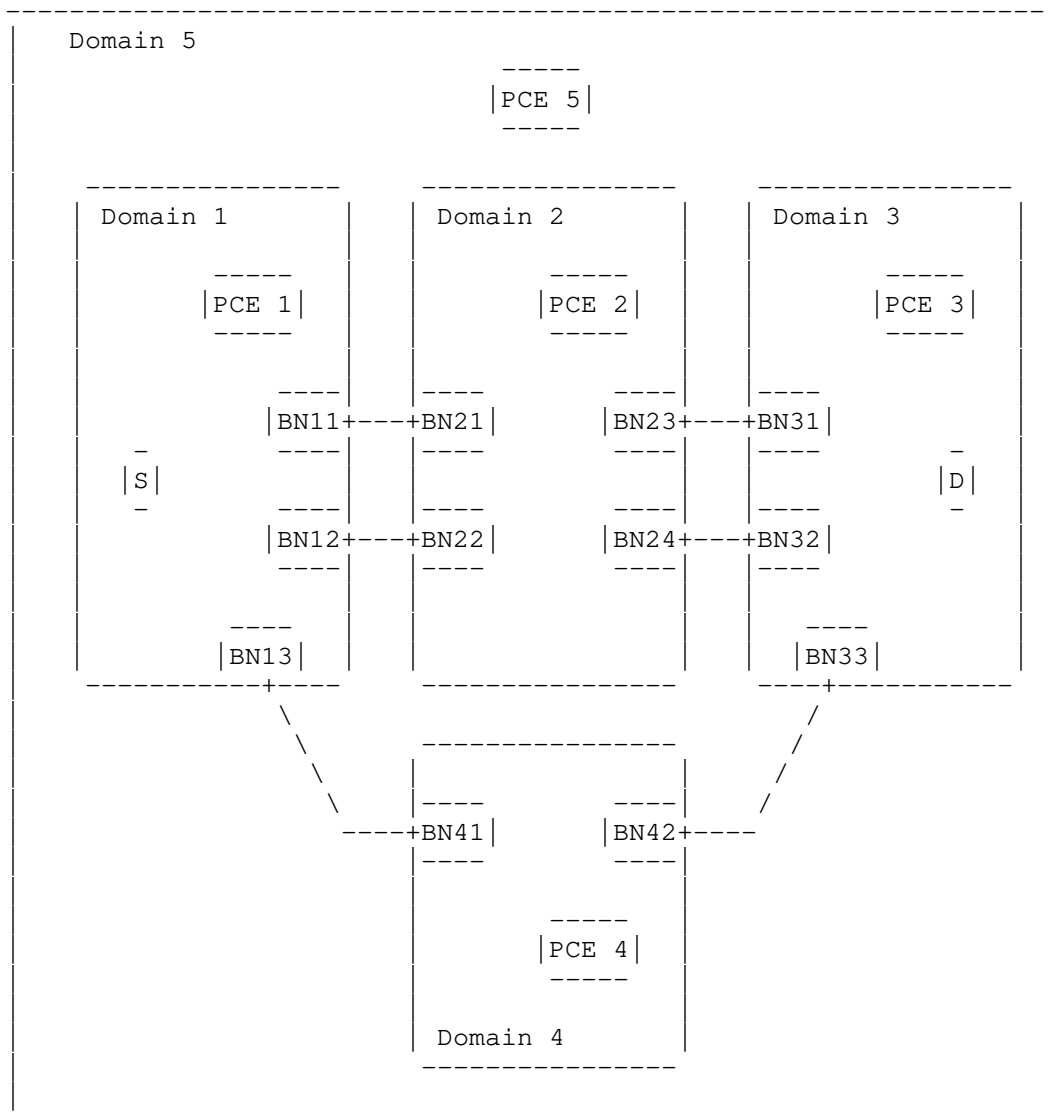


Figure 7 - Hierarchical domain topology from [RFC6805]

In this use case, a client can use the YANG model defined in this document to request to the Parent PCE a path from a source S to a destination D. The Parent PCE will in turn contact the child PCEs through PCEP protocol to compute the end-to-end path and then return the computed path to the client, using the YANG model defined in this document. For example the YANG model can be used to request to PCE5 acting as Parent PCE to compute a path from source S, within domain 1, to destination D, within domain 3. PCE5 will contact child PCEs of domain 1, 2 and 3 to obtain local part of the end-to-end path through the PCEP protocol. Once received the PCE Response message, it merges the answers to compute the end-to-end path and send it back to the client.

### 3. Motivations

This section provides the motivation for the YANG model defined in this document.

Section 3.1 describes the motivation for a YANG model to request path computation.

Section 3.2 describes the motivation for a YANG model which complements the TE Topology YANG model defined in [TE-TOPO].

Section 3.3 describes the motivation for a YANG RPC which complements the TE Tunnel YANG model defined in [TE-TUNNEL].

#### 3.1. Motivation for a YANG Model

##### 3.1.1. Benefits of common data models

The YANG data model for requesting path computation is closely aligned with the YANG data models that provide (abstract) TE topology information, i.e., [TE-TOPO] as well as that are used to configure and manage TE Tunnels, i.e., [TE-TUNNEL].

There are many benefits in aligning the data model used for path computation requests with the YANG data models used for TE topology information and for TE Tunnels configuration and management:

- o There is no need for an error-prone mapping or correlation of information.
- o It is possible to use the same endpoint identifiers in path computation requests and in the topology modeling.

- o The attributes used for path computation constraints are the same as those used when setting up a TE Tunnel.

### 3.1.2. Benefits of a single interface

The system integration effort is typically lower if a single, consistent interface is used by controllers, i.e., one data modeling language (i.e., YANG) and a common protocol (e.g., NETCONF or RESTCONF).

Practical benefits of using a single, consistent interface include:

1. **Simple authentication and authorization:** The interface between different components has to be secured. If different protocols have different security mechanisms, ensuring a common access control model may result in overhead. For instance, there may be a need to deal with different security mechanisms, e.g., different credentials or keys. This can result in increased integration effort.
2. **Consistency:** Keeping data consistent over multiple different interfaces or protocols is not trivial. For instance, the sequence of actions can matter in certain use cases, or transaction semantics could be desired. While ensuring consistency within one protocol can already be challenging, it is typically cumbersome to achieve that across different protocols.
3. **Testing:** System integration requires comprehensive testing, including corner cases. The more different technologies are involved, the more difficult it is to run comprehensive test cases and ensure proper integration.
4. **Middle-box friendliness:** Provider and consumer of path computation requests may be located in different networks, and middle-boxes such as firewalls, NATs, or load balancers may be deployed. In such environments it is simpler to deploy a single protocol. Also, it may be easier to debug connectivity problems.
5. **Tooling reuse:** Implementers may want to implement path computation requests with tools and libraries that already exist in controllers and/or orchestrators, e.g., leveraging the rapidly growing eco-system for YANG tooling.

### 3.1.3. Extensibility

Path computation is only a subset of the typical functionality of a controller. In many use cases, issuing path computation requests comes along with the need to access other functionality on the same system. In addition to obtaining TE topology, for instance also configuration of services (setup/modification/deletion) may be required, as well as:

1. Receiving notifications for topology changes as well as integration with fault management
2. Performance management such as retrieving monitoring and telemetry data
3. Service assurance, e.g., by triggering OAM functionality
4. Other fulfilment and provisioning actions beyond tunnels and services, such as changing QoS configurations

YANG is a very extensible and flexible data modeling language that can be used for all these use cases.

### 3.2. Interactions with TE Topology

The use cases described in section 2 have been described assuming that the topology view exported by each underlying SDN controller to the orchestrator is aggregated using the "virtual node model", defined in [RFC7926].

TE Topology information, e.g., as provided by [TE-TOPO], could in theory be used by an underlying SDN controllers to provide TE information to its client thus allowing a PCE available within its client to perform multi-domain path computation by its own, without requesting path computations to the underlying SDN controllers.

In case the client does not implement a PCE function, as discussed in section 1, it could not perform path computation based on TE Topology information and would instead need to request path computation to the underlying controllers to get the information it needs to find the optimal end-to-end path.

In case the client implements a PCE function, as discussed in section 1, the TE topology information needs to be complete and accurate, which would to scalability issues.

Using TE topology to provide a "virtual link model" aggregation, as described in [RFC7926], may be not sufficient, unless the aggregation provides a complete and accurate information, which would still cause scalability issues, as described in sections 3.2.1 below.

Using TE topology abstraction, as described in [RFC7926], may lead to compute unfeasible path, as described in [RFC7926] in section 3.2.2 below.

Therefore when computing an optimal multi-domain path, there is a scalability trade-off between providing complete and accurate the TE information and the number of path computation requests to the underlying SDN Domain Controllers.

The TE topology information used, in a complimentary way, to reduce the number for path computation requests to the underlying SDN domain controllers, as described in section 3.2.3 below.

### 3.2.1. TE Topology Aggregation

Using the TE Topology model, as defined in [TE-TOPO], the underlying SDN controller can export the whole TE domain as a single TE node with a "detailed connectivity matrix" (which provides specific TE attributes, such as delay, SRLGs and other TE metrics, between each ingress and egress links).

The information provided by the "detailed connectivity matrix" would be equivalent to the information that should be provided by "virtual link model" as defined in [RFC7926]. For example, in the Packet/Optical integration use case, described in section 2.1, the Optical network controller can make the information shown in Figure 3 available to the Coordinator as part of the TE Topology information and the Coordinator could use this information to calculate by its own the optimal path between R1 and R2, without requesting any additional information to the Optical network Controller.

However, when designing the amount of information to provide within the "detailed connectivity matrix", there is a tradeoff to be considered between accuracy (i.e., providing "all" the information that might be needed by the PCE available to Orchestrator) and scalability.



Figure 8 below shows another example, similar to Figure 3, where there are two possible Optical paths between VP1 and VP4 with different properties (e.g., available bandwidth and cost).

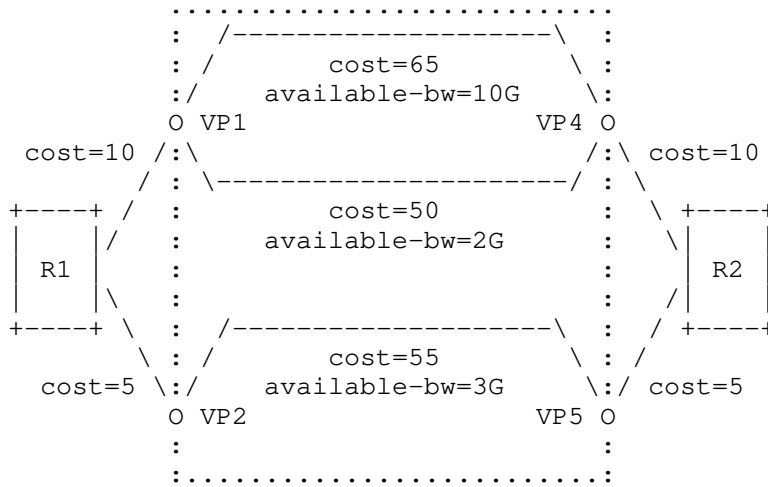


Figure 8 - Packet/Optical Path Computation Example with multiple choices

If the information in the "detailed connectivity matrix" is not complete/accurate, we can have the following drawbacks:

- o If only the VP1-VP4 path with available bandwidth of 2 Gb/s and cost 50 is reported, the client's PCE will fail to compute a 5 Gb/s path between routers R1 and R2, although this would be feasible;
- o If only the VP1-VP4 path with available bandwidth of 10 Gb/s and cost 60 is reported, the client's PCE will compute, as optimal, the 1 Gb/s path between R1 and R2 going through the VP2-VP5 path within the Optical domain while the optimal path would actually be the one going through the VP1-VP4 sub-path (with cost 50) within the Optical domain.

Reporting all the information, as in Figure 8, using the "detailed connectivity matrix", is quite challenging from a scalability perspective. The amount of this information is not just based on number of end points (which would scale as N-square), but also on many other parameters, including client rate, user

constraints/policies for the service, e.g. max latency < N ms, max cost, etc., exclusion policies to route around busy links, min OSNR margin, max preFEC BER etc. All these constraints could be different based on connectivity requirements.

Examples of how the "detailed connectivity matrix" can be dimensioned are described in Appendix A.

It is also worth noting that the "connectivity matrix" has been originally defined in WSON, [RFC7446], to report the connectivity constraints of a physical node within the WDM network: the information it contains is pretty "static" and therefore, once taken and stored in the TE data base, it can be always being considered valid and up-to-date in path computation request.

The "connectivity matrix" is sometimes confused with "optical reach table" that contain multiple (e.g. k-shortest) regen-free reachable paths for every A-Z node combination in the network. Optical reach tables can be calculated offline, utilizing vendor optical design and planning tools, and periodically uploaded to the Controller: these optical path reach tables are fairly static. However, to get the connectivity matrix, between any two sites, either a regen free path can be used, if one is available, or multiple regen free paths are concatenated to get from src to dest, which can be a very large combination. Additionally, when the optical path within optical domain needs to be computed, it can result in different paths based on input objective, constraints, and network conditions. In summary, even though "optical reach table" is fairly static, which regen free paths to build the connectivity matrix between any source and destination is very dynamic, and is done using very sophisticated routing algorithms.

Using the "basic connectivity matrix" with an abstract node to abstract the information regarding the connectivity constraints of an Optical domain, would make this information more "dynamic" since the connectivity constraints of an Optical domain can change over time because some optical paths that are feasible at a given time may become unfeasible at a later time when e.g., another optical path is established.

The information in the "detailed connectivity matrix" is even more dynamic since the establishment of another optical path may change some of the parameters (e.g., delay or available bandwidth) in the "detailed connectivity matrix" while not changing the feasibility of the path.

There is therefore the need to keep the information in the "detailed connectivity matrix" updated which means that there another tradeoff between the accuracy (i.e., providing "all" the information that might be needed by the client's PCE) and having up-to-date information. The more the information is provided and the longer it takes to keep it up-to-date which increases the likelihood that the client's PCE computes paths using not updated information.

It seems therefore quite challenging to have a "detailed connectivity matrix" that provides accurate, scalable and updated information to allow the client's PCE to take optimal decisions by its own.

Considering the example in Figure 8 with the approach defined in this document, the client, when it needs to setup an end-to-end path, it can request the Optical domain controller to compute a set of optimal paths (e.g., for VP1-VP4 and VP2-VP5) and take decisions based on the information received:

- o When setting up a 5 Gb/s path between routers R1 and R2, the Optical domain controller may report only the VP1-VP4 path as the only feasible path: the Orchestrator can successfully setup the end-to-end path passing through this Optical path;
- o When setting up a 1 Gb/s path between routers R1 and R2, the Optical domain controller (knowing that the path requires only 1 Gb/s) can report both the VP1-VP4 path, with cost 50, and the VP2-VP5 path, with cost 65. The Orchestrator can then compute the optimal path which is passing through the VP1-VP4 sub-path (with cost 50) within the Optical domain.

### 3.2.2. TE Topology Abstraction

Using the TE Topology model, as defined in [TE-TOPO], the underlying SDN controller can export an abstract TE Topology, composed by a set of TE nodes and TE links, representing the abstract view of the topology controlled by each domain controller.

Considering the example in Figure 4, the TE domain controller 1 can export a TE Topology encompassing the TE nodes A, B, C and D and the TE Link interconnecting them. In a similar way, TE domain controller 2 can export a TE Topology encompassing the TE nodes E, F, G and H and the TE Link interconnecting them.

In this example, for simplicity reasons, each abstract TE node maps with each physical node, but this is not necessary.

In order to setup a multi-domain TE path (e.g., between nodes A and H), the multi-domain controller can compute by its own an optimal end-to-end path based on the abstract TE topology information provided by the domain controllers. For example:

- o Multi-domain controller's PCE, based on its own information, can compute the optimal multi-domain path being A-B-C-E-G-H, and then request the TE domain controllers to setup the A-B-C and E-G-H intra-domain paths
- o But, during path setup, the domain controller may find out that A-B-C intra-domain path is not feasible (as discussed in section 2.2, in optical networks it is typical to have some paths not being feasible due to optical constraints that are known only by the optical domain controller), while only the path A-B-D is feasible
- o So what the multi-domain controller computed is not good and need to re-start the path computation from scratch

As discussed in section 3.2.1, providing more extensive abstract information from the TE domain controllers to the multi-domain controller may lead to scalability problems.

In a sense this is similar to the problem of routing and wavelength assignment within an Optical domain. It is possible to do first routing (step 1) and then wavelength assignment (step 2), but the chances of ending up with a good path is low. Alternatively, it is possible to do combined routing and wavelength assignment, which is known to be a more optimal and effective way for Optical path setup. Similarly, it is possible to first compute an abstract end-to-end path within the multi-domain Orchestrator (step 1) and then compute an intra-domain path within each Optical domain (step 2), but there are more chances not to find a path or to get a suboptimal path that performing per-domain path computation and then stitch them.

### 3.2.3. Complementary use of TE topology and path computation

As discussed in section 2.2, there are some scalability issues with path computation requests in a multi-domain TE network with many TE domains, in terms of the number of requests to send to the TE domain controllers. It would therefore be worthwhile using the TE topology

information provided by the domain controllers to limit the number of requests.

An example can be described considering the multi-domain abstract topology shown in Figure 9. In this example, an end-to-end TE path between domains A and F needs to be setup. The transit domain should be selected between domains B, C, D and E.

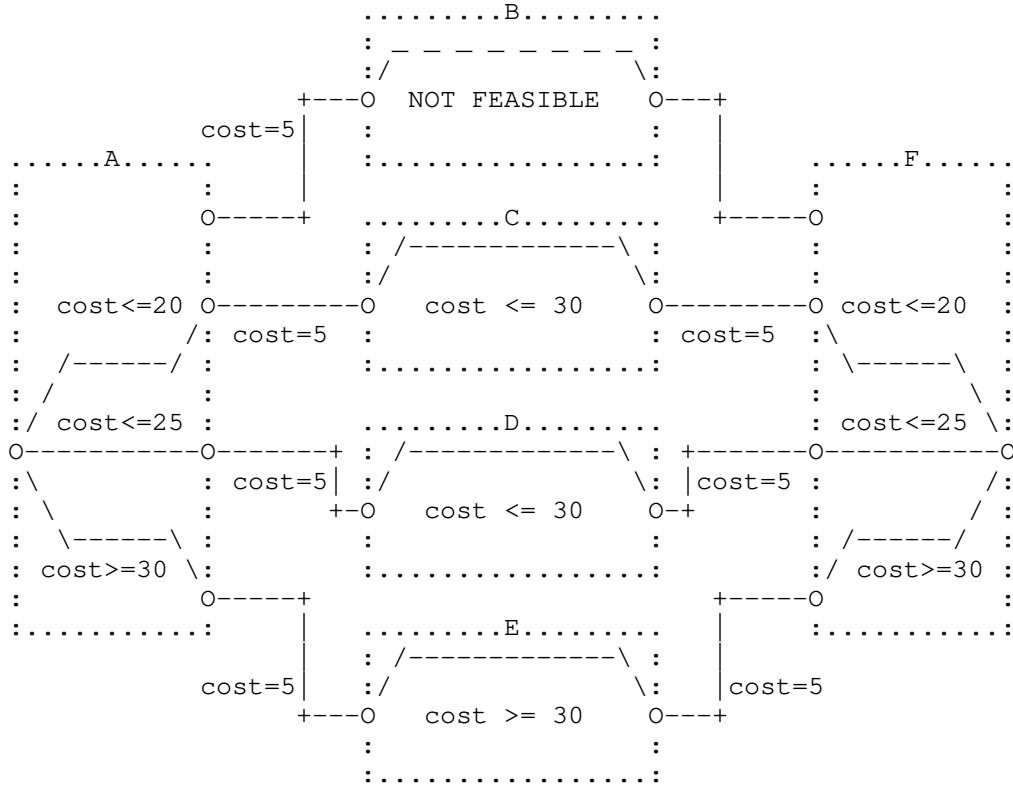


Figure 9 - Multi-domain with many domains (Topology information)

The actual cost of each intra-domain path is not known a priori from the abstract topology information. The Multi-domain controller only knows, from the TE topology provided by the underlying domain controllers, the feasibility of some intra-domain paths and some upper-bound and/or lower-bound cost information. With this information, together with the cost of inter-domain links, the Multi-domain controller can understand by its own that:

- o Domain B cannot be selected as the path connecting domains A and E is not feasible;
- o Domain E cannot be selected as a transit domain since it is know from the abstract topology information provided by domain controllers that the cost of the multi-domain path A-E-F (which is 100, in the best case) will be always be higher than the cost of the multi-domain paths A-D-F (which is 90, in the worst case) and A-E-F (which is 80, in the worst case)

Therefore, the Multi-domain controller can understand by its own that the optimal multi-domain path could be either A-D-F or A-E-F but it cannot known which one of the two possible option actually provides the optimal end-to-end path.

The Multi-domain controller can therefore request path computation only to the TE domain controllers A, D, E and F (and not to all the possible TE domain controllers).

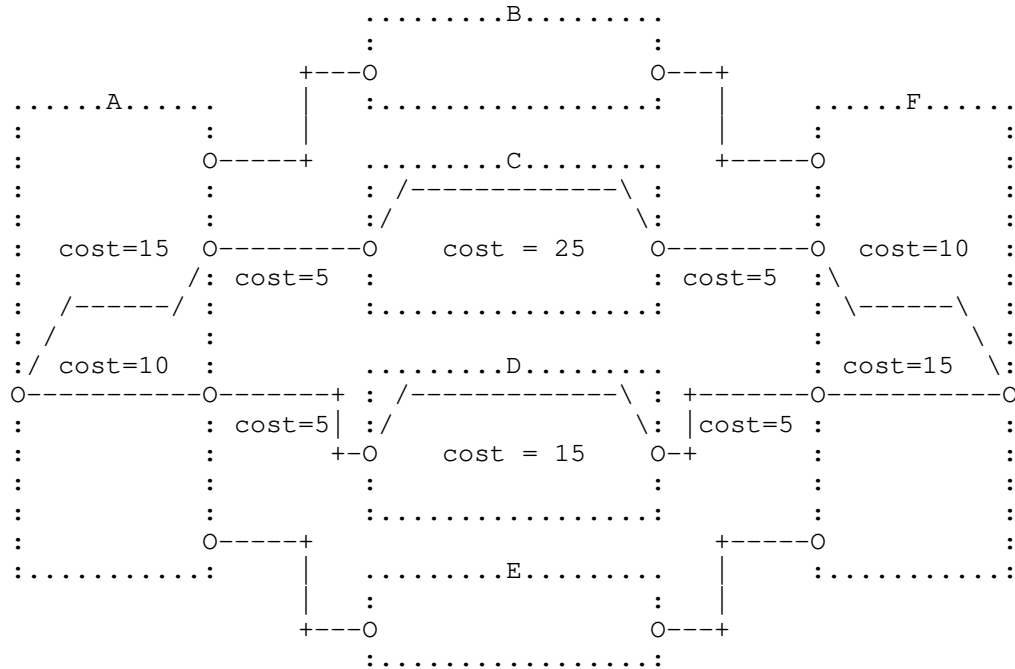


Figure 10 - Multi-domain with many domains (Path Computation information)

Based on these requests, the Multi-domain controller can know the actual cost of each intra-domain paths which belongs to potential optimal end-to-end paths, as shown in Figure 10, and then compute the optimal end-to-end path (e.g., A-D-F, having total cost of 50, instead of A-C-F having a total cost of 70).

### 3.3. Path Computation RPC

The TE Tunnel YANG model, defined in [TE-TUNNEL], can support the need to request path computation.

It is possible to request path computation by configuring a "compute-only" TE tunnel and retrieving the computed path(s) in the LSP(s) Record-Route Object (RRO) list as described in section 3.3.1 of [TE-TUNNEL].

This is a stateful solution since the state of each created "compute-only" TE tunnel needs to be maintained, in the YANG data-stores (at least in the running data-store and operational data-store), and updated, when underlying network conditions change.

It is very useful to provide both the options of using an RPC as well as of setting up TE Tunnel paths in "compute-only" mode. It is suggested to use the RPC as much as possible and to rely on "compute-only" TE Tunnel paths, when really needed.

The RPC mechanism allows requesting path computation using a simple atomic operation, without creating any state in the YANG data-stores, and it is the natural option/choice, especially with stateless PCE.

This solution assumes that the underlying SDN controller (e.g., a PNC) will compute a path twice during the process to setup an LSP: at time T1, when its client (e.g., an MDSC) sends a path computation RPC request to it, and later, at time T2, when the same client (MDSC) creates a te-tunnel requesting the setup of the LSP. The underlying assumption is that, if network conditions have not changed, the same path that has been computed at time T1 is also computed at time T2 by the underlying SDN controller (e.g. PNC) and therefore the path that is setup at time T2 is exactly the same path that has been computed at time T1.

Since the operation is stateless, there is no guarantee that the returned path would still be available when path setup is requested: this does not cause major issues in case the time between path

computation and path setup is short (especially if compared with the time that would be needed to update the information of a very detailed connectivity matrix).

In most of the cases, there is even no need to guarantee that the path that has been setup is the exactly same as the path that has been returned by path computation, especially if it has the same or even better metrics. Depending on the abstraction level applied by the server, the client may also not know the actual computed path.

The most important requirement is that the required global objectives (e.g., multi-domain path metrics and constraints) are met. For this reason a path verification phase is necessary to verify that the actual path that has been setup meets the global objectives (for example in a multi-domain network, the resulting end-to-end path meets the required end-to-end metrics and constraints).

In most of the cases, even if the setup path is not exactly the same as the path returned by path computation, its metrics and constraints are "good enough" (the path verification passes successfully). In the few corner cases where the path verification fails, it is possible repeat the whole process (path computation, path setup and path verification).

In case it is required to setup at T2 exactly the same path computed at T1, the RPC solution should not be used and, instead, a "compute-only" TE tunnel path should be setup, allowing also notifications in case the computed path has been changed.

In this case, at time T1, the client (MDSC) creates a te-tunnel in a compute-only mode in the config data-store and later, at time T2, changes the configuration of that te-tunnel (not to be any more in a compute-only mode) to trigger the setup of the LSP over the path which have been computed at time T1 and reported in the operational data-store.

It is worth noting that also using the "compute-only" TE Tunnel path, although increasing the likelihood that the computed path is available at path setup, does not guaranteed that because notifications may not be reliable or delivered on time. Path verification is needed also in this case.

The solution based on "compute-only" TE Tunnel path has also the following drawbacks:



- o Several messages required for any path computation
- o Requires persistent storage in the provider controller
- o Need for garbage collection for stranded paths
- o Process burden to detect changes on the computed paths in order to provide notifications update

### 3.3.1. Temporary reporting of the computed path state

This section describes an optional extension to the stateless behavior of the path computation RPC, where the underlying SDN controller, after having received a path computation RPC request, maintains some "transient state" associated with the computed path, allowing the client to request the setup of exactly that path, if still available.

This is similar to the "compute-only" TE Tunnel path solution but, to avoid the drawbacks of the stateful approach, is leveraging the path computation RPC and the separation between configuration and operational data-store, as defined in the NMDA architecture [RFC8342].

The underlying SDN controller, after having computed a path, as requested by a path computation RPC, also creates a te-tunnel instance within the operational data-store, to store that computed path. This would be similar to a "compute-only" TE Tunnel path, with the only difference that there is no associated te-tunnel instance within the running data-store.

Since underlying SDN controller stores in the operational data-store the computed path based on an abstract topology it exposes, it also remembers, internally, which is the actual native path (physical path), within its native topology (physical topology), associated with that compute-only te-tunnel instance.

Afterwards, the client (e.g., MDSC) can request to setup that specific path by creating a te-tunnel instance (not in compute-only mode) in the running data-store using the same tunnel-name of the existing te-tunnel in the operational data-store: this will trigger the underlying SDN controller to setup that path, if still available.

There are still cases where the path being setup is not exactly the same as the path that has been computed:

- o When the tunnel is configured with path constraints which are not compatible with the computed path
- o When the tunnel setup is requested after the resources of the computed path are no longer available
- o When the tunnel setup is requested after the computed path is no longer known (e.g. due to a server reboot) by the underlying SDN controller

In all these cases, the underlying SDN controller should compute and setup a new path.

Therefore the "path verification" phase, as described in section 3.3 above, is still needed to check that the path that has been setup is still "good enough".

Since this new approach is not completely stateless, garbage collection is implemented using a timeout that, when it expires, triggers the removal of the computed path from the operational data-store. This operation is fully controlled by the underlying SDN controller without the need for any action to be taken by the client that is not able to act on the operational data-store. The default value of this timeout is 10 minutes but a different value may be configured by the client.

In addition, it is possible for the client to tag each path computation requests with a transaction-id allowing for a faster removal of all the paths associated with a transaction-id, without waiting for their timers to expire.

The underlying SDN controller can remove from the operational data-store all the paths computed with a given transaction-id which have not been setup either when it receives a Path Compute Delete Tunnel Action RPC request for that transaction-id or, automatically, right after the setup up of a path that have been previously computed with that transaction-id.

This possibility is useful when multiple paths are computed but, at most, only one is setup (e.g., in multi-domain path computation scenario scenarios). After the selected path has been setup (e.g, in one domain during multi-domain path setup), all the other

alternative computed paths can be automatically deleted by the underlying SDN controller (since no longer needed). The client can also request, using the Path Delete RPC request, the underlying SDN controller to remove all the computed paths, if none of them is going to be setup (e.g., in a transit domain not being selected by multi-domain path computation and so not being automatically deleted).

This approach is complimentary and not alternative to the timer which is always needed to avoid stranded computed paths being stored in the operational data-store when no path is setup and no explicit delete RPC is received.

#### 4. Path Computation and Optimization for multiple paths

There are use cases, where it is advantageous to request path computation for a set of paths, through a network or through a network domain, using a single request [RFC5440].

In this case, sending a single request for multiple path computations, instead of sending multiple requests for each path computation, would reduce the protocol overhead and it would consume less resources (e.g., threads in the client and server).

In the context of a typical multi-domain TE network, there could be multiple choices for the ingress/egress points of a domain and the Multi-domain controller needs to request path computation between all the ingress/egress pairs to select the best pair. For example, in the example of section 2.2, the Multi-domain controller needs to request the TE network controller 1 to compute the A-C and the A-D paths and to the TE network controller 2 to compute the E-H and the F-H paths.

It is also possible that the Multi-domain controller receives a request to setup a group of multiple end to end connections. The multi-domain controller needs to request each TE domain controller to compute multiple paths, one (or more) for each end to end connection.

There are also scenarios where it can be needed to request path computation for a set of paths in a synchronized fashion.

One example could be computing multiple diverse paths. Computing a set of diverse paths in a not-synchronized fashion, leads to the possibility of not being able to satisfy the diversity requirement.

In this case, it is preferable to compute a sub-optimal primary path for which a diversely routed secondary path exists.

There are also scenarios where it is needed to request optimizing a set of paths using objective functions that apply to the whole set of paths, see [RFC5541], e.g. to minimize the sum of the costs of all the computed paths in the set.

## 5. YANG Model for requesting Path Computation

This document define a YANG RPC to request path computation as an "augmentation" of tunnel-rpc, defined in [TE-TUNNEL]. This model provides the RPC input attributes that are needed to request path computation and the RPC output attributes that are needed to report the computed paths.

```

augment /te:tunnels-path-compute/te:input/te:path-compute-info:
augment /te:tunnels-path-compute/te:input/te:path-compute-info:
  +-- path-request* [request-id]
  |   +-- request-id                               uint32
  |   .....
augment /te:tunnels-path-compute/te:output/te:path-compute-result:
  +--ro response* [response-id]
  |   +--ro response-id                             uint32
  |   +--ro computed-paths-properties
  |   |   +--ro computed-path-properties* [k-index]
  |   |   |   +--ro k-index                         uint8
  |   |   |   +--ro path-properties
  |   |   |   .....

```

This model extensively re-uses the grouping defined in [TE-TUNNEL] to ensure maximal syntax and semantics commonality.

This YANG model allows one RPC to include multiple path requests, each path request being identified by a request-id. Therefore, one RPC can return multiple responses, one for each path request, being identified by a response-id equal to the corresponding request-id. Each response reports one or more computed paths, as requested by the k-requested-paths attribute. By default, each response reports one computed path.

## 5.1. Synchronization of multiple path computation requests

The YANG model permits to synchronize a set of multiple path requests (identified by specific request-id) all related to a "svec" container emulating the syntax of "SVEC" PCEP object [RFC5440].

```

+-- synchronization* [synchronization-id]
  +-- synchronization-id   uint32
  +-- svec
    | +-- relaxable?         boolean
    | +-- disjointness?     te-path-disjointness
    | +-- request-id-number* uint32
  +-- svec-constraints
    | +-- path-metric-bound* [metric-type]
    |   +-- metric-type     identityref
    |   +-- upper-bound?   uint64
  +-- path-srlgs-lists
    | +-- path-srlgs-list* [usage]
    |   +-- usage           identityref
    |   +-- values*        srlg
  +-- path-srlgs-names
    | +-- path-srlgs-name* [usage]
    |   +-- usage           identityref
    |   +-- names*         string
  +-- exclude-objects
    | +-- excludes* [index]
    |   +-- index           uint32
    |   +-- (type)?
    |     +--:(numbered-node-hop)
    |       | +-- numbered-node-hop
    |       |   +-- node-id     te-node-id
    |       |   +-- hop-type?   te-hop-type
    |     +--:(numbered-link-hop)
    |       | +-- numbered-link-hop
    |       |   +-- link-tp-id   te-tp-id
    |       |   +-- hop-type?   te-hop-type
    |       |   +-- direction?  te-link-direction
    |     +--:(unnumbered-link-hop)
    |       | +-- unnumbered-link-hop
    |       |   +-- link-tp-id   te-tp-id

```

```

    |         |         +--- node-id           te-node-id
    |         |         +--- hop-type?        te-hop-type
    |         |         +--- direction?       te-link-direction
    |         |         +---:(as-number)
    |         |         |         +--- as-number-hop
    |         |         |         +--- as-number   inet:as-number
    |         |         |         +--- hop-type?   te-hop-type
    |         |         +---:(label)
    |         |         |         +--- label-hop
    |         |         |         |         +--- te-label
    |         |         |         |         |         +--- (technology)?
    |         |         |         |         |         |         +---:(generic)
    |         |         |         |         |         |         |         +--- generic?
    |         |         |         |         |         |         |         |         rt-types:generalized-label
    |         |         |         |         |         |         +--- direction?   te-label-direction
    |         |         +--- optimizations
    |         |         |         +--- (algorithm)?
    |         |         |         |         +---:(metric) {te-types:path-optimization-metric}?
    |         |         |         |         |         +--- optimization-metric* [metric-type]
    |         |         |         |         |         |         +--- metric-type   identityref
    |         |         |         |         |         |         +--- weight?       uint8
    |         |         |         |         +---:(objective-function)
    |         |         |         |         |         {te-types:path-optimization-objective-
function}?
    |         |         |         |         |         +--- objective-function
    |         |         |         |         |         |         +--- objective-function-type?   identityref

```

The model, in addition to the metric types, defined in [TE-TUNNEL], which can be applied to each individual path request, defines additional specific metrics types that apply to a set of synchronized requests, as referenced in [RFC5541].

```

identity svec-metric-type {
  description
    "Base identity for SVEC metric type.";
  reference
    "RFC5541: Encoding of Objective Functions in the Path
    Computation Element Communication Protocol (PCEP).";
}

```

```
identity svec-metric-cumul-te {
  base svec-metric-type;
  description
    "Cumulative TE cost.";
  reference
    "RFC5541: Encoding of Objective Functions in the Path
    Computation Element Communication Protocol (PCEP).";
}

identity svec-metric-cumul-igp {
  base svec-metric-type;
  description
    "Cumulative IGP cost.";
  reference
    "RFC5541: Encoding of Objective Functions in the Path
    Computation Element Communication Protocol (PCEP).";
}

identity svec-metric-cumul-hop {
  base svec-metric-type;
  description
    "Hop cumulative path metric.";
}

identity svec-metric-aggregate-bandwidth-consumption {
  base svec-metric-type;
  description
    "Aggregate bandwidth consumption.";
  reference
    "RFC5541: Encoding of Objective Functions in the Path
    Computation Element Communication Protocol (PCEP).";
}

identity svec-metric-load-of-the-most-loaded-link {
  base svec-metric-type;
  description
    "Load of the most loaded link.";
  reference
    "RFC5541: Encoding of Objective Functions in the Path
```

```

    Computation Element Communication Protocol (PCEP).";
}

```

## 5.2. Returned metric values

This YANG model provides a way to return the values of the metrics computed by the path computation in the output of RPC, together with other important information (e.g. srlg, affinities, explicit route), emulating the syntax of the "C" flag of the "METRIC" PCEP object [RFC5440]:

```

|      +--ro path-properties
|      |      +--ro path-metric* [metric-type]
|      |      |      +--ro metric-type          identityref
|      |      |      +--ro accumulative-value?  uint64
|      |      +--ro path-affinities-values
|      |      |      +--ro path-affinities-value* [usage]
|      |      |      |      +--ro usage          identityref
|      |      |      |      +--ro value?       admin-groups
|      |      +--ro path-affinity-names
|      |      |      +--ro path-affinity-name* [usage]
|      |      |      |      +--ro usage          identityref
|      |      |      |      +--ro affinity-name* [name]
|      |      |      |      |      +--ro name      string
|      |      +--ro path-srlgs-lists
|      |      |      +--ro path-srlgs-list* [usage]
|      |      |      |      +--ro usage          identityref
|      |      |      |      +--ro values*       srlg
|      |      +--ro path-srlgs-names
|      |      |      +--ro path-srlgs-name* [usage]
|      |      |      |      +--ro usage          identityref
|      |      |      |      +--ro names*       string
|      |      +--ro path-route-objects
|      |      .....

```

It also allows to request in the input of RPC which information (metrics, srlg and/or affinities) should be returned:

```

|      +-- request-id          uint32
|      .....
|      +-- requested-metrics* [metric-type]

```



```

| | +-- metric-type      identityref
| | +-- return-srlgs?   boolean
| | +-- return-affinities?  boolean
| | .....

```

This feature is essential for path computation in a multi-domain TE network as described in section 2.2. In this case, the metrics returned by a path computation requested to a given TE network controller must be used by the client to compute the best end-to-end path. If they are missing the client cannot compare different paths calculated by the TE network controllers and choose the best one for the optimal e2e path.

### 5.3. Multiple Paths Requests for the same TE Tunnel

The YANG model allows including multiple requests for different paths intended to be used within the same tunnel or within different tunnels.

When multiple requested paths are intended to be used within the same tunnel (e.g., requesting path computation for the primary and secondary paths of a protected tunnel), the set of attributes that are intended to be configured on per-tunnel basis rather than on per-path basis are common to all these path requests. These attributes includes both attributes which can be configured only a per-tunnel basis (e.g., tunnel-name, source/destination TTP, encoding and switching-type) as well attributes which can be configured also on a per-path basis (e.g., the te-bandwidth or the associations).

Therefore, a tunnel-attributes list is defined, within the path computation request RPC:

```

+-- tunnel-attributes* [tunnel-name]
|   +-- tunnel-name      string
|   +-- encoding?       identityref
|   +-- switching-type?  identityref
|   .....

```

The path requests that are intended to be used within the same tunnel should reference the same entry in the tunnel-attributes list. This allows:

- o avoiding repeating the same set of per-tunnel parameters on multiple requested paths;
- o the server to understand what attributes are intended to be configured on a per-tunnel basis (e.g., the te-bandwidth configured in the tunnel-attributes) and what attributes are intended to be configured on a per-path basis (e.g., the te-bandwidth configured in the path-request). This could be useful especially when the server also creates a te-tunnel instance within the operational data-store to report the computed paths, as described in section 3.3.1: in this case, the tunnel-name is also used as the suggested name for that te-tunnel instance.

The YANG model allows also including requests for paths intended to modify existing tunnels (e.g., adding a protection path for an existing un-protected tunnel). In this case, the per-tunnel attributes are already provided in the existing te-tunnel instance and do not need to be re-configured in the path computation request RPC. Therefore, these requests should reference an existing te-tunnel instance.

It is also possible to request computing paths without indicating in which tunnel they are intended to be used (e.g., in case of an unprotected tunnel). In this case, the per-tunnel attributes could be provided together with the per-path attributes in the path request, without using the tunnel-attributes list.

The choices below are defined to distinguish whether the per-tunnel attributes are configured by values (providing a set of attributes) or by reference (providing a leafref), to either a te-tunnel instance, if it exists, or to an entry of the tunnel-attributes list, if the te-tunnel instance does not exist):

```

|   +-- (tunnel-attributes)?
|   |   +--:(reference)
|   |   |   +-- (tunnel-exist)?
|   |   |   |   +--:(tunnel-ref)
|   |   |   |   |   +-- tunnel-ref           te:tunnel-ref
|   |   |   |   +--:(tunnel-attributes-ref)
|   |   |   |   |   +-- tunnel-attributes-ref   leafref
|   |   |   .....
|   |   |   +--:(value)
|   |   |   +-- tunnel-name?                   string

```

```

| | .....
| | +-- encoding?                identityref
| | +-- switching-type?         identityref
| | .....

```

The (values) case will provide the set of attributes that are configured only on per-tunnel basis (e.g., tunnel-name, source/destination TTP, encoding and switching-type). The role of the path being requested is specified by the (path-role) choice:

```

| | +-- (path-role)?
| | | +--:(primary-path)
| | | | +-- primary-path-name?    string
| | | +--:(secondary-path)
| | | | +-- secondary-path-name?  string

```

It is worth noting that a te-tunnel with only one path cannot have any reverse path.

The (reference) case provides the information needed to associate multiple path requests that are intended to be used within the same tunnel.

In order to indicate the role of the path being requested within the intended tunnel (e.g., primary or secondary path), the (tunnel-path-role) choice is defined:

```

| | | +-- (tunnel-path-role)
| | | | +--:(primary-path)
| | | | | +-- primary-path!
| | | | | .....
| | | | +--:(secondary-path)
| | | | | +-- secondary-path
| | | | | .....
| | | | +--:(primary-reverse-path)
| | | | | +-- primary-reverse-path
| | | | | .....
| | | | +--:(secondary-reverse-path)
| | | | | +-- secondary-reverse-path
| | | | | .....

```



A requested primary reverse path can reference either a requested primary paths, or, in case it is intended to be used within an existing te-tunnel, an existing primary-path.

The secondary-reverse-path container indicates that the requested path is indented to be used as a secondary reverse path and it contains at least references to one or more primary paths, whose primary reverse path can use it as a candidate secondary reverse path:

```

|   |   |   +-- secondary-reverse-path
|   |   |   .....
|   |   |   +-- primary-reverse-path-ref* [index]
|   |   |   |   +-- index                uint32
|   |   |   |   +-- (primary-reverse-path-exist)?
|   |   |   |   |   +--:(path-ref)
|   |   |   |   |   |   +-- primary-path-ref    leafref
|   |   |   |   |   |   +--:(path-request-ref)
|   |   |   |   |   |   +-- path-request-ref    leafref

```

A requested secondary reverse path can reference any requested primary paths, and, in case they are intended to be used within an existing te-tunnel, it could reference also existing primary-paths.

The secondary-reverse-path container can also contain some attributes which are configured only on secondary reverse paths (e.g., the protection-type).

#### 5.4. Multi-Layer Path Computation

The models supports requesting multi-layer path computation following the same approach based on dependency tunnels, as defined in [TE-TUNNEL].

The tunnel-attributes of a given client-layer path request can reference server-layer TE tunnels which can already exist in the YANG data-store or be specified in the tunnel-attributes list, within the same RPC request:

```

|   |   |   +-- dependency-tunnels
|   |   |   |   +-- dependency-tunnel* [name]
|   |   |   |   |   +-- name
|   |   |   |   |   |   -> ../../../../../../tunnels/tunnel/name

```

```

|   |   |   +-- encoding?          identityref
|   |   |   +-- switching-type?    identityref
|   |   |   +-- dependency-tunnel-attributes* [name]
|   |   |       +-- name            leafref
|   |   |       +-- encoding?      identityref
|   |   |       +-- switching-type? identityref

```

In a similar way as in [TE-TUNNEL], the server-layer tunnel attributes should provide the information of what would be the dynamic link in the client layer topology supported by that tunnel, if instantiated:

```

|   |   |   +-- hierarchical-link
|   |   |       +-- local-te-node-id?      te-types:te-node-id
|   |   |       +-- local-te-link-tp-id?   te-types:te-tp-id
|   |   |       +-- remote-te-node-id?     te-types:te-node-id
|   |   |       +-- te-topology-identifier
|   |   |           +-- provider-id?    te-global-id
|   |   |           +-- client-id?     te-global-id
|   |   |           +-- topology-id?   te-topology-id

```

It is worth noting that since path computation RPC is stateless, the dynamic hierarchical links configured for the server-layer tunnel attributes cannot be used for path computation of any client-layer path unless explicitly referenced in the dependency-tunnel-attributes list within the same RPC request.

## 6. YANG model for TE path computation

### 6.1. YANG Tree

Figure 11 below shows the tree diagram of the YANG model defined in module `ietf-te-path-computation.yang`.

```

module: ietf-te-path-computation
  augment /te:tunnels-path-compute/te:input/te:path-compute-info:
    +-- path-request* [request-id]
    |   +-- request-id                uint32
    |   +-- (tunnel-attributes)?
    |   |   +--:(reference)
    |   |   |   +-- (tunnel-exist)?

```

```

+---:(tunnel-ref)
|   +--- tunnel-ref                te:tunnel-ref
+---:(tunnel-attributes-ref)
|   +--- tunnel-attributes-ref     leafref
+--- path-name?                    string
+--- (tunnel-path-role)
+---:(primary-path)
|   +--- primary-path!
|   |   +--- preference?           uint8
|   |   +--- k-requested-paths?   uint8
+---:(secondary-path)
|   +--- secondary-path
|   |   +--- preference?           uint8
|   |   +--- protection-type?     identityref
|   |   +--- restoration-type?    identityref
|   |   +--- primary-path-ref* [index]
|   |   |   +--- index             uint32
|   |   |   +--- (primary-path-exist)?
|   |   |   |   +---:(path-ref)
|   |   |   |   |   +--- primary-path-ref     leafref
|   |   |   |   |   +---:(path-request-ref)
|   |   |   |   |   |   +--- path-request-ref     leafref
+---:(primary-reverse-path)
|   +--- primary-reverse-path
|   |   +--- (primary-path-exist)?
|   |   |   +---:(path-ref)
|   |   |   |   +--- primary-path-ref     leafref
|   |   |   |   +---:(path-request-ref)
|   |   |   |   |   +--- path-request-ref     leafref
+---:(secondary-reverse-path)
|   +--- secondary-reverse-path
|   |   +--- preference?           uint8
|   |   +--- protection-type?     identityref
|   |   +--- restoration-type?    identityref
|   |   +--- primary-reverse-path-ref* [index]
|   |   |   +--- index             uint32
|   |   |   +--- (primary-reverse-path-exist)?
|   |   |   |   +---:(path-ref)
|   |   |   |   |   +--- primary-path-ref     leafref

```

```

|                                     +---:(path-request-ref)
|                                     +--- path-request-ref  leafref
+---:(value)
  +--- tunnel-name?                    string
  +--- (path-role)?
  |   +---:(primary-path)
  |   |   +--- primary-path-name?      string
  |   +---:(secondary-path)
  |   |   +--- secondary-path-name?    string
  +--- k-requested-paths?              uint8
  +--- protection-type?                identityref
  +--- restoration-type?               identityref
  +--- encoding?                       identityref
  +--- switching-type?                 identityref
  +--- source?                         inet:ip-address
  +--- destination?                    inet:ip-address
  +--- src-tp-id?                      binary
  +--- dst-tp-id?                      binary
  +--- bidirectional?                  boolean
  +--- te-topology-identifier
  |   +--- provider-id?                te-global-id
  |   +--- client-id?                  te-global-id
  |   +--- topology-id?                te-topology-id
+--- association-objects
  +--- association-object* [type id source]
  |   +--- type                        identityref
  |   +--- id                          uint16
  |   +--- source                      te-generic-node-id
  +--- association-object-extended*
  |   [type id source global-source extended-id]
  |   +--- type                        identityref
  |   +--- id                          uint16
  |   +--- source                      te-generic-node-id
  |   +--- global-source                uint32
  |   +--- extended-id                 yang:hex-string
+--- optimizations
  +--- (algorithm)?
  |   +---:(metric) {path-optimization-metric}?
  |   |   +--- optimization-metric* [metric-type]

```



					+-- metric-type	identityref
					+-- weight?	uint8
					+-- explicit-route-exclude-objects	
					+-- route-object-exclude-object* [index]	
					+-- index	uint32
					+-- (type)?	
					+--:(numbered-node-hop)	
					+-- numbered-node-hop	
					+-- node-id	te-node-id
					+-- hop-type?	te-hop-type
					+--:(numbered-link-hop)	
					+-- numbered-link-hop	
					+-- link-tp-id	te-tp-id
					+-- hop-type?	te-hop-type
					+-- direction?	te-link-
direction					+--:(unnumbered-link-hop)	
					+-- unnumbered-link-hop	
					+-- link-tp-id	te-tp-id
					+-- node-id	te-node-id
					+-- hop-type?	te-hop-type
					+-- direction?	te-link-
					+--:(as-number)	
					+-- as-number-hop	
					+-- as-number	inet:as-number
					+-- hop-type?	te-hop-type
					+--:(label)	
					+-- label-hop	
					+-- te-label	
					+-- (technology)?	
					+--:(generic)	
					+-- generic?	
					rt-	
					+-- direction?	te-label-direction
types:generalized-label					+--:(srlg)	
					+-- srlg	

			<pre>             +--- srlg?   uint32 +--- explicit-route-include-objects     +--- route-object-include-object* [index]         +--- index           uint32         +--- (type)?             +---:(numbered-node-hop)                 +--- numbered-node-hop                     +--- node-id       te-node-id                     +--- hop-type?     te-hop-type             +---:(numbered-link-hop)                 +--- numbered-link-hop                     +--- link-tp-id    te-tp-id                     +--- hop-type?     te-hop-type                     +--- direction?    te-link- </pre>
direction			<pre> +---:(unnumbered-link-hop)     +--- unnumbered-link-hop         +--- link-tp-id    te-tp-id         +--- node-id       te-node-id         +--- hop-type?     te-hop-type         +--- direction?    te-link- </pre>
direction			<pre> +---:(as-number)     +--- as-number-hop         +--- as-number     inet:as-number         +--- hop-type?     te-hop-type +---:(label)     +--- label-hop         +--- te-label             +--- (technology)?                 +---:(generic)                     +--- generic?                         rt- </pre>
types:generalized-label			<pre> +--- direction?         te-label-direction +--- tiebreakers     +--- tiebreaker* [tiebreaker-type]         +--- tiebreaker-type  identityref </pre>

```

    +--:(objective-function)
      {path-optimization-objective-function}?
      +-- objective-function
        +-- objective-function-type?  identityref
+-- named-path-constraint?  leafref
  | {te-types:named-path-constraints}?
+-- te-bandwidth
  | +-- (technology)?
  |   +--:(generic)
  |     +-- generic?  te-bandwidth
+-- link-protection?  identityref
+-- setup-priority?  uint8
+-- hold-priority?  uint8
+-- signaling-type?  identityref
+-- path-metric-bounds
  | +-- path-metric-bound* [metric-type]
  |   +-- metric-type  identityref
  |   +-- upper-bound?  uint64
+-- path-affinities-values
  | +-- path-affinities-value* [usage]
  |   +-- usage  identityref
  |   +-- value?  admin-groups
+-- path-affinity-names
  | +-- path-affinity-name* [usage]
  |   +-- usage  identityref
  |   +-- affinity-name* [name]
  |     +-- name  string
+-- path-srlgs-lists
  | +-- path-srlgs-list* [usage]
  |   +-- usage  identityref
  |   +-- values*  srlg
+-- path-srlgs-names
  | +-- path-srlgs-name* [usage]
  |   +-- usage  identityref
  |   +-- names*  string
+-- disjointness?  te-path-
disjointness
  | +-- explicit-route-objects-always
  |   +-- route-object-exclude-always* [index]

```

```

+-- index                               uint32
+-- (type)?
  +--:(numbered-node-hop)
    |   +-- numbered-node-hop
    |       +-- node-id         te-node-id
    |       +-- hop-type?      te-hop-type
  +--:(numbered-link-hop)
    |   +-- numbered-link-hop
    |       +-- link-tp-id      te-tp-id
    |       +-- hop-type?      te-hop-type
    |       +-- direction?     te-link-direction
  +--:(unnumbered-link-hop)
    |   +-- unnumbered-link-hop
    |       +-- link-tp-id      te-tp-id
    |       +-- node-id         te-node-id
    |       +-- hop-type?      te-hop-type
    |       +-- direction?     te-link-direction
  +--:(as-number)
    |   +-- as-number-hop
    |       +-- as-number       inet:as-number
    |       +-- hop-type?      te-hop-type
  +--:(label)
    |   +-- label-hop
    |       +-- te-label
    |           +-- (technology)?
    |               +--:(generic)
    |                   +-- generic?
    |                       rt-types:generalized-label
    |               +-- direction?     te-label-direction
+-- route-object-include-exclude* [index]
+-- explicit-route-usage?               identityref
+-- index                               uint32
+-- (type)?
  +--:(numbered-node-hop)
    |   +-- numbered-node-hop
    |       +-- node-id         te-node-id
    |       +-- hop-type?      te-hop-type
  +--:(numbered-link-hop)
    |   +-- numbered-link-hop

```





```

+-- requested-state!
  +-- timer?          uint16
  +-- transaction-id? string
+-- tunnel-attributes* [tunnel-name]
  +-- tunnel-name      string
  +-- encoding?        identityref
  +-- switching-type?  identityref
  +-- source?          inet:ip-address
  +-- destination?    inet:ip-address
  +-- src-tp-id?       binary
  +-- dst-tp-id?       binary
  +-- bidirectional?   boolean
  +-- association-objects
    +-- association-object* [type id source]
      +-- type          identityref
      +-- id            uint16
      +-- source        te-generic-node-id
    +-- association-object-extended*
      [type id source global-source extended-id]
      +-- type          identityref
      +-- id            uint16
      +-- source        te-generic-node-id
      +-- global-source uint32
      +-- extended-id   yang:hex-string
  +-- protection-type? identityref
  +-- restoration-type? identityref
  +-- te-topology-identifier
    +-- provider-id?   te-global-id
    +-- client-id?     te-global-id
    +-- topology-id?  te-topology-id
  +-- te-bandwidth
    +-- (technology)?
      +--:(generic)
        +-- generic?   te-bandwidth
  +-- link-protection? identityref
  +-- setup-priority?   uint8
  +-- hold-priority?    uint8
  +-- signaling-type?   identityref
  +-- hierarchy

```

```

+-- dependency-tunnels
|   +-- dependency-tunnel* [name]
|   |   +-- name
|   |   |   -> ../../../../tunnels/tunnel/name
|   |   +-- encoding?      identityref
|   |   +-- switching-type? identityref
|   +-- dependency-tunnel-attributes* [name]
|   |   +-- name          leafref
|   |   +-- encoding?    identityref
|   |   +-- switching-type? identityref
+-- hierarchical-link
|   +-- local-te-node-id?      te-types:te-node-id
|   +-- local-te-link-tp-id?   te-types:te-tp-id
|   +-- remote-te-node-id?    te-types:te-node-id
|   +-- te-topology-identifier
|   |   +-- provider-id?   te-global-id
|   |   +-- client-id?    te-global-id
|   |   +-- topology-id?  te-topology-id
+-- synchronization* [synchronization-id]
+-- synchronization-id      uint32
+-- svec
|   +-- relaxable?          boolean
|   +-- disjointness?      te-path-disjointness
|   +-- request-id-number* uint32
+-- svec-constraints
|   +-- path-metric-bound* [metric-type]
|   |   +-- metric-type    identityref
|   |   +-- upper-bound?  uint64
+-- path-srlgs-lists
|   +-- path-srlgs-list* [usage]
|   |   +-- usage          identityref
|   |   +-- values*       srlg
+-- path-srlgs-names
|   +-- path-srlgs-name* [usage]
|   |   +-- usage          identityref
|   |   +-- names*        string
+-- exclude-objects
|   +-- excludes* [index]
|   |   +-- index          uint32

```



```

+-- (type)?
  +--:(numbered-node-hop)
  |   +-- numbered-node-hop
  |       +-- node-id      te-node-id
  |       +-- hop-type?   te-hop-type
  +--:(numbered-link-hop)
  |   +-- numbered-link-hop
  |       +-- link-tp-id   te-tp-id
  |       +-- hop-type?   te-hop-type
  |       +-- direction?  te-link-direction
  +--:(unnumbered-link-hop)
  |   +-- unnumbered-link-hop
  |       +-- link-tp-id   te-tp-id
  |       +-- node-id     te-node-id
  |       +-- hop-type?   te-hop-type
  |       +-- direction?  te-link-direction
  +--:(as-number)
  |   +-- as-number-hop
  |       +-- as-number    inet:as-number
  |       +-- hop-type?   te-hop-type
  +--:(label)
  |   +-- label-hop
  |       +-- te-label
  |           +-- (technology)?
  |               +--:(generic)
  |                   +-- generic?
  |                       rt-types:generalized-label
  |   +-- direction?      te-label-direction
+-- optimizations
  +-- (algorithm)?
  +--:(metric) {te-types:path-optimization-metric}?
  |   +-- optimization-metric* [metric-type]
  |       +-- metric-type      identityref
  |       +-- weight?         uint8
  +--:(objective-function)
  |   {te-types:path-optimization-objective-
function}?
  |   +-- objective-function
  |       +-- objective-function-type?  identityref

```

```

augment /te:tunnels-path-compute/te:output/te:path-compute-result:
  +--ro response* [response-id]
    +--ro response-id                               uint32
  +--ro computed-paths-properties
    +--ro computed-path-properties* [k-index]
      +--ro k-index                                 uint8
      +--ro path-properties
        +--ro path-metric* [metric-type]
          +--ro metric-type                         identityref
          +--ro accumulative-value?                 uint64
        +--ro path-affinities-values
          +--ro path-affinities-value* [usage]
            +--ro usage                             identityref
            +--ro value?                             admin-groups
        +--ro path-affinity-names
          +--ro path-affinity-name* [usage]
            +--ro usage                             identityref
            +--ro affinity-name* [name]
              +--ro name                             string
        +--ro path-srlgs-lists
          +--ro path-srlgs-list* [usage]
            +--ro usage                             identityref
            +--ro values*                             srlg
        +--ro path-srlgs-names
          +--ro path-srlgs-name* [usage]
            +--ro usage                             identityref
            +--ro names*                             string
        +--ro path-route-objects
          +--ro path-route-object* [index]
            +--ro index                               uint32
            +--ro (type)?
              +--:(numbered-node-hop)
                +--ro numbered-node-hop
                  +--ro node-id                       te-node-id
                  +--ro hop-type?                     te-hop-type
              +--:(numbered-link-hop)
                +--ro numbered-link-hop
                  +--ro link-tp-id                     te-tp-id
                  +--ro hop-type?                     te-hop-type

```

```

|
|
|
|      +---ro direction?      te-link-direction
+---: (unnumbered-link-hop)
|      +---ro unnumbered-link-hop
|      +---ro link-tp-id      te-tp-id
|      +---ro node-id        te-node-id
|      +---ro hop-type?      te-hop-type
|      +---ro direction?      te-link-direction
+---: (as-number)
|      +---ro as-number-hop
|      +---ro as-number      inet:as-number
|      +---ro hop-type?      te-hop-type
+---: (label)
|      +---ro label-hop
|      +---ro te-label
|      +---ro (technology)?
|      |      +---: (generic)
|      |      +---ro generic?
|      |      rt-
types:generalized-label
|
|      +---ro direction?
|      |      te-label-direction
+---ro te-bandwidth
|      +---ro (technology)?
|      |      +---: (generic)
|      |      +---ro generic?      te-bandwidth
+---ro disjointness-type?
|      |      te-types:te-path-disjointness
+---ro shared-resources-tunnels
|      |      +---ro lsp-shared-resources-tunnel*      tunnel-ref
+---ro computed-path-error-infos
|      +---ro computed-path-error-info* []
|      +---ro error-description?      string
|      +---ro error-timestamp?        yang:date-and-time
|      +---ro error-reason?            identityref
+---ro tunnel-ref?                      te:tunnel-ref
+---ro (path)?
|      +---: (primary)
|      |      +---ro primary-path-ref?            leafref
|      +---: (primary-reverse)

```

```

    | +--ro primary-reverse-path-ref?    leafref
    +--:(secondary)
    | +--ro secondary-path-ref?         leafref
    +--:(secondary-reverse)
      +--ro secondary-reverse-path-ref? leafref
augment /te:tunnels-actions/te:input/te:tunnel-info/te:filter-
type:
  +--:(path-compute-transactions)
    +-- path-compute-transaction-id*   string
augment /te:tunnels-actions/te:output:
  +--ro path-computed-delete-result
    +--ro path-compute-transaction-id* string

```

Figure 11 - TE path computation YANG tree

## 6.2. YANG Module

```

<CODE BEGINS>file "ietf-te-path-computation@2020-07-10.yang"
module ietf-te-path-computation {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-te-path-computation";

  prefix "te-pc";

  import ietf-inet-types {
    prefix "inet";
    reference
      "RFC6991: Common YANG Data Types";
  }

  import ietf-te {
    prefix "te";
    reference
      "RFCXXXX: A YANG Data Model for Traffic Engineering Tunnels
      and Interfaces";
  }

  /* Note: The RFC Editor will replace XXXX with the number assigned
  to the RFC once draft-ietf-teas-yang-te becomes an RFC.*/

```

```
import ietf-te-types {  
  prefix "te-types";  
  reference  
    "RFC8776: Common YANG Data Types for Traffic Engineering."  
}
```

```
organization  
  "Traffic Engineering Architecture and Signaling (TEAS)  
  Working Group";
```

```
contact  
  "WG Web: <http://tools.ietf.org/wg/teas/>  
  WG List: <mailto:teas@ietf.org>  
  
  Editor: Italo Busi  
    <mailto:italo.busi@huawei.com>  
  
  Editor: Sergio Belotti  
    <mailto:sergio.belotti@nokia.com>  
  
  Editor: Victor Lopez  
    <mailto:victor.lopezalvarez@telefonica.com>  
  
  Editor: Oscar Gonzalez de Dios  
    <mailto:oscar.gonzalezdedios@telefonica.com>  
  
  Editor: Anurag Sharma  
    <mailto:ansha@google.com>  
  
  Editor: Yan Shi  
    <mailto:shiyang49@chinaunicom.cn>  
  
  Editor: Ricard Vilalta  
    <mailto:ricard.vilalta@cttc.es>  
  
  Editor: Karthik Sethuraman  
    <mailto:karthik.sethuraman@necam.com>  
  
  Editor: Michael Scharf
```

<mailto:michael.scharf@gmail.com>

Editor: Daniele Ceccarelli  
<mailto:daniele.ceccarelli@ericsson.com>

";

description

"This module defines a YANG data model for requesting Traffic Engineering (TE) path computation. The YANG model defined in this document is based on RPCs augmenting the RPCs defined in the generic TE module (ietf-te).

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

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.";

// RFC Ed.: replace XXXX with actual RFC number and remove  
// this note

// replace the revision date with the module publication date  
// the format is (year-month-day)

```
revision "2020-07-10" {  
  description  
    "Initial revision";  
  reference
```

```
    "RFC XXXX: Yang model for requesting Path Computation";
}

// RFC Ed.: replace XXXX with actual RFC number and remove
// this note

/*
 * Identities
 */

identity svec-metric-type {
  description
    "Base identity for SVEC metric type.";
  reference
    "RFC5541: Encoding of Objective Functions in the Path
    Computation Element Communication Protocol (PCEP).";
}

identity svec-metric-cumul-te {
  base svec-metric-type;
  description
    "Cumulative TE cost.";
  reference
    "RFC5541: Encoding of Objective Functions in the Path
    Computation Element Communication Protocol (PCEP).";
}

identity svec-metric-cumul-igp {
  base svec-metric-type;
  description
    "Cumulative IGP cost.";
  reference
    "RFC5541: Encoding of Objective Functions in the Path
    Computation Element Communication Protocol (PCEP).";
}

identity svec-metric-cumul-hop {
  base svec-metric-type;
  description
```

```
    "Hop cumulative path metric.";
}

identity svec-metric-aggregate-bandwidth-consumption {
  base svec-metric-type;
  description
    "Aggregate bandwidth consumption.";
  reference
    "RFC5541: Encoding of Objective Functions in the Path
    Computation Element Communication Protocol (PCEP).";
}

identity svec-metric-load-of-the-most-loaded-link {
  base svec-metric-type;
  description
    "Load of the most loaded link.";
  reference
    "RFC5541: Encoding of Objective Functions in the Path
    Computation Element Communication Protocol (PCEP).";
}

identity tunnel-action-path-compute-delete {
  base te:tunnel-actions-type;
  description
    "Action type to delete the transient states
    of computed paths, as described in section 3.3.1.";
}

/*
 * Groupings
 */

grouping protection-restoration-properties {
  description
    "This grouping defines the restoration and protection types
    for a path in the path computation request.";
  leaf protection-type {
    type identityref {
      base te-types:lsp-protection-type;
    }
  }
}
```



```
    }
    default te-types:lsp-protection-unprotected;
    description "LSP protection type.";
  }
  leaf restoration-type {
    type identityref {
      base te-types:lsp-restoration-type;
    }
    default te-types:lsp-restoration-restore-any;
    description "LSP restoration type.";
  }
} // grouping protection-restoration-properties

grouping requested-info {
  description
    "This grouping defines the information (e.g., metrics)
    which is requested, in the path computation request, to be
    returned in the path computation response.";
  list requested-metrics {
    key 'metric-type';
    description
      "The list of the requested metrics.
      The metrics listed here must be returned in the response.
      Returning other metrics in the response is optional.";
    leaf metric-type {
      type identityref {
        base te-types:path-metric-type;
      }
      description
        "The metric that must be returned in the response";
    }
  }
}
leaf return-srlgs {
  type boolean;
  default false;
  description
    "If true, path srlgs must be returned in the response.
    If false, returning path srlgs in the response optional.";
}
```

```
leaf return-affinities {
  type boolean;
  default false;
  description
    "If true, path affinities must be returned in the response.
     If false, returning path affinities in the response is
     optional.";
}
} // grouping requested-info

grouping requested-state {
  description
    "Configuration for the transient state used
     to report the computed path";
  leaf timer {
    type uint16;
    units minutes;
    default 10;
    description
      "The timeout after which the transient state reporting
       the computed path should be removed.";
  }
  leaf transaction-id {
    type string;
    description
      "The transaction-id associated with this path computation
       to be used for fast deletion of the transient states
       associated with multiple path computations.

       This transaction-id can be used to explicitly delete all
       the transient states of all the computed paths associated
       with the same transaction-id.

       When one path associated with a transaction-id is setup,
       the transient states of all the other computed paths
       with the same transaction-id are automatically removed.

       If not specified, the transient state is removed only
       when the timer expires (when the timer is specified)";
  }
}
```

```
        or not created at all (stateless path computation,
        when the timer is not specified).";
    }
} // grouping requested-state

grouping reported-state {
  description
    "This grouping defines the information, returned in the path
    computation response, reporting the transient state related
    to the computed path";

  leaf tunnel-ref {
    type te:tunnel-ref;
    description
      "
      Reference to the tunnel that reports the transient state
      of the computed path.

      If no transient state is created, this attribute is
      omitted.
      ";
  }
}
choice path {
  description
    "The transient state of the computed path can be reported
    as a primary, primary-reverse, secondary or
    a secondary-reverse path of a te-tunnel";
  case primary {
    leaf primary-path-ref {
      type leafref {
        path "/te:te/te:tunnels/" +
          "te:tunnel[te:name=current()/../tunnel-ref]/" +
          "te:primary-paths/te:primary-path/" +
          "te:name";
      }
      must "../tunnel-ref" {
        description
          "The primary-path name can only be reported
          if also the tunnel name is reported.";
      }
    }
  }
}
```

```
    }
    description
    "
        Reference to the primary-path that reports
        the transient state of the computed path.

        If no transient state is created,
        this attribute is omitted.
    ";
}
} // case primary
case primary-reverse {
    leaf primary-reverse-path-ref {
        type leafref {
            path "/te:te/te:tunnels/" +
                "te:tunnel[te:name=current()/../tunnel-ref]/" +
                "te:primary-paths/te:primary-path/" +
                "te:name";
        }
        must "../tunnel-ref" {
            description
            "The primary-reverse-path name can only be reported
            if also the tunnel name is reported.";
        }
    }
    description
    "
        Reference to the primary-reverse-path that reports
        the transient state of the computed path.

        If no transient state is created,
        this attribute is omitted.
    ";
}
} // case primary-reverse
case secondary {
    leaf secondary-path-ref {
        type leafref {
            path "/te:te/te:tunnels/" +
                "te:tunnel[te:name=current()/../tunnel-ref]/" +
```

```
        "te:secondary-paths/te:secondary-path/" +
        "te:name";
    }
    must "../tunnel-ref" {
        description
            "The secondary-path name can only be reported
            if also the tunnel name is reported.";
    }
    description
        "
        Reference to the secondary-path that reports
        the transient state of the computed path.

        If no transient state is created,
        this attribute is omitted.
        ";
    }
} // case secondary
case secondary-reverse {
    leaf secondary-reverse-path-ref {
        type leafref {
            path "/te:te/te:tunnels/" +
                "te:tunnel[te:name=current()../tunnel-ref]/" +
                "te:secondary-reverse-paths/" +
                "te:secondary-reverse-path/te:name";
        }
    }
    must "../tunnel-ref" {
        description
            "The secondary-reverse-path name can only be reported
            if also the tunnel name is reported.";
    }
    description
        "
        Reference to the secondary-reverse-path that reports
        the transient state of the computed path.

        If no transient state is created,
        this attribute is omitted.
        ";
    }
}
```

```
    }
  } // case secondary
} // choice path
} // grouping reported-state

grouping svec-metrics-bounds {
  description
    "This grouping defines the bounds for the SVEC metrics used
    by a set of synchronized path computation requests.";
  leaf metric-type {
    type identityref {
      base svec-metric-type;
    }
    description
      "SVEC metric type.";
    reference
      "RFC5541: Encoding of Objective Functions in the Path
      Computation Element Communication Protocol (PCEP).";
  }
  leaf upper-bound {
    type uint64;
    description "Upper bound on SVEC metric";
  }
} // grouping svec-metrics-bounds

grouping svec-metrics-optimization {
  description
    "TE path metric bounds grouping for computing a set of
    synchronized requests";

  leaf metric-type {
    type identityref {
      base svec-metric-type;
    }
    description "TE path metric type usable for computing a set of
    synchronized requests";
  }
  leaf weight {
    type uint8;
  }
}
```

```
        description "Metric normalization weight";
    }
} // grouping svec-metrics-optimization

grouping svec-exclude {
    description "List of resources to be excluded by all the paths
    in the SVEC";
    container exclude-objects {
        description "resources to be excluded";
        list excludes {
            key index;
            ordered-by user;
            leaf index {
                type uint32;
                description "XRO subobject index";
            }
            description
                "List of explicit route objects to always exclude
                from synchronized path computation";
            uses te-types:explicit-route-hop;
        }
    }
} // grouping svec-exclude

grouping synchronization-constraints {
    description
        "Global constraints applicable to synchronized path
        computation requests.";
    container svec-constraints {
        description "global svec constraints";
        list path-metric-bound {
            key metric-type;
            description "list of bound metrics";
            uses svec-metrics-bounds;
        }
    }
    uses te-types:generic-path-srlgs;
    uses svec-exclude;
} // grouping synchronization-constraints
```

```
grouping synchronization-optimization {
  description
    "Optimizations applicable to synchronized path
    computation requests.";
  container optimizations {
    description
      "The objective function container that includes attributes
      to impose when computing a synchronized set of paths";

    choice algorithm {
      description "Optimizations algorithm.";
      case metric {
        if-feature te-types:path-optimization-metric;
        list optimization-metric {
          key "metric-type";
          description "svec path metric type";
          uses svec-metrics-optimization;
        }
      }
      case objective-function {
        if-feature te-types:path-optimization-objective-function;
        container objective-function {
          description
            "The objective function container that includes
            attributes to impose when computing a TE path";
          leaf objective-function-type {
            type identityref {
              base te-types:objective-function-type;
            }
            default te-types:of-minimize-cost-path;
            description "Objective function entry";
          }
        }
      }
    }
  }
} // grouping synchronization-optimization
```



```
grouping synchronization-info {
  description "Information for sync";
  list synchronization {
    key "synchronization-id";
    description "sync list";
    leaf synchronization-id {
      type uint32;
      description "index";
    }
  }
  container svec {
    description
      "Synchronization VECtor";
    leaf relaxable {
      type boolean;
      default true;
      description
        "If this leaf is true, path computation process is
         free to ignore svec content.
         Otherwise, it must take into account this svec.";
    }
    uses te-types:generic-path-disjointness;
    leaf-list request-id-number {
      type uint32;
      description
        "This list reports the set of path computation
         requests that must be synchronized.";
    }
  }
  uses synchronization-constraints;
  uses synchronization-optimization;
} // grouping synchronization-info

grouping encoding-and-switching-type {
  description
    "Common grouping to define the LSP encoding and
     switching types";

  leaf encoding {
```

```
    type identityref {
      base te-types:lsp-encoding-types;
    }
    description "LSP encoding type";
    reference "RFC3945";
  }
  leaf switching-type {
    type identityref {
      base te-types:switching-capabilities;
    }
    description "LSP switching type";
    reference "RFC3945";
  }
}

grouping tunnel-common-attributes {
  description
    "Common grouping to define the TE tunnel parameters";

  uses encoding-and-switching-type;
  leaf source {
    type inet:ip-address;
    description "TE tunnel source address.";
  }
  leaf destination {
    type inet:ip-address;
    description "te-tunnel destination address";
  }
  leaf src-tp-id {
    type binary;
    description
      "TE tunnel source termination point identifier.";
  }
  leaf dst-tp-id {
    type binary;
    description
      "TE tunnel destination termination point identifier.";
  }
  leaf bidirectional {
```

```
        type boolean;
        default 'false';
        description "TE tunnel bidirectional";
    }
}

/*
 * Augment TE RPCs
 */

augment "/te:tunnels-path-compute/te:input/te:path-compute-info" {
    description "Path Computation RPC input";
    list path-request {
        key "request-id";
        description "The list of the requested paths to be computed";
        leaf request-id {
            type uint32;
            mandatory true;
            description
                "Each path computation request is uniquely identified
                within the RPC request by the request-id-number.";
        }
    }

    choice tunnel-attributes {
        default value;
        description
            "Whether the tunnel attributes are specified by value
            within this path computation request or by reference.
            The reference could be either to an existing te-tunnel
            or to an entry in the tunnel-attributes list";
        case reference {
            choice tunnel-exist {
                description
                    "Whether the tunnel reference is to an existing
                    te-tunnel or to an entry in the tunnel-attributes
                    list";
                case tunnel-ref {
                    leaf tunnel-ref {
                        type te:tunnel-ref;
                    }
                }
            }
        }
    }
}
```

```
        mandatory true;
        description "The referenced te-tunnel instance";
    }
} // case tunnel-ref
case tunnel-attributes-ref {
    leaf tunnel-attributes-ref {
        type leafref {
            path "/te:tunnels-path-compute/te:input/"
                + "te:path-compute-info/"
                + "te-pc:tunnel-attributes/te-pc:tunnel-name";
        }
    }
}
/*
 * Open issue: RPC path

    leaf tunnel-attributes-ref {
        type leafref {
            path "/te:tunnels-path-compute/"
                + "te:path-compute-info/"
                + "te-pc:tunnel-attributes/te-pc:tunnel-name";
        }
    }
*/

        mandatory true;
        description "The referenced te-tunnel instance";
    }
} // case tunnel-attributes-ref
} // choice tunnel-exist
leaf path-name {
    type string;
    description "TE path name.";
}
choice tunnel-path-role {
    mandatory true;
    description
        "Whether this path is a primary, or a reverse primary,
        or a secondary, or a reverse secondary path";
    case primary-path {
        container primary-path {
            presence
                "Indicates that the requested path
```

```
        is a primary path";
    description "TE primary path";
    uses te:path-preference;
    uses te:k-requested-paths;
} // container primary-path
} // case primary-path
case secondary-path {
    container secondary-path {
        description "TE secondary path";
        uses te:path-preference;
        uses protection-restoration-properties;
        list primary-path-ref {
            key index;
            min-elements 1;
            description
                "The list of primary paths that reference
                 this path as a candidate secondary path";
            leaf index {
                type uint32;
                description
                    "The index used by the
                     primary-path-ref list";
            }
            choice primary-path-exist {
                description
                    "Whether the path reference is to an existing
                     te-tunnel path or to another path request";
                case path-ref {
                    leaf primary-path-ref {
                        type leafref {
                            path "/te:te/te:tunnels/te:tunnel[te:name"
                                + "=current()/../../../../tunnel-ref]/"
                                + "te:primary-paths/te:primary-path/"
                                + "te:name";
                        }
                    }
                    must "../../../../tunnel-ref" {
                        description
                            "The primary-path can be referenced
                             if also the tunnel is referenced.";
                    }
                }
            }
        }
    }
}
```

```

    }
    mandatory true;
    description "The referenced primary path";
  }
} // case path-ref
case path-request-ref {
  leaf path-request-ref {
    type leafref {
      path "/te:tunnels-path-compute/te:input/"
        + "te:path-compute-info/"
        + "te-pc:path-request/"
        + "te-pc:request-id";
    }
  }
}
/*
 * Open issue: RPC path

  leaf path-request-ref {
    type leafref {
      path "/te:tunnels-path-compute/"
        + "te:path-compute-info/"
        + "te-pc:path-request/"
        + "te-pc:request-id";
    }
  }
*/

  mandatory true;
  description
    "The referenced primary path request";
}
} // case path-request-ref
} // choice primary-path-exist
} // list primary-path-ref
} // container secondary-path
} // case secondary-path
case primary-reverse-path {
  container primary-reverse-path {
    description "TE primary reverse path";
    choice primary-path-exist {
      description
        "Whether the path reference to the primary paths

```

```
for which this path is the reverse-path is to
an existing te-tunnel path or to another path
request";
case path-ref {
  leaf primary-path-ref {
    type leafref {
      path "/te:te/te:tunnels/te:tunnel[te:name"
        + "=current()/../../../../tunnel-ref]/"
        + "te:primary-paths/te:primary-path/"
        + "te:name";
    }
    must "../../../../tunnel-ref" {
      description
        "The primary-path can be referenced
        if also the tunnel is referenced.";
    }
    mandatory true;
    description "The referenced primary path";
  }
} // case path-ref
case path-request-ref {
  leaf path-request-ref {
    type leafref {
      path "/te:tunnels-path-compute/te:input/"
        + "te:path-compute-info/"
        + "te-pc:path-request/"
        + "te-pc:request-id";
    }
  }
}
/*
 * Open issue: RPC path

  leaf path-request-ref {
    type leafref {
      path "/te:tunnels-path-compute/"
        + "te:path-compute-info/"
        + "te-pc:path-request/"
        + "te-pc:request-id";
    }
  }
*/
```

```
        mandatory true;
        description
            "The referenced primary path request";
    }
} // case path-request-ref
} // choice primary-path-exist
} // container primary-reverse-path
} // case primary-reverse-path
case secondary-reverse-path {
    container secondary-reverse-path {
        description "TE secondary reverse path";
        uses te:path-preference;
        uses protection-restoration-properties;
        list primary-reverse-path-ref {
            key index;
            min-elements 1;
            description
                "The list of primary reverse paths that
                 reference this path as a candidate
                 secondary reverse path";
            leaf index {
                type uint32;
                description
                    "The index used by the
                     primary-reverse-path-ref list";
            }
        }
        choice primary-reverse-path-exist {
            description
                "Whether the path reference is to an existing
                 te-tunnel path or to another path request";
            case path-ref {
                leaf primary-path-ref {
                    type leafref {
                        path "/te:te/te:tunnels/te:tunnel[te:name"
                            + "=current()/../../../../tunnel-ref]/"
                            + "te:primary-paths/te:primary-path/"
                            + "te:name";
                    }
                }
                must "../../../../tunnel-ref" {

```



```

        description
            "The primary-path can be referenced
            if also the tunnel is referenced.";
        }
        mandatory true;
        description
            "The referenced primary path";
        }
    } // case path-ref
    case path-request-ref {
        leaf path-request-ref {
            type leafref {
                path "/te:tunnels-path-compute/te:input/"
                    + "te:path-compute-info/"
                    + "te-pc:path-request/"
                    + "te-pc:request-id";
            }
        }
    }
/*
 * Open issue: RPC path

        leaf path-request-ref {
            type leafref {
                path "/te:tunnels-path-compute/"
                    + "te:path-compute-info/"
                    + "te-pc:path-request/"
                    + "te-pc:request-id";
            }
        }
*/

        mandatory true;
        description
            "The referenced primary reverse path
            request";
        }
    } // case path-request-ref
    } // choice primary-reverse-path-exist
} // list primary-reverse-path-ref
} // container secondary-reverse-path
} // case secondary-reverse-path
} // choice tunnel-path-role

```

```
    } // case reference
  case value {
    leaf tunnel-name {
      type string;
      description "TE tunnel name.";
    }
    choice path-role {
      default primary-path;
      description
        "Whether this path is a primary or a secondary path";
      case primary-path {
        leaf primary-path-name {
          type string;
          description "TE path name.";
        }
      } // case primary-path
      case secondary-path {
        leaf secondary-path-name {
          type string;
          description "TE path name.";
        }
      } // case secondary-path
    } // choice path-role
  }
}
/*
 * Open issue: should protection-restoration-properties be moved
 *              under secondary-path?
 */
  uses te:k-requested-paths;
  uses protection-restoration-properties;
  uses tunnel-common-attributes;
  uses te-types:te-topology-identifier;
} // case value
} // choice tunnel-attributes
uses te:path-compute-info;
uses requested-info;
container requested-state {
  presence
    "Request temporary reporting of the computed path state";
  description
```

```
        "Configures attributes for the temporary reporting of the
        computed path state (e.g., expiration timer).";
    uses requested-state;
} // container requested-state
} // list path-request
list tunnel-attributes {
    key "tunnel-name";
    description
        "Tunnel attributes common to multiple request paths";
    leaf tunnel-name {
        type string;
        description "TE tunnel name.";
    }
    uses tunnel-common-attributes;
    uses te:tunnel-associations-properties;
    uses protection-restoration-properties;
    uses te-types:tunnel-constraints;
    uses te:tunnel-hierarchy-properties {
        augment "hierarchy/dependency-tunnels" {
            description
                "Augment with the list of dependency tunnel requests.";
            list dependency-tunnel-attributes {
                key "name";
                description
                    "A tunnel request entry that this tunnel request can
                    potentially depend on.";
                leaf name {
                    type leafref {
                        path "/te:tunnels-path-compute/te:input/"
                            + "te:path-compute-info/te-pc:tunnel-attributes/"
                            + "te-pc:tunnel-name";
                    }
                }
            }
        }
    }
}
/*
* Open issue: RPC path

    leaf name {
        type leafref {
            path "/te:tunnels-path-compute/"
                + "te:path-compute-info/te-pc:tunnel-attributes/"
```

```
        + "te-pc:tunnel-name";
    }
*/
    description
        "Dependency tunnel request name.";
    }
    uses encoding-and-switching-type;
} // list dependency-tunnel-request
}
} // list tunnel-attributes
uses synchronization-info;
} // path-compute rpc input

augment "/te:tunnels-path-compute/te:output/"
    + "te:path-compute-result" {
    description "Path Computation RPC output";
    list response {
        key "response-id";
        config false;
        description "response";
        leaf response-id {
            type uint32;
            description
                "The response-id has the same value of the
                corresponding request-id.";
        }
        uses te:path-computation-response;
        uses reported-state;
    }
} // path-compute rpc output

augment "/te:tunnels-actions/te:input/te:tunnel-info/"
    + "te:filter-type" {
    description "Augment Tunnels Action RPC input filter types";

    case path-compute-transactions {
        when "derived-from-or-self(..te:action-info/te:action, "
            + "'tunnel-action-path-compute-delete')";
    }
}
```

```
    description "Path Delete Action RPC";
    leaf-list path-compute-transaction-id {
      type string;
      description
        "The list of the transaction-id values of the
         transient states to be deleted";
    }
  } // case path-compute-transactions
} // path-delete rpc input

augment "/te:tunnels-actions/te:output" {
  description
    "Augment Tunnels Action RPC input with path delete result";

  container path-computed-delete-result {
/*
* Open issue: RPC path
*/
    when "derived-from-or-self(..../te:input/te:action-info/"
      + "te:action, 'tunnel-action-path-compute-delete')";
    description "Path Delete RPC output";
    leaf-list path-compute-transaction-id {
      type string;
      description
        "The list of the transaction-id values of the
         transient states that have been successfully deleted";
    }
  } // container path-computed-delete-result
} // path-delete rpc output
}
<CODE ENDS>
```

Figure 12 - TE path computation YANG module

## 7. Security Considerations

This document describes use cases of requesting Path Computation using YANG models, which could be used at the ABNO Control Interface [RFC7491] and/or between controllers in ACTN [RFC8453]. As such, it

does not introduce any new security considerations compared to the ones related to YANG specification, ABNO specification and ACTN Framework defined in [RFC7950], [RFC7491] and [RFC8453].

The YANG module defined in this draft is designed to be accessed via the NETCONF protocol [RFC6241] or RESTCONF protocol [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].

This document also defines common data types using the YANG data modeling language. The definitions themselves have no security impact on the Internet, but the usage of these definitions in concrete YANG modules might have. The security considerations spelled out in the YANG specification [RFC7950] apply for this document as well.

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.

Note - The security analysis of each leaf is for further study.

## 8. IANA Considerations

This document registers the following URIs in the "ns" subregistry within the "IETF XML registry" [RFC3688].

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

This document registers a YANG module in the "YANG Module Names" registry [RFC7950].

name: ietf-te-path-computation  
namespace: urn:ietf:params:xml:ns:yang:ietf-te-path-computation  
prefix: te-pc  
reference: this document

## 9. References

### 9.1. Normative References

- [RFC3688] Mealling, M., "The IETF XML Registry", RFC 3688, January 2004.
- [RFC3945] Mannie, E., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", RFC 3945, DOI 10.17487/RFC3945, October 2004, <<https://www.rfc-editor.org/info/rfc3945>>.
- [RFC5440] Vasseur, JP., Le Roux, JL. et al., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.
- [RFC5441] Vasseur, JP., Ed., Zhang, R., Bitar, N., and JL. Le Roux, "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths", RFC 5441, DOI 10.17487/RFC5441, April 2009, <<https://www.rfc-editor.org/info/rfc5441>>.
- [RFC5541] Le Roux, JL. et al., "Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)", RFC 5541, June 2009.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, June 2011.
- [RFC6991] Schoenwaelder, J., "Common YANG Data Types", RFC 6991, July 2013.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, January 2017.
- [RFC8341] Bierman, A., and M. Bjorklund, "Network Configuration Access Control Model", RFC 8341, March 2018.

- [RFC7926] Farrel, A. et al., "Problem Statement and Architecture for Information Exchange Between Interconnected Traffic Engineered Networks", RFC 7926, July 2016.
- [RFC7950] Bjorklund, M., "The YANG 1.1 Data Modeling Language", RFC 7950, August 2016.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, January 2017.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, March 2018.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, August 2018.
- [RFC8776] Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Common YANG Data Types for Traffic Engineering", RFC8776, June 2020.
- [TE-TOPO] Liu, X. et al., "YANG Data Model for TE Topologies", draft-ietf-teas-yang-te-topo, work in progress.
- [TE-TUNNEL] Saad, T. et al., "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te, work in progress.

## 9.2. Informative References

- [RFC4655] Farrel, A. et al., "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.
- [RFC6805] King, D., Ed. and A. Farrel, Ed., "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, DOI 10.17487/RFC6805, November 2012, <<https://www.rfc-editor.org/info/rfc6805>>.
- [RFC7139] Zhang, F. et al., "GMPLS Signaling Extensions for Control of Evolving G.709 Optical Transport Networks", RFC 7139, March 2014.
- [RFC7446] Lee, Y. et al., "Routing and Wavelength Assignment Information Model for Wavelength Switched Optical Networks", RFC 7446, February 2015.



- [RFC7491] Farrel, A., King, D., "A PCE-Based Architecture for Application-Based Network Operations", RFC 7491, March 2015.
- [RFC8233] Dhody, D. et al., "Extensions to the Path Computation Element Communication Protocol (PCEP) to Compute Service-Aware Label Switched Paths (LSPs)", RFC 8233, September 2017
- [RFC8342] Bjorklund, M. et al. "Network Management Datastore Architecture (NMDA)", RFC 8342, March 2018
- [RFC8453] Ceccarelli, D., Lee, Y. et al., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC8453, August 2018.
- [RFC8454] Lee, Y. et al., "Information Model for Abstraction and Control of TE Networks (ACTN)", RFC8454, September 2018.
- [OTN-TOPO] Zheng, H. et al., "A YANG Data Model for Optical Transport Network Topology", draft-ietf-ccamp-otn-topo-yang, work in progress.
- [ITU-T G.709-2016] ITU-T Recommendation G.709 (06/16), "Interface for the optical transport network", June 2016.

## Appendix A. Examples of dimensioning the "detailed connectivity matrix"

In the following table, a list of the possible constraints, associated with their potential cardinality, is reported.

The maximum number of potential connections to be computed and reported is, in first approximation, the multiplication of all of them.

Constraint	Cardinality
End points	$N(N-1)/2$ if connections are bidirectional (OTN and WDM), $N(N-1)$ for unidirectional connections.
Bandwidth	In WDM networks, bandwidth values are expressed in GHz.  On fixed-grid WDM networks, the central frequencies are on a 50GHz grid and the channel width of the transmitters are typically 50GHz such that each central frequency can be used, i.e., adjacent channels can be placed next to each other in terms of central frequencies.  On flex-grid WDM networks, the central frequencies are on a 6.25GHz grid and the channel width of the transmitters can be multiples of 12.5GHz.  For fixed-grid WDM networks typically there is only one possible bandwidth value (i.e., 50GHz) while for flex-grid WDM networks typically there are 4 possible bandwidth values (e.g., 37.5GHz, 50GHz, 62.5GHz, 75GHz).  In OTN (ODU) networks, bandwidth values are expressed as pairs of ODU type and, in case of ODUflex, ODU rate in bytes/sec as described in section 5 of [RFC7139].  For "fixed" ODUk types, 6 possible bandwidth values are possible (i.e., ODU0, ODU1, ODU2, ODU2e, ODU3, ODU4).  For ODUflex(GFP), up to 80 different bandwidth values can be specified, as defined in Table 7-8 of [ITU-T G.709-2016].  For other ODUflex types, like ODUflex(CBR), the number of possible bandwidth values depends on the rates of the

clients that could be mapped over these ODUflex types, as shown in Table 7.2 of [ITU-T G.709-2016], which in theory could be a continuum of values. However, since different ODUflex bandwidths that use the same number of TSs on each link along the path are equivalent for path computation purposes, up to 120 different bandwidth ranges can be specified.

Ideas to reduce the number of ODUflex bandwidth values in the detailed connectivity matrix, to less than 100, are for further study.

Bandwidth specification for ODUCn is currently for further study but it is expected that other bandwidth values can be specified as integer multiples of 100Gb/s.

In IP we have bandwidth values in bytes/sec. In principle, this is a continuum of values, but in practice we can identify a set of bandwidth ranges, where any bandwidth value inside the same range produces the same path.

The number of such ranges is the cardinality, which depends on the topology, available bandwidth and status of the network. Simulations (Note: reference paper submitted for publication) show that values for medium size topologies (around 50-150 nodes) are in the range 4-7 (5 on average) for each end points couple.

**Metrics** IGP, TE and hop number are the basic objective metrics defined so far. There are also the 2 objective functions defined in [RFC5541]: Minimum Load Path (MLP) and Maximum Residual Bandwidth Path (MBP). Assuming that one only metric or objective function can be optimized at once, the total cardinality here is 5.

With [RFC8233], a number of additional metrics are defined, including Path Delay metric, Path Delay Variation metric and Path Loss metric, both for point-to-point and point-to-multipoint paths. This increases the cardinality to 8.

**Bounds** Each metric can be associated with a bound in order to find a path having a total value of that metric lower than the given bound. This has a potentially very high cardinality (as any value for the bound is allowed). In

practice there is a maximum value of the bound (the one with the maximum value of the associated metric) which results always in the same path, and a range approach like for bandwidth in IP should produce also in this case the cardinality. Assuming to have a cardinality similar to the one of the bandwidth (let say 5 on average) we should have 6 (IGP, TE, hop, path delay, path delay variation and path loss; we don't consider here the two objective functions of [RFC5541] as they are conceived only for optimization)\*5 = 30 cardinality.

#### Technology

constraints For further study

Priority We have 8 values for setup priority, which is used in path computation to route a path using free resources and, where no free resources are available, resources used by LSPs having a lower holding priority.

Local prot It's possible to ask for a local protected service, where all the links used by the path are protected with fast reroute (this is only for IP networks, but line protection schemas are available on the other technologies as well). This adds an alternative path computation, so the cardinality of this constraint is 2.

#### Administrative

Colors Administrative colors (aka affinities) are typically assigned to links but when topology abstraction is used affinity information can also appear in the detailed connectivity matrix.

There are 32 bits available for the affinities. Links can be tagged with any combination of these bits, and path computation can be constrained to include or exclude any or all of them. The relevant cardinality is 3 (include-any, exclude-any, include-all) times  $2^{32}$  possible values. However, the number of possible values used in real networks is quite small.

#### Included Resources

A path computation request can be associated to an ordered set of network resources (links, nodes) to be included along the computed path. This constraint would

have a huge cardinality as in principle any combination of network resources is possible. However, as far as the Orchestrator doesn't know details about the internal topology of the domain, it shouldn't include this type of constraint at all (see more details below).

#### Excluded Resources

A path computation request can be associated to a set of network resources (links, nodes, SRLGs) to be excluded from the computed path. Like for included resources, this constraint has a potentially very high cardinality, but, once again, it can't be actually used by the Orchestrator, if it's not aware of the domain topology (see more details below).

As discussed above, the Orchestrator can specify include or exclude resources depending on the abstract topology information that the domain controller exposes:

- o In case the domain controller exposes the entire domain as a single abstract TE node with his own external terminations and detailed connectivity matrix (whose size we are estimating), no other topological details are available, therefore the size of the detailed connectivity matrix only depends on the combination of the constraints that the Orchestrator can use in a path computation request to the domain controller. These constraints cannot refer to any details of the internal topology of the domain, as those details are not known to the Orchestrator and so they do not impact size of the detailed connectivity matrix exported.

- o Instead in case the domain controller exposes a topology including more than one abstract TE nodes and TE links, and their attributes (e.g. SRLGs, affinities for the links), the Orchestrator knows these details and therefore could compute a path across the domain referring to them in the constraints. The detailed connectivity matrixes, whose size need to be estimated here, are the ones relevant to the abstract TE nodes exported to the Orchestrator. These detailed connectivity matrixes and therefore their sizes, while cannot depend on the other abstract TE nodes and TE links, which are external to the given abstract node, could depend to SRLGs (and other attributes, like affinities) which could be present also in the portion of the topology represented by the abstract nodes, and therefore contribute to the size of the related detailed connectivity matrix.

We also don't consider here the possibility to ask for more than one path in diversity or for point-to-multi-point paths, which are for further study.

Considering for example an IP domain without considering SRLG and affinities, we have an estimated number of paths depending on these estimated cardinalities:

Endpoints =  $N*(N-1)$ , Bandwidth = 5, Metrics = 6, Bounds = 20,  
Priority = 8, Local prot = 2

The number of paths to be pre-computed by each IP domain is therefore  $24960 * N(N-1)$  where N is the number of domain access points.

This means that with just 4 access points we have nearly 300000 paths to compute, advertise and maintain (if a change happens in the domain, due to a fault, or just the deployment of new traffic, a substantial number of paths need to be recomputed and the relevant changes advertised to the upper controller).

This seems quite challenging. In fact, if we assume a mean length of 1K for the json describing a path (a quite conservative estimate), reporting 300000 paths means transferring and then parsing more than 300 Mbytes for each domain. If we assume that 20% (to be checked) of this paths change when a new deployment of traffic occurs, we have 60 Mbytes of transfer for each domain traversed by a new end-to-end path. If a network has, let say, 20 domains (we want to estimate the load for a non-trivial domain setup) in the beginning a total

initial transfer of 6Gigs is needed, and eventually, assuming 4-5 domains are involved in mean during a path deployment we could have 240-300 Mbytes of changes advertised to the higher order controller.

Further bare-bone solutions can be investigated, removing some more options, if this is considered not acceptable; in conclusion, it seems that an approach based only on the information provided by the detailed connectivity matrix is hardly feasible, and could be applicable only to small networks with a limited meshing degree between domains and renouncing to a number of path computation features.

## Acknowledgments

The authors would like to thank Igor Bryskin and Xian Zhang for participating in the initial discussions that have triggered this work and providing valuable insights.

The authors would like to thank the authors of the TE Tunnel YANG model [TE-TUNNEL], in particular Igor Bryskin, Vishnu Pavan Beeram, Tarek Saad and Xufeng Liu, for their inputs to the discussions and support in having consistency between the Path Computation and TE Tunnel YANG models.

The authors would like to thank Adrian Farrel, Dhruv Dhody, Igor Bryskin, Julien Meuric and Lou Berger for their valuable input to the discussions that has clarified that the path being setup is not necessarily the same as the path that have been previously computed and, in particular to Dhruv Dhody, for his suggestion to describe the need for a path verification phase to check that the actual path being setup meets the required end-to-end metrics and constraints.

The authors would like to thank Tom Petch for his review and valuable comments to this document.

This document was prepared using 2-Word-v2.0.template.dot.

## Contributors

Dieter Beller  
Nokia  
Email: dieter.beller@nokia.com

Gianmarco Bruno  
Ericsson  
Email: gianmarco.bruno@ericsson.com

Francesco Lazzeri  
Ericsson  
Email: francesco.lazzeri@ericsson.com



Young Lee  
Huawei  
Email: leeyoung@huawei.com

Carlo Perocchio  
Ericsson  
Email: carlo.perocchio@ericsson.com

Olivier Dugeon  
Orange Labs  
Email: olivier.dugeon@orange.com

Julien Meuric  
Orange Labs  
Email: julien.meuric@orange.com

#### Authors' Addresses

Italo Busi (Editor)  
Huawei  
Email: italo.busi@huawei.com

Sergio Belotti (Editor)  
Nokia  
Email: sergio.belotti@nokia.com

Victor Lopez  
Telefonica  
Email: victor.lopezalvarez@telefonica.com

Oscar Gonzalez de Dios  
Telefonica  
Email: oscar.gonzalezdedios@telefonica.com

Anurag Sharma  
Google  
Email: ansha@google.com

Yan Shi  
China Unicom  
Email: shiyan49@chinaunicom.cn

Ricard Vilalta  
CTTC  
Email: ricard.vilalta@cttc.es

Karthik Sethuraman  
NEC  
Email: karthik.sethuraman@necam.com

Michael Scharf  
Nokia  
Email: michael.scharf@gmail.com

Daniele Ceccarelli  
Ericsson  
Email: daniele.ceccarelli@ericsson.com



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

X. Liu  
Volta Networks  
J. Tantsura  
Apstra Networks  
I. Bryskin  
Individual  
L. Contreras  
Telefonica  
Q. Wu  
Huawei  
S. Belotti  
R. Rokui  
Nokia  
July 12, 2020

Transport Network Slice YANG Data Model  
draft-liu-teas-transport-network-slice-yang-01

Abstract

This document describes a YANG data model for managing and controlling transport network slices, defined as transport slices in [I-D.nsdt-teas-transport-slice-definition].

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 . . . . .	2
1.1. Terminology . . . . .	3
1.2. Tree Diagrams . . . . .	3
2. Modeling Considerations . . . . .	3
2.1. Relationships to Related Topology Models . . . . .	3
2.2. Network Slice with TE . . . . .	4
2.3. ACTN for Network Slicing . . . . .	5
3. Model Applicability . . . . .	5
3.1. Network Slicing by Virtualization . . . . .	5
3.2. Network Slicing by TE Overlay . . . . .	8
4. Model Tree Structure . . . . .	10
5. YANG Module . . . . .	10
6. IANA Considerations . . . . .	16
7. Security Considerations . . . . .	17
8. Acknowledgements . . . . .	18
9. References . . . . .	18
9.1. Normative References . . . . .	18
9.2. Informative References . . . . .	20
Appendix A. Data Tree for the Example in Section 3.1. . . . .	22
A.1. Native Topology . . . . .	22
A.2. Network Slice Blue . . . . .	26
Authors' Addresses . . . . .	32

## 1. Introduction

This document defines a YANG [RFC7950] data model for for representing, managing, and controlling transport network slices, defined as transport slices in [I-D.nsdt-teas-transport-slice-definition]

The defined data model is an interface between clients and providers for configurations and state retrievals, so as to support transport network slicing as a service. Through this model, a client can learn the slicing capabilities and the available resources of the provider. A client can request or negotiate with a transport network slicing provider to create an instance. The client can incrementally update

its requirements on individual topology elements in the slice instance, and retrieve the operational states of these elements. With the help of other mechanisms and data models defined in IETF, the telemetry information can be published to the client.

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

### 1.1. Terminology

The key words "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 BCP 14, [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following terms are defined in [RFC7950] and are not redefined here:

- o augment
- o data model
- o data node

### 1.2. Tree Diagrams

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

## 2. Modeling Considerations

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

### 2.1. Relationships to Related Topology Models

There are several related YANG data models that have been defined in IETF. Some of these are:

Network Topology Model:  
Defined in [RFC8345].

OTN Topology Model:  
Defined in [I-D.ietf-ccamp-otn-topo-yang].



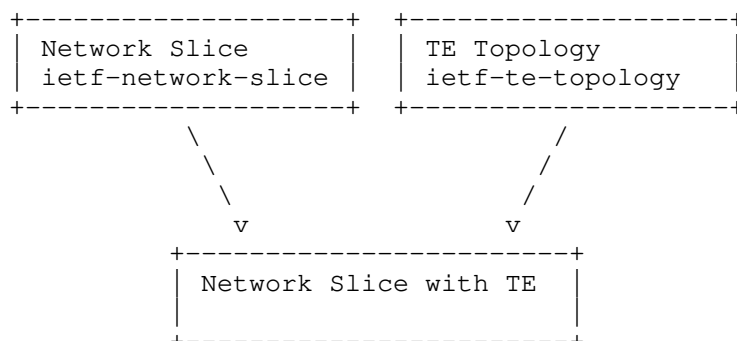


Figure 2: Network Slice with TE

This method can be applied to other types of network topology models too. For example, when a network topology instance is configured to have the types of "network-slice" defined in this document, "te-topology" defined in [I-D.ietf-teas-yang-te-topo], and "l3-unicast-topology" defined in [RFC8346], this network topology instance becomes a transport slice instance that can perform layer 3 traffic engineering.

### 2.3. ACTN for Network Slicing

Since ACTN topology data models are based on the network topology model defined in [RFC8345], the augmentations defined in this document are effective augmentations to the ACTN topology data models, resulting in making the ACTN framework [RFC8453] and data models [I-D.ietf-teas-actn-yang] capable of slicing networks with the required network characteristics.

## 3. Model Applicability

There are many technologies to achieve transport network slicing. The data model defined in this document can be applied to a wide ranges of cases. This section describes how this data model is applied to a few cases.

### 3.1. Network Slicing by Virtualization

In the case shown in Figure 3, node virtualization is used to separate and allocate resources in physical devices. Two virtual routers VR1 and VR2 are created over physical router R1. Each of the virtual routers takes a portion of the resources such as ports and memory in the physical router. Depending on the requirements and the implementations, they may share certain resources such as processors, ASICs, and switch fabric.



As an example, Appendix A. shows the JSON encoded data instances of the native topology and the customized topology for Network Slice Blue.

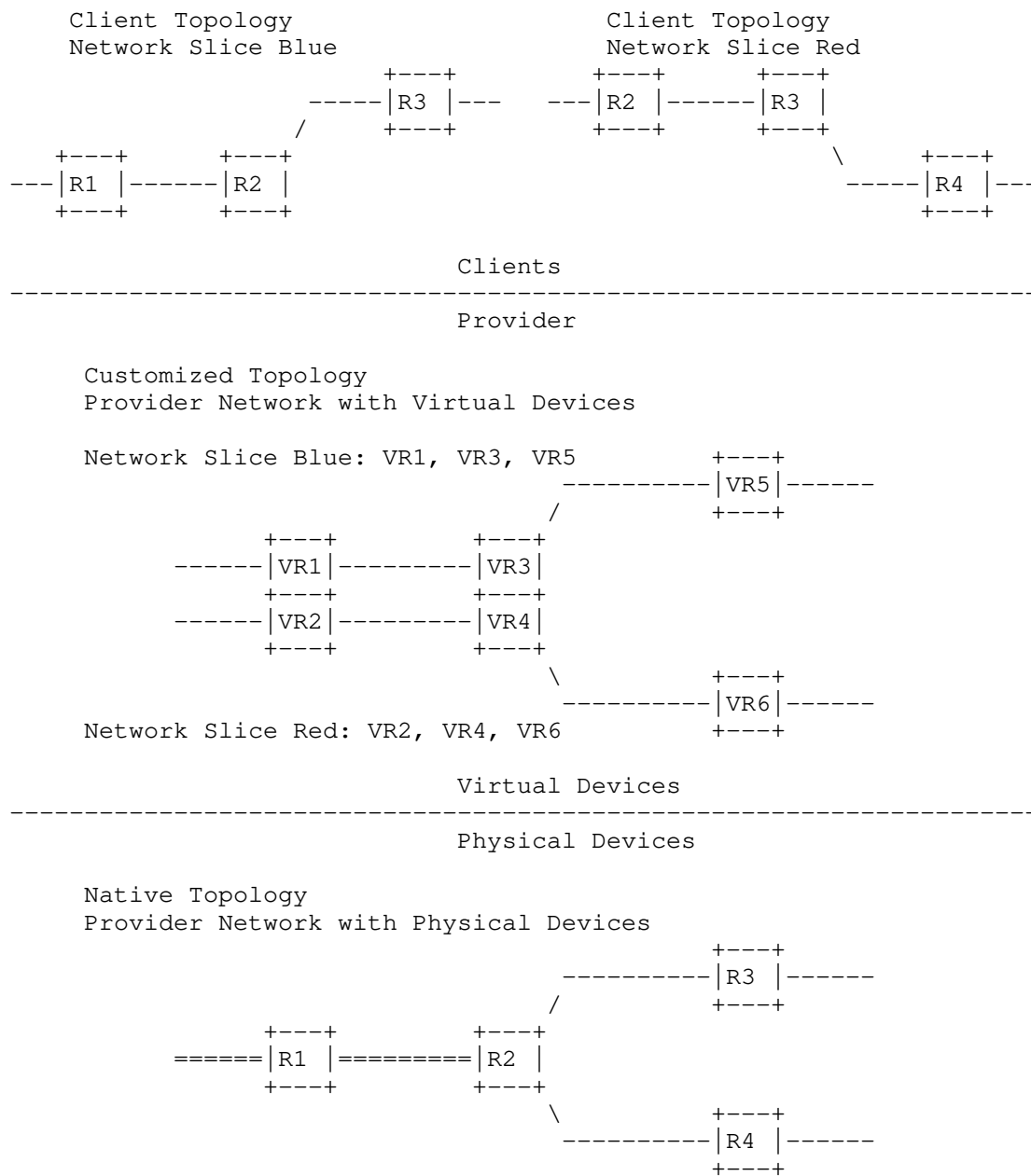


Figure 3: Network Slicing by Virtualization

### 3.2. Network Slicing by TE Overlay

Figure 5 shows a case where TE (Traffic Engineering) overlay is applied to achieve logically separated client transport network slices. In the underlay TE capable network, TE tunnels are established to support the TE links in the overlay network. These links and tunnels maintain the characteristics required by the clients. The provider selects the proper logical nodes and links in the overlay network, assigns them to specific transport network slices, and uses the data model defined in this document to send the results to the clients.

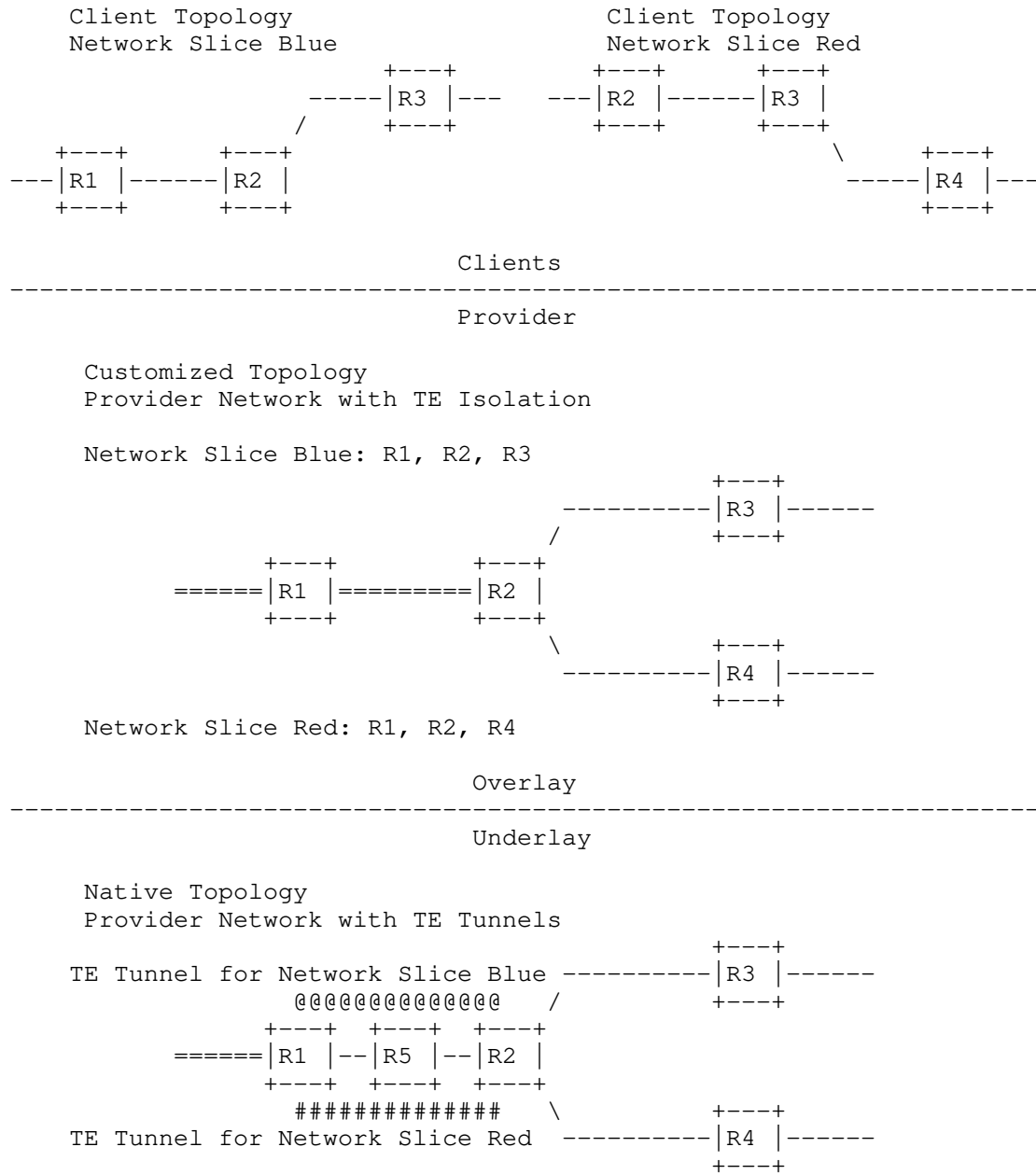


Figure 4: Network Slicing by TE Overlay

#### 4. Model Tree Structure

```
module: ietf-network-slice
  augment /nw:networks/nw:network/nw:network-types:
    +--rw network-slice!
  augment /nw:networks/nw:network:
    +--rw network-slice
      +--rw optimization-criterion?  identityref
      +--rw delay-tolerance?         boolean
      +--rw periodicity*             uint64
      +--rw isolation-level?         identityref
  augment /nw:networks/nw:network/nw:node:
    +--rw network-slice
      +--rw isolation-level?         identityref
      +--rw compute-node-id?        string
      +--rw storage-id?             string
  augment /nw:networks/nw:network/nt:link:
    +--rw network-slice
      +--rw delay-tolerance?        boolean
      +--rw periodicity*            uint64
      +--rw isolation-level?        identityref
```

#### 5. YANG Module

This module references [RFC8345], [RFC8776], and [GSMA-NS-Template]

```
<CODE BEGINS> file "ietf-network-slice@2020-07-12.yang"
module ietf-network-slice {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-network-slice";
  prefix "ns";

  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-types {
    prefix "te-types";
    reference

```

```
    "RFC 8776: Traffic Engineering Common YANG Types";
  }

organization
  "IETF Traffic Engineering Architecture and Signaling (TEAS)
  Working Group";

contact
  "WG Web: <http://tools.ietf.org/wg/teas/>
  WG List: <mailto:teas@ietf.org>

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

  Editor: Jeff Tantsura
  <mailto:jefftant.ietf@gmail.com>

  Editor: Igor Bryskin
  <mailto:i_bryskin@yahoo.com>

  Editor: Luis Miguel Contreras Murillo
  <mailto:luismiguel.contrerasmurillo@telefonica.com>

  Editor: Qin Wu
  <mailto:bill.wu@huawei.com>

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

  Editor: Reza Rokui
  <mailto:reza.rokui@nokia.com>
  ";

description
  "YANG data model for representing and managing network
  slices.

  Copyright (c) 2019 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 revision";
  reference
    "RFC XXXX: YANG Data Model for Network Slices";
}

/*
 * Identities
 */
identity isolation-level {
  description
    "Base identity for the isolation-level.";
  reference
    "GSMA-NS-Template: Generic Network Slice Template,
    Version 1.0.";
}
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 network-slice-topology-attributes {
    description "Network Slice topology scope attributes.";
    container network-slice {
      description
        "Containing Network Slice attributes.";
      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 1.0.";
      }
    }
    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 1.0.";
    }
    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:";
    }
} // network-slice
} // network-slice-topology-attributes

grouping network-slice-node-attributes {
    description "Network Slice node scope attributes.";
    container network-slice {
        description
            "Containing Network Slice attributes.";
        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:";
        }
        leaf compute-node-id {
            type string;
            description
                "Reference to a compute node instance specified in
                a data model specifying the computing resources.";
        }
        leaf storage-id {
            type string;
            description
                "Reference to a storage instance specified in
                a data model specifying the storage resources.";
        }
    }
} // network-slice
} // network-slice-node-attributes
```

```
grouping network-slice-link-attributes {
  description "Network Slice link scope attributes";
  container network-slice {
    description
      "Containing Network Slice attributes.";
    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 1.0.";
    }
    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 1.0.";
    }
    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:";
    }
  } // network-slice
} // network-slice-link-attributes

/*
 * Data nodes
 */
augment "/nw:networks/nw:network/nw:network-types" {
  description
    "Defines the Network Slice topology type.";
  container network-slice {
    presence "Indicates Network Slice topology";
    description
      "Its presence identifies the Network Slice type.";
  }
}
```

```
augment "/nw:networks/nw:network" {
  when "nw:network-types/ns:network-slice" {
    description "Augment only for Network Slice topology.";
  }
  description "Augment topology configuration and state.";
  uses network-slice-topology-attributes;
}

augment "/nw:networks/nw:network/nw:node" {
  when "../nw:network-types/ns:network-slice" {
    description "Augment only for Network Slice topology.";
  }
  description "Augment node configuration and state.";
  uses network-slice-node-attributes;
}

augment "/nw:networks/nw:network/nt:link" {
  when "../nw:network-types/ns:network-slice" {
    description "Augment only for Network Slice topology.";
  }
  description "Augment link configuration and state.";
  uses network-slice-link-attributes;
}
}
<CODE ENDS>
```

## 6. IANA Considerations

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number (and remove this note).

This document registers the following namespace URIs in the IETF XML registry [RFC3688]:

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

This document registers the following YANG modules in the YANG Module Names registry [RFC6020]:

```
-----  
name:          ietf-l3-te-topology  
namespace:     urn:ietf:params:xml:ns:yang:ietf-network-slice  
prefix:        ns  
reference:     RFC XXXX  
-----
```

## 7. Security Considerations

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

The Network Configuration Access Control Model (NACM) [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. These are the subtrees and data nodes and their sensitivity/vulnerability:

```
/nw:networks/nw:network/nw:network-types/ns:network-slice  
  This subtree specifies the network slice type. Modifying the  
  configurations can make network slice type invalid and cause  
  interruption to transport network slices.
```

```
/nw:networks/nw:network/ns:network-slice  
  This subtree specifies the topology-wide configurations.  
  Modifying the configurations here can cause traffic  
  characteristics changed in this transport network slice and  
  related networks.
```

```
/nw:networks/nw:network/nw:node/ns:network-slice  
  This subtree specifies the configurations of the nodes in a  
  transport network slice. Modifying the configurations in this  
  subtree can change the traffic characteristics on this node and  
  the related networks.
```

```
/nw:networks/nw:network/nt:link/ns:network-slice
```

This subtree specifies the configurations of the links in a transport network slice. Modifying the configurations in this subtree can change the traffic characteristics on this link and the related networks.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

/nw:networks/nw:network/nw:network-types/ns:network-slice  
Unauthorized access to this subtree can disclose the network slice type.

/nw:networks/nw:network/ns:network-slice  
Unauthorized access to this subtree can disclose the topology-wide states.

/nw:networks/nw:network/nw:node/ns:network-slice  
Unauthorized access to this subtree can disclose the operational state information of the nodes in a transport network slice.

/nw:networks/nw:network/nt:link/ns:network-slic  
Unauthorized access to this subtree can disclose the operational state information of the links in a transport network slice.

## 8. Acknowledgements

The TEAS Network Slicing Design Team (NSDT) members included Aijun Wang, Dong Jie, Eric Gray, Jari Arkko, Jeff Tantsura, John E Drake, Luis M. Contreras, Rakesh Gandhi, Ran Chen, Reza Rokui, Ricard Vilalta, Ron Bonica, Sergio Belotti, Tomonobu Niwa, Xuesong Geng, and Xufeng Liu.

## 9. References

### 9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.

- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [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>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <<https://www.rfc-editor.org/info/rfc8345>>.
- [RFC8346] Clemm, A., Medved, J., Varga, R., Liu, X., Ananthakrishnan, H., and N. Bahadur, "A YANG Data Model for Layer 3 Topologies", RFC 8346, DOI 10.17487/RFC8346, March 2018, <<https://www.rfc-editor.org/info/rfc8346>>.

- [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>>.
- [RFC8776] Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Common YANG Data Types for Traffic Engineering", RFC 8776, DOI 10.17487/RFC8776, June 2020, <<https://www.rfc-editor.org/info/rfc8776>>.
- [I-D.ietf-teas-yang-te-topo]  
Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", draft-ietf-teas-yang-te-topo-22 (work in progress), June 2019.
- [GSMA-NS-Template]  
GSM Association, "Generic Network Slice Template, Version 1.0", NG.116, May 2019.
- [I-D.nsd-t-teas-transport-slice-definition]  
Rokui, R., Homma, S., Makhijani, K., and L. Contreras, "IETF Definition of Transport Slice", draft-nsd-t-teas-transport-slice-definition-02 (work in progress), April 2020.

## 9.2. Informative References

- [RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <<https://www.rfc-editor.org/info/rfc7951>>.
- [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>>.
- [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>>.
- [I-D.ietf-ccamp-otn-topo-yang]  
Zheng, H., Busi, I., Liu, X., Belotti, S., and O. Dios, "A YANG Data Model for Optical Transport Network Topology", draft-ietf-ccamp-otn-topo-yang-10 (work in progress), March 2020.

[I-D.ietf-i2rs-yang-l2-network-topology]

Dong, J., Wei, X., WU, Q., Boucadair, M., and A. Liu, "A YANG Data Model for Layer 2 Network Topologies", draft-ietf-i2rs-yang-l2-network-topology-14 (work in progress), June 2020.

[I-D.ietf-teas-actn-yang]

Lee, Y., Zheng, H., Ceccarelli, D., Yoon, B., Dios, O., Shin, J., and S. Belotti, "Applicability of YANG models for Abstraction and Control of Traffic Engineered Networks", draft-ietf-teas-actn-yang-05 (work in progress), February 2020.

[I-D.nsd-t-teas-ns-framework]

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



## Appendix A. Data Tree for the Example in Section 3.1.

## A.1. Native Topology

This section contains an example of an instance data tree in the JSON encoding [RFC7951]. The example instantiates "ietf-network" for the native topology depicted in Figure 3.

```
{
  "ietf-network:networks": {
    "network": [
      {
        "network-id": "example-native-topology",
        "network-types": {
        },
        "node": [
          {
            "node-id": "R1",
            "ietf-network-topology:termination-point": [
              {
                "tp-id": "1-0-1"
              },
              {
                "tp-id": "1-0-2"
              },
              {
                "tp-id": "1-2-1"
              },
              {
                "tp-id": "1-2-2"
              }
            ]
          },
          {
            "node-id": "R2",
            "ietf-network-topology:termination-point": [
              {
                "tp-id": "2-1-1"
              },
              {
                "tp-id": "2-1-2"
              },
              {
                "tp-id": "2-3-1"
              },
              {
                "tp-id": "2-4-1"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```
    ]
  },
  {
    "node-id": "R3",
    "ietf-network-topology:termination-point": [
      {
        "tp-id": "3-0-1"
      },
      {
        "tp-id": "3-2-1"
      }
    ]
  },
  {
    "node-id": "R4",
    "ietf-network-topology:termination-point": [
      {
        "tp-id": "4-0-1"
      },
      {
        "tp-id": "4-2-1"
      }
    ]
  }
],
"ietf-network-topology:link": [
  {
    "link-id": "R1,1-0-1,,",
    "source": {
      "source-node": "R1",
      "source-tp": "1-0-1"
    }
  },
  {
    "link-id": ",,R1,1-0-1",
    "destination": {
      "dest-node": "R1",
      "dest-tp": "1-0-1"
    }
  },
  {
    "link-id": "R1,1-0-2,,",
    "source": {
      "source-node": "R1",
      "source-tp": "1-0-2"
    }
  },
  {
    {
```

```
    "link-id":",,R1,1-0-2",
    "destination": {
      "dest-node":"R1",
      "dest-tp":"1-0-2"
    }
  },
  {
    "link-id":"R1,1-2-1,R2,2-1-1",
    "source": {
      "source-node":"R1",
      "source-tp":"1-2-1"
    },
    "destination": {
      "dest-node":"R2",
      "dest-tp":"2-1-1"
    }
  },
  {
    "link-id":"R2,2-1-1,R1,1-2-1",
    "source": {
      "source-node":"R2",
      "source-tp":"2-1-1"
    },
    "destination": {
      "dest-node":"R1",
      "dest-tp":"1-2-1"
    }
  },
  {
    "link-id":"R1,1-2-2,R2,2-1-2",
    "source": {
      "source-node":"R1",
      "source-tp":"1-2-2"
    },
    "destination": {
      "dest-node":"R2",
      "dest-tp":"2-1-2"
    }
  },
  {
    "link-id":"R2,2-1-2,R1,1-2-2",
    "source": {
      "source-node":"R2",
      "source-tp":"2-1-2"
    },
    "destination": {
      "dest-node":"R1",
      "dest-tp":"1-2-2"
    }
  }
```

```
    }
  },
  {
    "link-id": "R2,2-3-1,R3,3-2-1",
    "source": {
      "source-node": "R2",
      "source-tp": "2-3-1"
    },
    "destination": {
      "dest-node": "R3",
      "dest-tp": "3-2-1"
    }
  },
  {
    "link-id": "R3,3-2-1,R2,2-3-1",
    "source": {
      "source-node": "R3",
      "source-tp": "3-2-1"
    },
    "destination": {
      "dest-node": "R2",
      "dest-tp": "2-3-1"
    }
  },
  {
    "link-id": "R2,2-4-1,R4,4-2-1",
    "source": {
      "source-node": "R2",
      "source-tp": "2-4-1"
    },
    "destination": {
      "dest-node": "R4",
      "dest-tp": "4-2-1"
    }
  },
  {
    "link-id": "R4,4-2-1,R2,2-4-1",
    "source": {
      "source-node": "R4",
      "source-tp": "4-2-1"
    },
    "destination": {
      "dest-node": "R2",
      "dest-tp": "2-4-1"
    }
  },
  {
    "link-id": "R3,3-0-1,,",
```

```

        "source": {
          "source-node": "R3",
          "source-tp": "3-0-1"
        }
      },
      {
        "link-id": ",,R3,3-0-1",
        "destination": {
          "dest-node": "R3",
          "dest-tp": "3-0-1"
        }
      },
      {
        "link-id": "R4,4-0-1,,",
        "source": {
          "source-node": "R4",
          "source-tp": "4-0-1"
        }
      },
      {
        "link-id": ",,R4,4-0-1",
        "destination": {
          "dest-node": "R4",
          "dest-tp": "4-0-1"
        }
      }
    ]
  }
}

```

## A.2. Network Slice Blue

This section contains an example of an instance data tree in the JSON encoding [RFC7951]. The example instantiates "ietf-network-slice" for the topology customized for Network Slice Blue depicted in Figure 3.

```

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

```

```
    }
  },
  "supporting-network": [
    {
      "network-ref": "example-native-topology"
    }
  ],
  "node": [
    {
      "node-id": "VR1",
      "supporting-node": [
        {
          "network-ref": "example-native-topology",
          "node-ref": "R1"
        }
      ],
      "ietf-network-slice:network-slice": {
        "isolation-level":
          "ietf-network-slice:physical-memory-isolation"
      },
      "ietf-network-topology:termination-point": [
        {
          "tp-id": "1-0-1"
        },
        {
          "tp-id": "1-3-1"
        }
      ]
    },
    {
      "node-id": "VR3",
      "supporting-node": [
        {
          "network-ref": "example-native-topology",
          "node-ref": "R2"
        }
      ],
      "ietf-network-slice:network-slice": {
        "isolation-level":
          "ietf-network-slice:physical-memory-isolation"
      },
      "ietf-network-topology:termination-point": [
        {
          "tp-id": "3-1-1"
        },
        {
          "tp-id": "3-5-1"
        }
      ]
    }
  ]
}
```

```
    ]
  },
  {
    "node-id": "VR5",
    "supporting-node": [
      {
        "network-ref": "example-native-topology",
        "node-ref": "R3"
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-memory-isolation"
    },
    "ietf-network-topology:termination-point": [
      {
        "tp-id": "5-3-1"
      },
      {
        "tp-id": "5-0-1"
      }
    ]
  }
],
"ietf-network-topology:link": [
  {
    "link-id": "VR1,1-0-1,,",
    "source": {
      "source-node": "VR1",
      "source-tp": "1-0-1"
    },
    "supporting-link": [
      {
        "network-ref": "example-native-topology",
        "link-ref": "R1,1-0-1,,"
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-network-isolation"
    }
  },
  {
    "link-id": ",,VR1,1-0-1",
    "destination": {
      "dest-node": "VR1",
      "dest-tp": "1-0-1"
    }
  },

```

```
"supporting-link": [
  {
    "network-ref": "example-native-topology",
    "link-ref": ",,R1,1-0-1"
  }
],
"ietf-network-slice:network-slice": {
  "isolation-level":
    "ietf-network-slice:physical-network-isolation"
}
},
{
  "link-id": "VR1,1-3-1,VR3,3-1-1",
  "source": {
    "source-node": "VR1",
    "source-tp": "1-3-1"
  },
  "destination": {
    "dest-node": "VR3",
    "dest-tp": "3-1-1"
  },
  "supporting-link": [
    {
      "network-ref": "example-native-topology",
      "link-ref": "R1,1-2-1,R2,2-1-1"
    }
  ],
  "ietf-network-slice:network-slice": {
    "isolation-level":
      "ietf-network-slice:physical-network-isolation"
  }
},
{
  "link-id": "VR3,3-1-1,VR1,1-3-1",
  "source": {
    "source-node": "VR3",
    "source-tp": "3-1-1"
  },
  "destination": {
    "dest-node": "R1",
    "dest-tp": "1-3-1"
  },
  "supporting-link": [
    {
      "network-ref": "example-native-topology",
      "link-ref": "R2,2-1-1,R1,1-2-1"
    }
  ],
}
```



```
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-network-isolation"
    }
  },
  {
    "link-id": "VR3,3-5-1,VR5,5-3-1",
    "source": {
      "source-node": "VR3",
      "source-tp": "3-5-1"
    },
    "destination": {
      "dest-node": "VR5",
      "dest-tp": "5-3-1"
    },
    "supporting-link": [
      {
        "network-ref": "example-native-topology",
        "link-ref": "R2,2-3-1,R3,3-2-1"
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-network-isolation"
    }
  },
  {
    "link-id": "VR5,5-3-1,VR3,3-5-1",
    "source": {
      "source-node": "VR5",
      "source-tp": "5-3-1"
    },
    "destination": {
      "dest-node": "VR3",
      "dest-tp": "3-5-1"
    },
    "supporting-link": [
      {
        "network-ref": "example-native-topology",
        "link-ref": "R3,3-2-1,R2,2-3-1"
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-network-isolation"
    }
  },
  {
```

```

    "link-id": "VR5,5-0-1,,",
    "source": {
      "source-node": "VR5",
      "source-tp": "5-0-1"
    },
    "supporting-link": [
      {
        "network-ref": "example-native-topology",
        "link-ref": "R3,3-0-1,,",
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-network-isolation"
    }
  },
  {
    "link-id": ",,VR5,5-0-1",
    "destination": {
      "dest-node": "VR5",
      "dest-tp": "5-0-1"
    },
    "supporting-link": [
      {
        "network-ref": "example-native-topology",
        "link-ref": ",,R3,3-0-1"
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-network-isolation"
    }
  }
],
"ietf-network-slice:network-slice": {
  "optimization-criterion":
    "ietf-te-types:of-minimize-cost-path",
  "isolation-level":
    "ietf-network-slice:physical-isolation"
}
]
}
}

```

Authors' Addresses

Xufeng Liu  
Volta Networks

EMail: xufeng.liu.ietf@gmail.com

Jeff Tantsura  
Apstra Networks

EMail: jefftant.ietf@gmail.com

Igor Bryskin  
Individual

EMail: i\_bryskin@yahoo.com

Luis Miguel Contreras Murillo  
Telefonica

EMail: luismiguel.contrerasmurillo@telefonica.com

Qin Wu  
Huawei

EMail: bill.wu@huawei.com

Sergio Belotti  
Nokia

EMail: sergio.belotti@nokia.com

Reza Rokui  
Nokia  
Canada

EMail: reza.rokui@nokia.com

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: 14 January 2021

E. Gray, Ed.  
Ericsson  
J. Drake, Ed.  
Juniper Networks  
13 July 2020

Framework for Transport Network Slices  
draft-nsdt-teas-ns-framework-04

Abstract

This memo discusses setting up special-purpose transport connections using existing IETF technologies. These connections are called transport slices for the purposes of this memo. The memo discusses the general framework for this setup, the necessary system components and interfaces, and how abstract requests can be mapped to more specific technologies. The memo also discusses related considerations with monitoring and security.

This memo is intended for discussing interfaces and technologies. It is not intended to be a new set of concrete interfaces or technologies. Rather, it should be seen as an explanation of how some existing, concrete IETF VPN and traffic-engineering technologies can be used to create transport slices. Note that there are a number of these technologies, and new technologies or capabilities keep being added. This memo is also not intended presume any particular technology choice.

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 14 January 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 . . . . .	2
2. Transport Slice Objectives . . . . .	4
3. Framework . . . . .	5
3.1. Management systems or other applications . . . . .	6
3.2. Expressing connectivity intents . . . . .	6
3.3. Transport Slice Controller (TSC) . . . . .	8
3.3.1. Northbound Interface (NBI) . . . . .	9
3.4. Mapping . . . . .	9
3.5. Underlying technology . . . . .	9
4. Applicability of ACTN to Transport Slices . . . . .	10
5. Considerations . . . . .	12
5.1. Monitoring . . . . .	12
5.2. Security Considerations . . . . .	13
5.3. Privacy Considerations . . . . .	13
5.4. IANA Considerations . . . . .	13
6. Acknowledgments . . . . .	13
7. References . . . . .	14
7.1. Normative References . . . . .	14
7.2. Informative References . . . . .	14
Contributors . . . . .	17
Authors' Addresses . . . . .	18

## 1. Introduction

This draft provides a framework for discussing transport slices, as defined in [I-D.nsdt-teas-transport-slice-definition] It is the intention in this document to use terminology consistent with this and other definitions provided in that draft.

In particular, this document uses the following terminology defined in the definitions document:

- \* Transport Slice
- \* Transport Slice Controller (TSC)
- \* Transport Network Controller (TNC)
- \* Northbound Interface (NBI)
- \* Southbound Interface (SBI)

This framework is intended as a structure for discussing interfaces and technologies. It is not intended to be a new set of concrete interfaces or technologies. Rather, the idea is that existing or under-development IETF technologies (plural) can be used to realize the ideas expressed here.

For example, virtual private networks (VPNs) have served the industry well as a means of providing different groups of users with logically isolated access to a common network. The common or base network that is used to provide the VPNs is often referred to as an underlay network, and the VPN is often called an overlay network. As an example technology, a VPN may in turn serve as an underlay network for transport slices.

Note: It is conceivable that extensions to these IETF technologies are needed in order to fully support all the ideas that can be implemented with slices, but at least in the beginning there is no plan for the creation of new protocols or interfaces.

Driven largely by needs surfacing from 5G, the concept of network slicing has gained traction ([NGMN-NS-Concept], [TS23501], [TS28530], and [BBF-SD406]). In [TS23501], Network Slice is defined as "a logical network that provides specific network capabilities and network characteristics", and a Network Slice Instance is defined as "A set of Network Function instances and the required resources (e.g. compute, storage and networking resources) which form a deployed Network Slice". According to [TS28530], an end-to-end network slice consists of three major types of network segments: Radio Access Network (RAN), Transport Network (TN) and Core Network (CN). Transport network provides the required connectivity between different entities in RAN and CN segments of an end-to-end network slice, with a specific performance commitment. For each end-to-end network slice, the topology and performance requirement on transport network can be very different, which requires the transport network to have the capability of supporting multiple different transport slices.

While network slices are commonly discussed in the context of 5G, it is important to note that transport slices are a narrower concept, and focus primarily on particular network connectivity aspects. Other systems, including 5G deployments, may use transport slices as a component to create entire systems and concatenated constructs that match their needs, including end-to-end connectivity.

A transport slice could span multiple technologies and multiple administrative domains. Depending on the transport slice consumer's requirements, a transport slice could be isolated from other, often concurrent transport slices in terms of data, control and management planes.

The consumer expresses requirements for a particular transport slice by specifying what is required rather than how the requirement is to be fulfilled. That is, the transport slice consumer's view of a transport slice is an abstract one.

Thus, there is a need to create logical network structures with required characteristics. The consumer of such a logical network can require a degree of isolation and performance that previously might not have been satisfied by traditional overlay VPNs. Additionally, the transport slice consumer might ask for some level of control of their virtual networks, e.g., to customize the service paths in a network slice.

This document specifies a framework for the use of existing technologies as components to provide a transport slice service, and might also discuss (or reference) modified and potential new technologies, as they develop (such as candidate technologies described in section 5 of [I-D.ietf-teas-enhanced-vpn]).

## 2. Transport Slice Objectives

It is intended that transport slices can be created to meet specific requirements, typically expressed as bandwidth, latency, latency variation, and other desired or required characteristics. Creation is initiated by a management system or other application used to specify network-related conditions for particular traffic flows.

And it is intended that, once created, these slices can be monitored, modified, deleted, and otherwise managed.

It is also intended that applications and components will be able to use these transport slices to move packets between the specified end-points in accordance with specified characteristics.

As an example of requirements that might apply to transport slices, see [I-D.ietf-teas-enhanced-vpn] (in particular, section 3).

### 3. Framework

A number of transport slice services will typically be provided over a shared underlying network infrastructure. Each transport slice consists of both the overlay connectivity and a specific set of dedicated network resources and/or functions allocated in a shared underlay network to satisfy the needs of the transport slice consumer. In at least some examples of underlying network technologies, the integration between the overlay and various underlay resources is needed to ensure the guaranteed performance requested for different transport slices.

#### Transport Slice Definition

([I-D.nsdt-teas-transport-slice-definition]) defines the role of a Customer (or User) and a Transport Slice Controller. That draft also defines a TSC Northbound Interface (NBI).

A transport slice user is served by the Transport Slice Controller (TSC), as follows:

- \* The TSC takes requests from a management system or other application, which are then communicated via an NBI. This interface carries data objects the transport slice user provides, describing the needed transport slices in terms of topology, applicable service level objectives (SLO), and any monitoring and reporting requirements that may apply. Note that - in this context - "topology" means what the transport slice connectivity is meant to look like from the user's perspective; it may be as simple as a list of mutually (and symmetrically) connected end points, or it may be complicated by details of connection asymmetry, per-connection SLO requirements, etc.
- \* These requests are assumed to be translated by one or more underlying systems, which are used to establish specific transport slice instances on top of an underlying network infrastructure.
- \* The TSC maintains a record of the mapping from user requests to slice instantiations, as needed to allow for subsequent control functions (such as modification or deletion of the requested slices), and as needed for any requested monitoring and reporting functions.

Section 3 of [I-D.ietf-teas-enhanced-vpn] provides an example architecture that might apply in using the technology described in that document.



### 3.1. Management systems or other applications

The transport slice system is used by a management system or other application. These systems and applications may also be a part of a higher level function in the system, e.g., putting together network functions, access equipment, application specific components, as well as the transport slices.

### 3.2. Expressing connectivity intents

The Transport Slice Controller (TSC) northbound interface (NBI) can be used to communicate between transport slice users (or consumers) and the TSC.

A transport slice user may be a network operator who, in turn, provides the transport slice to another transport slice user or consumer.

Using the NBI, a consumer expresses requirements for a particular slice by specifying what is required rather than how that is to be achieved. That is, the consumer's view of a slice is an abstract one. Consumers normally have limited (or no) visibility into the provider network's actual topology and resource availability information.

This should be true even if both the consumer and provider are associated with a single administrative domain, in order to reduce the potential for adverse interactions between transport slice consumers and other users of the transport network infrastructure.

The benefits of this model can include:

- \* **Security:** because the transport network (or network operator) does not need to expose network details (topology, capacity, etc.) to transport slice consumers the transport network components are less exposed to attack;
- \* **Layered Implementation:** the transport network comprises network elements that belong to a different layer network than consumer applications, and network information (advertisements, protocols, etc.) that a consumer cannot interpret or respond to (note - a consumer should not use network information not exposed via the TSC NBI, even if that information is available);
- \* **Scalability:** consumers do not need to know any information beyond that which is exposed via the NBI.

The general issues of abstraction in a TE network is described more fully in [RFC7926].

This framework document does not assume any particular layer at which transport slices operate as a number of layers (including virtual L2, Ethernet or IP connectivity) could be employed.

Data models and interfaces are of course needed to set up transport slices, and specific interfaces may have capabilities that allow creation of specific layers.

Layered virtual connections are comprehensively discussed in IETF documents and are widely supported. See, for instance, GMPLS-based networks ([RFC5212] and [RFC4397]), or ACTN ([RFC8453] and [RFC8454]). The principles and mechanisms associated with layered networking are applicable to transport slices.

There are several IETF-defined mechanisms for expressing the need for a desired logical network. The NBI carries data either in a protocol-defined format, or in a formalism associated with a modeling language.

For instance:

- \* Path Computation Element (PCE) Communication Protocol (PCEP) [RFC5440] and GMPLS User-Network Interface (UNI) using RSVP-TE [RFC4208] use a TLV-based binary encoding to transmit data.
- \* Network Configuration Protocol (NETCONF) [RFC6241] and RESTCONF Protocol [RFC8040] use XML and JSON encoding.
- \* gRPC/GNMI [I-D.openconfig-rtgwg-gnmi-spec] uses a binary encoded programmable interface;
- \* SNMP ([RFC3417], [RFC3412] and [RFC3414] uses binary encoding (ASN.1).
- \* For data modeling, YANG ([RFC6020] and [RFC7950]) may be used to model configuration and other data for NETCONF, RESTCONF, and GNMI - among others; ProtoBufs can be used to model gRPC and GNMI data; Structure of Management Information (SMI) [RFC2578] may be used to define Management Information Base (MIB) modules for SNMP, using an adapted subset of OSI's Abstract Syntax Notation One (ASN.1, 1988).

While several generic formats and data models for specific purposes exist, it is expected that transport slice management may require enhancement or augmentation of existing data models.

### 3.3. Transport Slice Controller (TSC)

The transport slice controller takes abstract requests for transport slices and implements them using a suitable underlying technology. A transport slice controller is the key building block for control and management of the transport slice. It provides the creation/modification/deletion, monitoring and optimization of transport Slices in a multi-domain, a multi-technology and multi-vendor environment.

A TSC northbound interface (NBI) is needed for communicating details of a transport slice (configuration, selected policies, operational state, etc.), as well as providing information to a slice requester/consumer about transport slice status and performance. The details for this NBI are not in scope for this document.

The controller provides the following functions:

- \* Provides a technology-agnostic NBI for creation/modification/deletion of the transport slices. The API exposed by this NBI communicates the endpoints of the transport slice, transport slice SLO parameters (and possibly monitoring thresholds), applicable input selection (filtering) and various policies, and provides a way to monitor the slice.
- \* Determines an abstract topology connecting the endpoints of the transport slice that meets criteria specified via the NBI. The TSC also retains information about the mapping of this abstract topology to underlying components of the transport slice as necessary to monitor transport slice status and performance.
- \* Provides "Mapping Functions" for the realization of transport slices. In other words, it will use the mapping functions that:  
  
map technology-agnostic NBI request to technology-specific SBIs.  
  
map filtering/selection information as necessary to entities in the underlay network.
- \* Via an SBI, the controller collects telemetry data (e.g. OAM results, statistics, states etc.) for all elements in the abstract topology used to realize the transport slice.
- \* Using the telemetry data from the underlying realization of a transport slice (i.e. services/paths/tunnels), evaluates the current performance against transport slice SLO parameters and exposes them to the transport slice consumer via the NBI. The TSC NBI may also include a capability to provide notification in case

the transport slice performance reaches threshold values defined by the transport slice consumer.

### 3.3.1. Northbound Interface (NBI)

The Transport Slice Controller provides a Northbound Interface (NBI) that allows consumers of network slices to request and monitor transport slices. Consumers operate on abstract transport slices, with details related to their realization hidden.

The NBI complements various IETF services, tunnels, path models by providing an abstract layer on top of these models.

The NBI is independent of type of network functions or services that need to be connected, i.e. it is independent of any specific storage, software, protocol, or platform used to realize physical or virtual network connectivity or functions in support of transport slices.

The NBI uses protocol mechanisms and information passed over those mechanisms to convey desired attributes for transport slices and their status. The information is expected to be represented as a well-defined data model, and should include at least endpoint and connectivity information, SLO specification, and status information.

To accomplish this, the NBI needs to convey information needed to support communication across the NBI, in terms of identifying the transport slices, as well providing the above model information.

### 3.4. Mapping

The main task of the transport slice controller is to map abstract transport slice requirements to concrete technologies and establish the required connectivity, and ensuring that required resources are allocated to the transport slice.

### 3.5. Underlying technology

There are a number of different technologies that can be used, including physical connections, MPLS, TSN, Flex-E, etc.

See [I-D.ietf-teas-enhanced-vpn] - section 5 - for instance, for example underlying technologies.

Also, as outlined in "applicability of ACTN to Transport Slices" below, ACTN ([RFC8453]) offers a framework that is used elsewhere in IETF specifications to create virtual network (VN) services similar to Transport Slices.

A transport slice can be realized in a network, using specific underlying technology or technologies. The creation of a new transport slice will be initiated with following three steps:

- \* Step 1: A higher level system requests connections with specific characteristics via NBI.
- \* Step 2: This request will be processed by a Transport Slice Controller which specifies a mapping between northbound request to any IETF Services, Tunnels, and paths models.
- \* Step 3: A series of requests for creation of services, tunnels and paths will be sent to the network to realize the transport slice.

It is very clear that regardless of how transport slice is realized in the network (i.e. using tunnels of type RSVP or SR), the definition of transport slice does not change at all but rather its realization.

#### 4. Applicability of ACTN to Transport Slices

Abstraction and Control of TE Networks (ACTN - [RFC8453]) is an example of similar IETF work. ACTN defines three controllers to support virtual network (VN) services -

- \* Customer Network Controller (CNC),
- \* Multi-Domain Service Coordinator (MDSC) and
- \* Provisioning Network Controller (PNC).

A CNC is responsible for communicating a customer's VN requirements.

A MDSC is responsible for multi-domain coordination, virtualization (or abstraction), customer mapping/translation and virtual service coordination to realize the VN requirement. Its key role is to detach the network/service requirements from the underlying technology.

A PNC oversees the configuration, monitoring and collection of the network topology. The PNC is a underlay technology specific controller.

While the ACTN framework is a generic VN framework that is used for various VN service beyond the transport slice, it is still a suitable basis to understand how the various controllers interact to realize a transport slice.

One possible mapping between the transport slice, and ACTN, definitions is as shown in Figure 1 below.

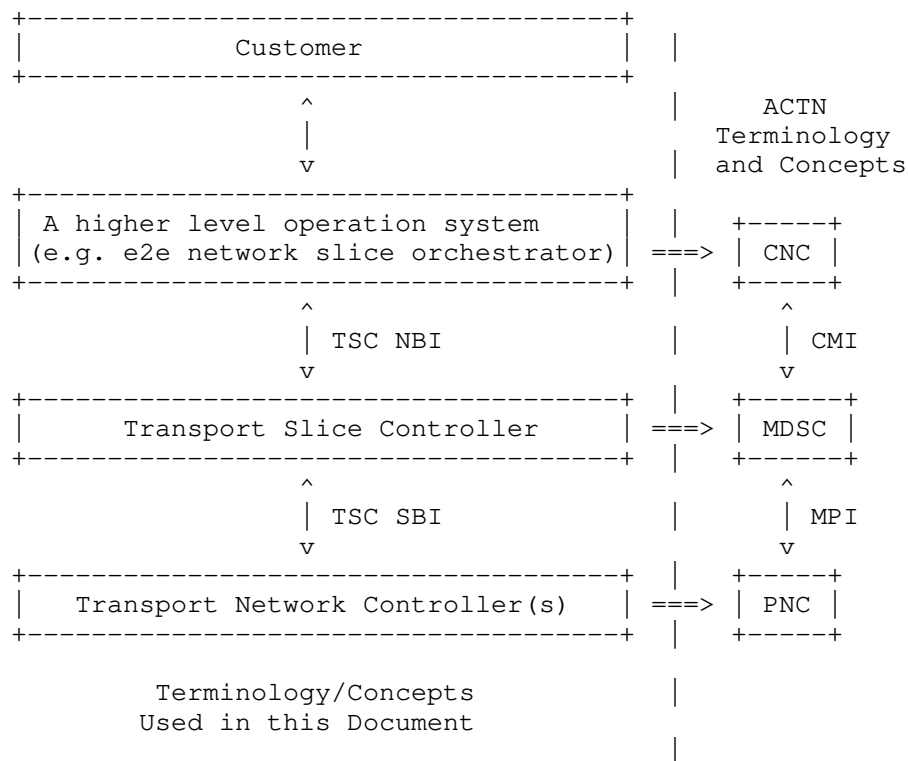


Figure 1

Note that the left-hand side of this figure comes from Transport Slice Definition ([I-D.nsdtd-teas-transport-slice-definition]).

The TSC NBI conveys the generic transport slice requirements. These may then be realized using an SBI within the TSC.

As per [RFC8453] and [I-D.ietf-teas-actn-yang], the CNC-MDSC Interface (CMI) is used to convey the virtual network service requirements along with the service models and the MDSC-PNC Interface (MPI) is used to realize the service along network configuration models. [I-D.ietf-teas-te-service-mapping-yang] further describe how the VPN services can be mapped to the underlying TE resources.

The Transport Network Controller (TNC) is depicted as a single block, analogous to the Provisioning Network Controller (in this example). In the ACTN framework, however, it is also possible that the TNC

function is decomposed into MDSC and PNC - that is, the TNC may comprise hierarchy as needed to handle the multiple domains and various underlay technologies, whereas a PNC in ACTN is intended to be specific to at most a single underlay technology and (likely) to individual devices (or functional components).

Note that the details of potential implementations of everything that is below the TSC in Figure 1 are out of scope in this document - hence the specifics of the relationship between TNC and PNC, and the possibility that the MDSC and PNC may be combined are at most academically interesting in this context. Another way to view this is that, in the same way that ACTN might combine MDSC and PNC, the TSC might also directly include TNC functionality.

[RFC8453] also describes TE Network Slicing in the context of ACTN as a collection of resources that is used to establish a logically dedicated virtual network over one or more TE networks. In case of TE enabled underlying network, ACTN VN can be used as a base to realize the transport network slicing by coordination among multiple peer domains as well as underlay technology domains.

Figure 1 shows only one possible mapping as each ACTN component (or interface) in the figure may be a composed differently in other mappings, and the exact role of both components and subcomponents will not be always an exact analogy between the concepts used in this document and those defined in ACTN.

This is - in part - shown in a previous paragraph in this section where it is pointed out that the TNC may actually subsume some aspects of both the MDSC and PNC.

Similarly, in part depending on how "customer" is interpreted, CNC might merge some aspects of the higher level system and the TSC. As in the TNC/PNC case, this way of comparing ACTN to this work is not useful as the TSC and TSC NBI are the focus on this document.

## 5. Considerations

### 5.1. Monitoring

Transport slice realization needs to be instrumented in order to track how it is working, and it might be necessary to modify the transport slice as requirements change. Dynamic reconfiguration might be needed.

## 5.2. Security Considerations

Transport slices might use underlying virtualized networking. All types of virtual networking require special consideration to be given to the separation of traffic between distinct virtual networks, as well as some degree of protection from effects of traffic use of underlying network (and other) resources from other virtual networks sharing those resources.

For example, if a service requires a specific upper bound of latency, then that service can be degraded by added delay in transmission of service packets through the activities of another service or application using the same resources.

Similarly, in a network with virtual functions, noticeably impeding access to a function used by another transport slice (for instance, compute resources) can be just as service degrading as delaying physical transmission of associated packet in the network.

While a transport slice might include encryption and other security features as part of the service, consumers might be well advised to take responsibility for their own security needs, possibly by encrypting traffic before hand-off to a service provider.

## 5.3. Privacy Considerations

Privacy of transport network slice service consumers must be preserved. It should not be possible for one transport slice consumer to discover the presence of other consumers, nor should sites that are members of one transport slice be visible outside the context of that transport slice.

In this sense, it is of paramount importance that the system use the privacy protection mechanism defined for the specific underlying technologies used, including in particular those mechanisms designed to preclude acquiring identifying information associated with any transport slice consumer.

## 5.4. IANA Considerations

There are no requests to IANA in this framework document.

## 6. Acknowledgments

The entire TEAS NS design team and everyone participating in related discussions has contributed to this draft. Some text fragments in the draft have been copied from the [I-D.ietf-teas-enhanced-vpn], for which we are grateful.



Significant contributions to this document were gratefully received from the contributing authors listed in the "Contributors" section. In addition we would like to also thank those others who have attended one or more of the design team meetings, including:

- \* Aihua Guo
- \* Bo Wu
- \* Greg Mirsky
- \* Jeff Tantsura
- \* Kiran Makhijani
- \* Lou Berger
- \* Luis M. Contreras
- \* Rakesh Gandhi
- \* Ren Chen
- \* Sergio Belotti
- \* Shunsuke Homma
- \* Stewart Bryant
- \* Tomonobu Niwa
- \* Xuesong Geng

## 7. References

### 7.1. Normative References

[I-D.nsdt-teas-transport-slice-definition]  
Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "IETF Definition of Transport Slice", Work in Progress, Internet-Draft, draft-nsdt-teas-transport-slice-definition-03, 12 July 2020, <<http://www.ietf.org/internet-drafts/draft-nsdt-teas-transport-slice-definition-03.txt>>.

### 7.2. Informative References

[BBF-SD406]

Broadband Forum, ., "End-to-end network slicing", BBF SD-406 , n.d..

[I-D.ietf-teas-actn-yang]

Lee, Y., Zheng, H., Ceccarelli, D., Yoon, B., Dios, O., Shin, J., and S. Belotti, "Applicability of YANG models for Abstraction and Control of Traffic Engineered Networks", Work in Progress, Internet-Draft, draft-ietf-teas-actn-yang-05, 19 February 2020, <<http://www.ietf.org/internet-drafts/draft-ietf-teas-actn-yang-05.txt>>.

[I-D.ietf-teas-enhanced-vpn]

Dong, J., Bryant, S., Li, Z., Miyasaka, T., and Y. Lee, "A Framework for Enhanced Virtual Private Networks (VPN+) Services", Work in Progress, Internet-Draft, draft-ietf-teas-enhanced-vpn-05, 18 February 2020, <<http://www.ietf.org/internet-drafts/draft-ietf-teas-enhanced-vpn-05.txt>>.

[I-D.ietf-teas-te-service-mapping-yang]

Lee, Y., Dhody, D., Fioccola, G., WU, Q., Ceccarelli, D., and J. Tantsura, "Traffic Engineering (TE) and Service Mapping Yang Model", Work in Progress, Internet-Draft, draft-ietf-teas-te-service-mapping-yang-03, 8 March 2020, <<http://www.ietf.org/internet-drafts/draft-ietf-teas-te-service-mapping-yang-03.txt>>.

[I-D.openconfig-rtgwg-gnmi-spec]

Shakir, R., Shaikh, A., Borman, P., Hines, M., Lebsack, C., and C. Morrow, "gRPC Network Management Interface (gNMI)", Work in Progress, Internet-Draft, draft-openconfig-rtgwg-gnmi-spec-01, 5 March 2018, <<http://www.ietf.org/internet-drafts/draft-openconfig-rtgwg-gnmi-spec-01.txt>>.

[NGMN-NS-Concept]

NGMN Alliance, ., "Description of Network Slicing Concept", [https://www.ngmn.org/uploads/media/161010\\_NGMN\\_Network\\_Slicing\\_framework\\_v1.0.8.pdf](https://www.ngmn.org/uploads/media/161010_NGMN_Network_Slicing_framework_v1.0.8.pdf) , 2016.

[RFC2578]

McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIPv2)", STD 58, RFC 2578, DOI 10.17487/RFC2578, April 1999, <<https://www.rfc-editor.org/info/rfc2578>>.

- [RFC3412] Case, J., Harrington, D., Presuhn, R., and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3412, DOI 10.17487/RFC3412, December 2002, <<https://www.rfc-editor.org/info/rfc3412>>.
- [RFC3414] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", STD 62, RFC 3414, DOI 10.17487/RFC3414, December 2002, <<https://www.rfc-editor.org/info/rfc3414>>.
- [RFC3417] Presuhn, R., Ed., "Transport Mappings for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3417, DOI 10.17487/RFC3417, December 2002, <<https://www.rfc-editor.org/info/rfc3417>>.
- [RFC4208] Swallow, G., Drake, J., Ishimatsu, H., and Y. Rekhter, "Generalized Multiprotocol Label Switching (GMPLS) User-Network Interface (UNI): Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Support for the Overlay Model", RFC 4208, DOI 10.17487/RFC4208, October 2005, <<https://www.rfc-editor.org/info/rfc4208>>.
- [RFC4397] Bryskin, I. and A. Farrel, "A Lexicography for the Interpretation of Generalized Multiprotocol Label Switching (GMPLS) Terminology within the Context of the ITU-T's Automatically Switched Optical Network (ASON) Architecture", RFC 4397, DOI 10.17487/RFC4397, February 2006, <<https://www.rfc-editor.org/info/rfc4397>>.
- [RFC5212] Shiomoto, K., Papadimitriou, D., Le Roux, JL., Vigoureux, M., and D. Brungard, "Requirements for GMPLS-Based Multi-Region and Multi-Layer Networks (MRN/MLN)", RFC 5212, DOI 10.17487/RFC5212, July 2008, <<https://www.rfc-editor.org/info/rfc5212>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<https://www.rfc-editor.org/info/rfc5440>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.

- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC7926] Farrel, A., Ed., Drake, J., Bitar, N., Swallow, G., Ceccarelli, D., and X. Zhang, "Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks", BCP 206, RFC 7926, DOI 10.17487/RFC7926, July 2016, <<https://www.rfc-editor.org/info/rfc7926>>.
- [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>>.
- [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>>.
- [RFC8454] Lee, Y., Belotti, S., Dhody, D., Ceccarelli, D., and B. Yoon, "Information Model for Abstraction and Control of TE Networks (ACTN)", RFC 8454, DOI 10.17487/RFC8454, September 2018, <<https://www.rfc-editor.org/info/rfc8454>>.
- [TS23501] 3GPP, ., "System architecture for the 5G System (5GS)", 3GPP TS 23.501 , 2019.
- [TS28530] 3GPP, ., "Management and orchestration; Concepts, use cases and requirements", 3GPP TS 28.530 , 2019.

#### Contributors

The following authors contributed significantly to this document:

Jari Arkko  
Ericsson

Email: [jari.arkko@piuha.net](mailto:jari.arkko@piuha.net)

Dhruv Dhody

Huawei, India

Email: dhruv.ietf@gmail.com

Reza Rokui  
Nokia

Email: reza.rokui@nokia.com

Xufeng Liu

Email: xufeng.liu.ietf@gmail.com

Jie Dong  
Huawei

Email: jie.dong@huawei.com

#### Authors' Addresses

Eric Gray (editor)  
Ericsson

Email: eric.gray@ericsson.com

John Drake (editor)  
Juniper Networks

Email: jdrake@juniper.net

teas  
Internet-Draft  
Intended status: Informational  
Expires: January 13, 2021

R. Rokui  
Nokia  
S. Homma  
NTT  
K. Makhijani  
Futurewei  
LM. Contreras  
Telefonica  
J. Tantsura  
Apstra, Inc.  
July 12, 2020

IETF Definition of Transport Slice  
draft-nsdt-teas-transport-slice-definition-03

Abstract

This document describes the definition of a slice in the transport networks and its characteristics. The purpose here is to bring clarity and a common understanding of the transport slice concept and describe related terms and their meaning. It explains how transport slices can be used in combination with end to end network slices, or independently.

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 . . . . .	2
1.1. Rationale . . . . .	3
2. Terms and Abbreviations . . . . .	3
3. Definition and Scope of Transport Slice . . . . .	4
4. Transport Slice System Characteristics . . . . .	5
4.1. Service Level Objectives for Transport Slices . . . . .	5
4.1.1. Minimal Set of SLOs . . . . .	5
4.1.2. Other Objectives . . . . .	7
4.2. Transport Slice Endpoints . . . . .	8
4.2.1. Transport Slice Connectivity Types . . . . .	9
4.3. Vertical Composition of Transport Slice . . . . .	9
4.4. Horizontal Composition of Transport Slice . . . . .	11
5. Transport Slice Structure . . . . .	11
5.1. Stakeholders . . . . .	13
5.2. Transport Slice Controller Interfaces . . . . .	14
5.3. Transport slice Realization . . . . .	15
6. Relationship with End-to-End Network Slicing . . . . .	15
7. Security Considerations . . . . .	17
8. IANA Considerations . . . . .	17
9. Acknowledgment . . . . .	17
10. Informative References . . . . .	17
Appendix A. Discussions . . . . .	19
A.1. On Isolation Requirements In a Transport Slice . . . . .	19
Authors' Addresses . . . . .	20

## 1. Introduction

A number of use cases benefit from establishing network connectivity providing transport and assurance of a specific set of network resources. In this document, as detailed in the subsequent sections, we refer to this connectivity and resource commitment as the transport slice. Services that might benefit from the transport slices include but not limited to:

- o 5G services (e.g. eMBB, URLLC, mMTC) (See [TS.23.501-3GPP])

- o Network wholesale services
- o Network infrastructure sharing among operators
- o NFV connectivity and Data Center Interconnect

This document defines the concept of transport slices that provide connectivity with a specific commitment of network resources between a number of end points over a shared network infrastructure.

### 1.1. Rationale

Transport slices are created and managed within the scope of one or more underlying network technologies (e.g., IP, MPLS, optical). Transport slices are expected to enable a diverse set of applications that have different requirements to coexist on the same network infrastructure.

Transport slice is described as a construct that specifies connectivity requirements, emphasizing on assurance of those requirements. Transport slice is unaware of the underlying infrastructure connectivity (hence, the term "transport"). The types of underlying networking technologies can be based on any combination of IP, Ethernet, MPLS, and optical technologies. Transport slices also include specification of resources related to network functions required by customer applications.

Traditionally, VPNs have focussed on segmentation, i.e., creation and management of the private networks. They are bound to a specific traffic type and are technology specific. In contrast, transport slices concern with the assurance of resources required from the network and provide a common user interface for describing those resources. A service provider can use many aspects of the VPNs to build the transport slices.

Transport slices relate to a more general topic of network slicing. It is not the goal of this document to define this broader concept, but in general, it is to identify the methodology to describe the logical (or abstract) partitioning of network resources associated with a service or an application.

## 2. Terms and Abbreviations

The terms and abbreviations used in this document are listed below.

- o E2E NS: End to End Network Slice
- o TS: Transport Slice



- o TSC: Transport Slice Controller
- o EP: Endpoint
- o EU: End User
- o NBI: NorthBound Interface
- o SBI: SouthBound Interface
- o SLI: Service Level Indicator A well defined quantitative measure of some aspect of the level of service that is provided.
- o SLO: Service Level Objective A target value or range of values for a service level that is measured by an SLI. A natural structure for SLOs is thus  $SLI \leq target$ , or lower bound  $\leq SLI \leq$  upper bound.
- o SLA: Service Level Agreement An explicit or implicit contract with the end users that includes consequences of meeting (or missing) the SLOs they contain.

The above terminology is described in greater detail in the remainder of this document.

### 3. Definition and Scope of Transport Slice

The definition of a transport slice is as follows:

"A transport slice is a logical network topology connecting a number of endpoints with a set of shared or dedicated network resources, that are used to satisfy specific Service Level Objectives (SLOs)".

The text below describes transport slices in more details.

Transport slice specification is technology-agnostic, and the means for transport slice realization can be chosen depending on several factors such as: service requirements, specifications or capabilities of underlying infrastructure. The structure and different characteristics of transport slices are described in the following sections.

The term "transport" in transport slice is derived from the definition of Transport Network in the section 1.3.1 of [RFC5921] : A Transport Network provides transparent transmission of user traffic between attached client devices by establishing and maintaining point-to-point or point-to-multipoint connections between such devices. "Slice" refers to a set of characteristics that separate

one type of user-traffic from other types. Transport slice assumes that an underlying transport network is capable of changing the configurations of the network devices on demand, through in-band signaling or via controller(s) and to provide transport transmissions with fulfilling all or some of SLOs to all of the traffic in the slice or to specific flows.

#### 4. Transport Slice System Characteristics

The following subsections describe the characteristics needed for support of transport slices.

##### 4.1. Service Level Objectives for Transport Slices

A transport slice is defined in terms of several quantifiable characteristics or service level objectives (SLOs). These objectives define a set of network resource parameters or values necessary to provide a service as requested for a given transport slice. SLOs do not describe 'how' the transport slices will be implemented or realized in the underlying network layers. Instead, they are defined in terms of dimensions of operations (time, capacity, etc.), availability and other attributes. A transport slice can have one or more SLOs associated with it, all SLO's combined to form an SLA. The SLO values are defined unidirectionally and for specific subsets of two or more endpoints (i.e. for a subset of connections in transport slice).

The SLOs and values associated with them that are exposed to the end user, are in the form of Service Level Indicators (SLIs). If for example the range of latencies a network can provide is 50ms-100ms, then this would be the range of values the end user should be able to request, it would be as low as 50ms or as high as 100ms or anything in between. The values of requested SLOs should always be in the range of values supported. The underlying networks must provide means to monitor and measure the performance of transport slices against the SLOs requested and verify that they are being met. Some SLOs can be measured directly through a collection of metrics and statistics from the network (commonly known as 'telemetry'), while others are deduced from measurable objectives and may require additional tools or mechanisms to measure their target values.

##### 4.1.1. Minimal Set of SLOs

This document defines a minimal set of SLOs and later systems or standards could extend this set and define more SLOs. For example, we included Guaranteed bandwidth which is the minimum requested bandwidth for the transport slice. The later standard might define other SLOs related to bandwidth if needed.

Accordingly, SLOs can be categorized in to 'Directly Measurable Objectives' or 'Indirectly Measurable Objectives' as follows:

Some of the 'Directly Measurable Objectives' are:

- o Guaranteed Minimum Bandwidth
- o Guaranteed Maximum Latency
- o Maximum permissible delay variation
- o Maximum permissible packet loss rate
- o Availability
- o Other objectives could be specified

Some of the 'Indirectly Measurable Objectives' are:

- o Security
- o others objectives such as geographical restrictions, maximum occupancy level, etc. could be specified

The definition of these objectives are as follows:

- o **Guaranteed Minimum Bandwidth:** Minimum guaranteed bandwidth between two endpoints at any time. The bandwidth is measured in data rate units of bits per second and is measured unidirectionally.
- o **Guaranteed Maximum Latency:** Upper bound of network latency when transmitting between two endpoints. The latency is measured in terms of network characteristics (excluding application-level latency). [RFC2681] and [RFC7679] discuss round trip times and one-way metrics, respectively.
- o **Maximum permissible delay variation:** Packet delay variation (PDV) as defined by [RFC3393], is measured by the difference in the one-way delay between sequential packets in a flow. Minimizing variations in the delay is important for real-time applications.
- o **Maximum permissible packet loss rate:** is defined by the ratio of packets dropped to packets transmitted between two endpoints. See [RFC7680]
- o **Availability:** is defined as the ratio of uptime to total\_time(uptime+downtime), where uptime is the time the

transport slice is available in accordance with the SLOs associated with it.

- o Security: This objective may request for encryption [RFC4303] between two end-points explicitly to meet architecture recommendations as in [TS33.210] or for compliance with [HIPAA] [PCI]. Other security requests may be made as specified in [draft-ietf-i2nsf-capability].

\* Note: Security violations are not directly observable and cannot be measured as quantifiable metrics. Still, the user of the transport slice should be able to request certain criteria for compliance and identify exceptions and unexpected traffic. For this purpose [i2nsf-nsf-monitoring-data-model] can be leveraged.

#### 4.1.2. Other Objectives

Additional objectives, such as certain geographical restrictions or well defined domains that a slice may transit may be necessary.

Optionally, when the customer is traffic aware, other traffic specific characteristics may be provided. These include for example, MTU, traffic-type (e.g., IPv4, IPv6, Ethernet or unstructured), or a higher-level behavior to process traffic according to user-application (which may be realized using network functions).

Maximal occupancy for a transport slice should be provided. Since it carries traffic for multiple flows between the two endpoints, the objectives should also say if they are for the entire connection, group of flows or on per flow basis. Maximal occupancy should specify the scale of the flows (i.e. maximum number of accommodatable flows) and optionally a maximum number of countable resource units, e.g IP or MAC addresses a slice might consume.

With these objectives incorporated, a customer sees transport slice as a dedicated network for its exclusive use. Achieving this may require different types of isolation techniques in provider networks as described in Appendix A.1.

Additional description of slice attributes is covered in a broader context of 'Generic Network Slice Template' in [I-D.contreras-teas-slice-nbi].

## 4.2. Transport Slice Endpoints

The transport slice endpoints are the conceptual entities that perform any required conversion, or adaptation, and forwarding of the user traffic. The characteristics of the transport slice endpoints (TSE) are:

- o They are conceptual points of connection of a network function, device or application to the transport slice
- o They are identified in a request provided by the customer of transport slice (i.e. higher level operation systems) during the creation of the transport slice
- o They are associated with a device, application and/or network function nodes. A non-exhaustive list of such nodes are routers, switches, firewalls, WAN, 4G/5G RAN nodes, 4G/5G Core nodes, application acceleration, Deep Packet Inspection (DPI), server load balancers, NAT44 [RFC3022], NAT64 [RFC6146], HTTP header enrichment functions, and TCP optimizers
- o A TSE is identified by its associated node (its IP address, name, ID, etc.), a unique identifier and/or a unique name and other data. A non-exhaustive list of other data includes IP address (v4 or v6), VLAN, port, connectivity type (P2P, P2MP, MP2MP). TBD for more

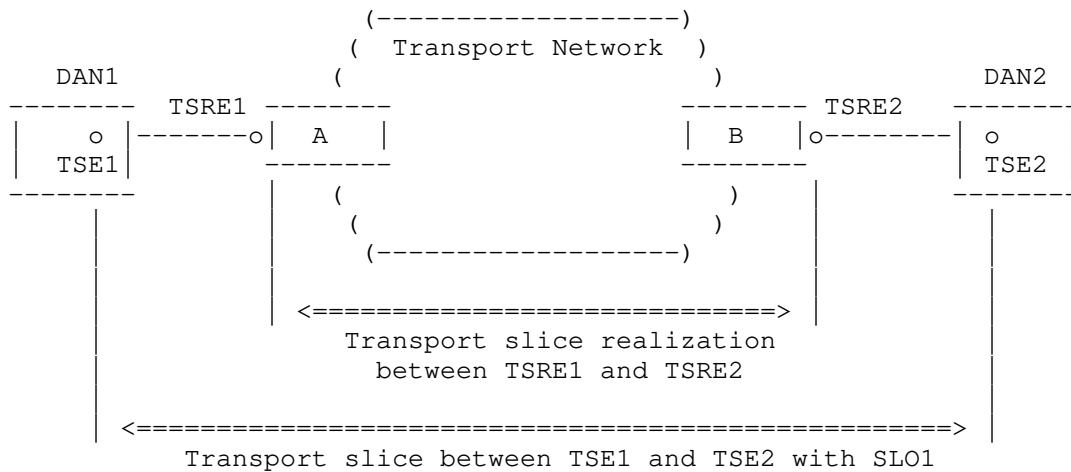
Note that the TSE is different from access points (AP) defined in [RFC8453] as an AP is a logical identifier to identify the shared link between the customer and the operator where as TSE is an identifier of an endpoint. Also TSE is different from TE Link Termination Point (LTP) defined in [I-D.ietf-teas-yang-te-topo] as it is a conceptual point of connection of a TE node to one of the TE links on a TE node.

The TSE is similar to the Termination Point (TP) defined in [RFC8345] and can contain more attributes. TSE could be modeled by augmenting the TP model.

There is another type of the endpoints called "Transport Slice Realization endpoints (TSREs)". These endpoints are allocated and assigned by the network controller during the realization of a transport slice and are technology-specific, i.e. they depend on the network technology used during the transport slice realization. They are identified by a node and some associated data. A non-exhaustive list of nodes containing TSREs are routers, switches, PON nodes, Wireless nodes and Optical devices.

Note that there will be a mapping between TSE and TSRE on Transport Slice Controller (TSC). When TSC receives a request via its NBI to create a transport slice between multiple TSEs, it will send the request via its SBI to realize the transport slice. The TSRE will be notified by network controller during TS realization to enable mapping between TSREs and the TSEs.

Figure 1 shows an example of a transport slice and its realization between multiple TSEs and TSREs.



Legend:

DAN: Device, application and/or network function

Figure 1: A transport slice between TSEs and its realization between TSREs

#### 4.2.1. Transport Slice Connectivity Types

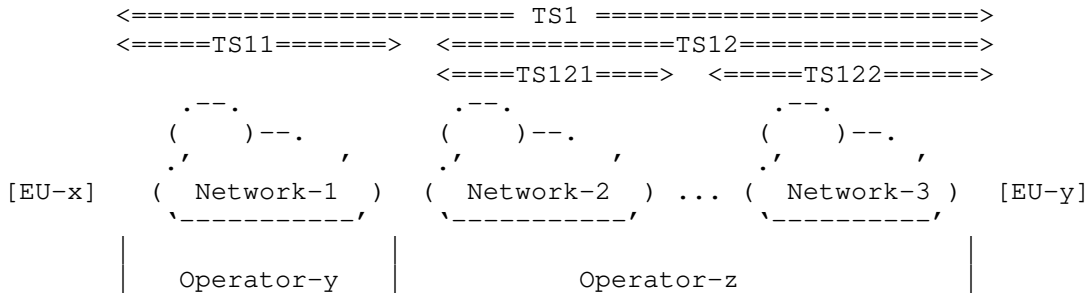
The transport slices connection types can be point to point (P2P), point to multipoint (P2MP), multi-point to point (MP2P), or multi-point to multi-point (MP2MP). The transport slice connection type will requested by the higher level operation system.

#### 4.3. Vertical Composition of Transport Slice

Transport slice may follow a hierarchical relationship to provide a vertical structure to it. This is used for composing multi-layer slices in which each layer provides an abstraction, as well as an independent monitoring, performance, control and management of the

resources. The vertical transport slice characteristic could be used in 2 forms:

- o The Transport slice itself where it represents a hierarchy of abstracted transport slices. In this case, the realization will be done just once with a particular technology. Thus, the lowest transport slice in the hierarchy that can not be decomposed further will be one to one mapping to its instance of the realized transport slice.
- o Each layer (physical, datalink, or IP) has its own set of resources that can be provided to the upper layer as a transport slice. Thus, transport slice at one layer is used by the layer above. This type of multi-layer vertical transport slice associates resources at different layers. For example, an IP transport slice would utilize one or more optical transport slice. In this case, the realization will be done for a particular technology at that particular layer. Thus, the lowest transport slice in this type of hierarchy that can not be decomposed further will be an instance of realized physical layer transport slice.



Legend:

- TSnnn: Level 3 vertical transport slice nnn
- TSnn: Level 2 vertical transport slice nn
- TSn: Level 1 transport Slice n

Figure 2: Transport Slice Vertical and Horizontal Composition

Figure 2 shows the transport slice hierarchy. Slices TS11 and TS12 are composed together to form TS1 that is the top level transport slice definition, TS121 and TS122 collectively define TS12. The SLO for bandwidth guarantee will be shared and latency guarantee will be split into latency in networks 2 and 3. To emphasize the hierarchical structure, consider Network-2 and Network-3 are in the same administrative domain but use different transport technologies respectively. Then instead of presenting 2 transport slices,

Operator-z can expose only one transport slice TS12 abstracting the underlying transport technology details.

Note: The specification to connect TS121 and TS122 are similar to those connecting TS12 and TS11.

#### 4.4. Horizontal Composition of Transport Slice

In contrast, horizontal transport slices enable the composition of multiple realized transport slices. Since transport slices are not necessarily a single encapsulation tunnel and may traverse through different data planes, each realized transport slice will require a stitching, interworking or mapping function. These stitching functions can be viewed as a type of intermediate network function endpoints. For instance in Figure 2, TS11 and TS12 are horizontal transport slices. If we assume that TS11 is an L2 tunnel and TS12 is an SRV6 based path, then a 'Service type EP' (not shown in the figure) is needed for translation.

Author's notes: This service type EP is a new type of transport slice specific service function. We may call it transport slice gateway.

#### 5. Transport Slice Structure

A transport slice is a set of connections among various endpoints to form a logical network that meets the SLOs agreed upon.



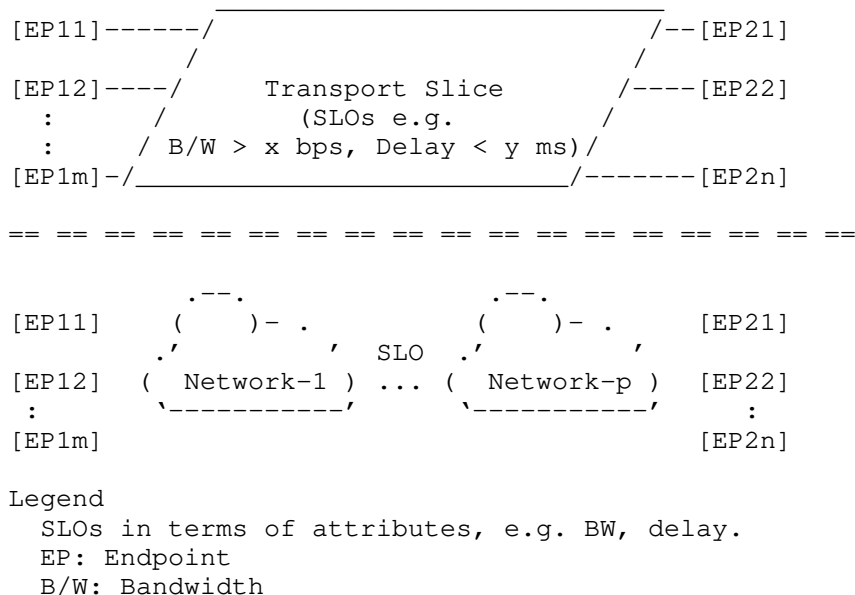


Figure 3: Transport slice

Figure 3 illustrates a case where a transport slice provides connectivity between a set of endpoints pairs with specific characteristics for each SLO (e.g. guaranteed minimum bandwidth of x bps and guaranteed delay of less than y ms). The endpoints may be distributed in the underlay networks, and a transport slice can be deployed across multiple network domains. Also, the endpoints on the same transport slice may belong to the same address space.

Transport slices involve both customer’s and provider’s views. A customer ‘describes’ its requirements in terms of connectivity with specific SLOs. Provider networks address those requirements through ‘transport slice realization’ (its implementation) using provider network specific technologies.

A transport slice is requested from an entity (such as an orchestrator or a system-wide controller) performing broader service or application specific functions. The interface from such an entity should express the needed connectivity in a technology-agnostic way and donot need to recognize configurations based on the technologies (e.g. being more declarative than imperative). The request to instantiate a transport slice is only represented with some indicators such as SLOs based on which the underlying technologies are selected and managed.

Often, in other SDOs the term sub-slice or slice-subnet comes up. Some of those are mapped to transport network requirements in the form of a transport slice. Within the scope of transport slices (w.r.t. the IP/MPLS based transport networks) there are no definitions for 'sub-slice' or 'slice subnets'. 'Transport slice' term universally represents SLO and connectivity requirements from the transport networks.

Furthermore, the structure of transport slices may be layered vertically or composed horizontally, i.e. operationally, a transport slice maybe decomposed in two or more transport slices which are then independently realized and managed. This is further described in Section 4.3.

### 5.1. Stakeholders

A transport slice and its realization involves the following stakeholders and it is relevant to define them for consistent terminology.

**Customer or User:** A customer is a user of a transport slice. Customers may request monitoring of associated resources or specific changes. A user may either directly manage its service by interfacing with the transport slice controller or indirectly through an orchestrator.

**Orchestrator:** An orchestrator is an entity that composes different services, resource and network requirements. It interfaces with the transport slice controllers.

**Transport Slice Controller (TSC):** It realizes a transport slice in the network, maintains and monitors the run-time state of resources and topologies associated with it. A well-defined interface is needed between different types of transport slice controllers and different types of orchestrators. A transport slice operator (or slice operator for short) manages one or more transport slices using the Transport Slice Controller(s).

**Transport Network Controller:** is a form of network infrastructure controller that offers network resources to TSC to realize a particular transport slice. These may be existing network controllers associated with one or more specific technologies that may be adapted to the function of realizing transport slices in a network.

### 5.2. Transport Slice Controller Interfaces

The interworking and interoperability among the different stakeholders to provide common means of provisioning, operating and monitoring the transport slices is a mandatory requirement. The following communication interfaces are identified (see Figure 4).

**TSC Northbound Interface (NBI):** The TSC Northbound Interface is an interface between a higher level operation system, e.g. 'E2E network slice orchestrator' and the 'Transport slice controller'. It is a technology agnostic interface. Over this NBI, slice characteristics and other requirements can be communicated to TSC and the operational state of a transport slice may be requested.

**TSC Southbound Interface (SBI):** The TSC Southbound Interface is an interface between 'Transport slice controller (TSC)' and network controller(s). These interfaces are technology-specific and utilize many of the network models.

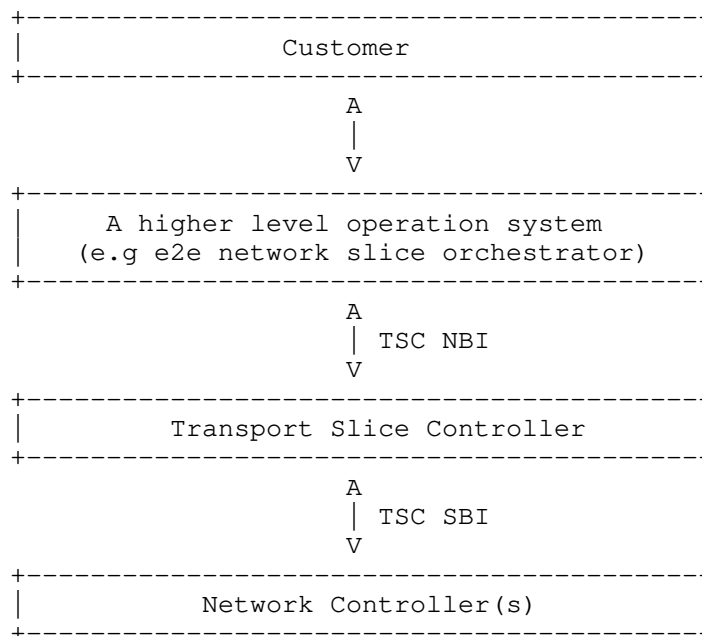


Figure 4: Interface of Transport Slice Controller

### 5.3. Transport slice Realization

Realization of a Transport Slice is a mapping of underlying infrastructure with its definition. It is a technology specific entity that is created and maintained over its southbound interfaces. The Network controller(s) export the connectivity and resource mappings to the TSC. The network controller abstracts the details of underlying resources from the TSC.

The realization can be achieved in the form of either physical or logical connectivity through VPNs, a variety of tunneling technologies such as Segment Routing, SFC, etc. Accordingly, endpoints may be realized as physical or logical service or network functions.

## 6. Relationship with End-to-End Network Slicing

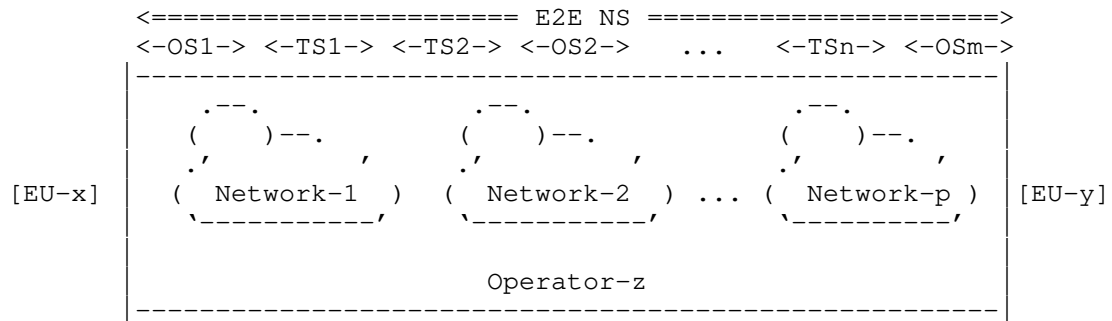
An end-to-end (E2E) network slice is a complete logical network that provides a service in its entirety with a specific assurance to the customer. A transport slice concerns with those assurance aspects only within the transport networks. Consider Figure 5, where a network operator has an E2E network slice that traverses multiple technology-specific networks. Each of these networks might use any number of technologies, including but not limited to IP, MPLS, Fiber-Optics (e.g. WDM, DWDM), Passive Optical Networking (PON), Microwave, etc.

Each of these networks includes multiple (physical or virtual) nodes and may also provide network functions beyond simply carrying of technology-specific protocol data units. The types of nodes used in any of these networks may include:

- o Packet/frame processing nodes (e.g., Routers, Switches)
- o Application servers
- o Service Functions (e.g., Firewall, Loadbalancer)
- o Radio Access Network (RAN) components
- o Mobile Core components
- o Microwave transceivers
- o Optical repeaters
- o etc.

Each network may support different technologies and an E2E network slice is a combination of these networks. As an example:

- o Network 1 might contain multiple 5G RAN nodes connected to a few Cell Site Gateways (CSG) routers.
- o Network 2 might have one or more layer-3 routers and layer-2 switches which may run on top of an optical network.
- o Network 3 might have a number of 5G RAN nodes connected to Passive Optical Network (PON) switches.



Legend:

- E2E NS: End-to-end network slice
- TSn: Transport Slice n
- OSm: Other Slice m
- EU-x: End User-x
- EU-y: End User-y

Figure 5: E2E network slice

When operator-z creates a specific E2E network slice, it may create one or more of transport slices and other slices (application logic or other system functions).

An independent E2E logical network (called E2E network slice) is created for a service (e.g. CCTV, autonomous driving, HD map, etc.) with a specific network SLOs, e.g. a secure connection with an E2E latency less than 5ms, from End User-x (EU-x) to End User-y (EU-y). EU-x maybe a 5G user equipment such as an infotainment unit in a car, CCTV, or a car for autonomous driving, etc. and EU-y in 5G is 5G application server, IMS, etc.

In Figure 5, "E2E NS" is that logical network with requested SLO between EU-x to EU-y and is associated with a customer and a specific service type.

#### 7. Security Considerations

Not applicable in this memo.

#### 8. IANA Considerations

This memo includes no request to IANA.

#### 9. Acknowledgment

The entire TEAS NS design team and everyone participating in those discussion has contributed to this draft. Particularly, Eric Gray, Xufeng Liu, Jie Dong, and Jari Arkko for a thorough review among other contributions.

#### 10. Informative References

[HIPAA] HHS, "Health Insurance Portability and Accountability Act - The Security Rule", February 2003, <<https://www.hhs.gov/hipaa/for-professionals/security/index.html>>.

[I-D.contreras-teas-slice-nbi]  
Contreras, L., Homma, S., and J. Ordonez-Lucena, "Considerations for defining a Transport Slice NBI", draft-contreras-teas-slice-nbi-01 (work in progress), March 2020.

[I-D.ietf-teas-enhanced-vpn]  
Dong, J., Bryant, S., Li, Z., Miyasaka, T., and Y. Lee, "A Framework for Enhanced Virtual Private Networks (VPN+) Services", draft-ietf-teas-enhanced-vpn-05 (work in progress), February 2020.

[I-D.ietf-teas-sf-aware-topo-model]  
Bryskin, I., Liu, X., Lee, Y., Guichard, J., Contreras, L., Ceccarelli, D., and J. Tantsura, "SF Aware TE Topology YANG Model", draft-ietf-teas-sf-aware-topo-model-05 (work in progress), March 2020.

- [I-D.ietf-teas-yang-te-topo] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", draft-ietf-teas-yang-te-topo-22 (work in progress), June 2019.
- [I-D.nsdt-teas-ns-framework] Gray, E. and J. Drake, "Framework for Transport Network Slices", draft-nsdt-teas-ns-framework-02 (work in progress), March 2020.
- [NFVGST] ETSI, "NFVI Compute and Network Metrics Specification", February 2018, <[https://www.etsi.org/deliver/etsi\\_gs/NFV-TST/001\\_099/008/02.04.01\\_60/gs\\_nfv-tst008v020401p.pdf](https://www.etsi.org/deliver/etsi_gs/NFV-TST/001_099/008/02.04.01_60/gs_nfv-tst008v020401p.pdf)>.
- [PCI] PCI Security Standards Council, "PCI DSS", May 2018, <<https://www.pcisecuritystandards.org>>.
- [RFC2681] Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, DOI 10.17487/RFC2681, September 1999, <<https://www.rfc-editor.org/info/rfc2681>>.
- [RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, DOI 10.17487/RFC3022, January 2001, <<https://www.rfc-editor.org/info/rfc3022>>.
- [RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, DOI 10.17487/RFC3393, November 2002, <<https://www.rfc-editor.org/info/rfc3393>>.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<https://www.rfc-editor.org/info/rfc4303>>.
- [RFC5921] Bocci, M., Ed., Bryant, S., Ed., Frost, D., Ed., Levrau, L., and L. Berger, "A Framework for MPLS in Transport Networks", RFC 5921, DOI 10.17487/RFC5921, July 2010, <<https://www.rfc-editor.org/info/rfc5921>>.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <<https://www.rfc-editor.org/info/rfc6146>>.

- [RFC7679] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <<https://www.rfc-editor.org/info/rfc7679>>.
- [RFC7680] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Loss Metric for IP Performance Metrics (IPPM)", STD 82, RFC 7680, DOI 10.17487/RFC7680, January 2016, <<https://www.rfc-editor.org/info/rfc7680>>.
- [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <<https://www.rfc-editor.org/info/rfc8345>>.
- [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>>.
- [TS.23.501-3GPP]  
3rd Generation Partnership Project (3GPP), "3GPP TS 23.501 (V16.2.0): System Architecture for the 5G System (5GS); Stage 2 (Release 16)", September 2019, <[http://www.3gpp.org/ftp//Specs/archive/23\\_series/23.501/23501-g20.zip](http://www.3gpp.org/ftp//Specs/archive/23_series/23.501/23501-g20.zip)>.
- [TS33.210]  
3GPP, "3G security; Network Domain Security (NDS); IP network layer security (Release 14).", December 2016, <<https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2279>>.

## Appendix A. Discussions

### A.1. On Isolation Requirements In a Transport Slice

Transport slices are perceived as if slice was provisioned for the customer as a dedicated network with specific SLOs. These committed SLOs for a given customer should be maintained during the lifetime of the slice, even in the face of potential disruptions. Such disruptions include sudden traffic volume changes either from the customer itself or others, equipment failures in the service provider network, and various misbehaviors or attacks.

The service provider needs to ensure that its network can provide the requested slices with the availability agreed with its customers. Some of the main technical approaches to ensuring guarantees are



about network planning, managing capacity, prioritizing, policing or shaping customer traffic, selecting dedicated resources, and so on.

One term that has commonly been used in this context is "isolation" and is also discussed in the [I-D.ietf-teas-enhanced-vpn].

A transport slice customer may ask for traffic separation, selection of dedicated resources, or interference avoidance from other traffic. The term "isolation" can refer to any or all of them. For instance, dedicated resources can help assure that traffic in other slices does not affect a given slice. Similarly, VPN technologies can provide traffic separation, and interference avoidance may be provided by mechanisms such as technical approaches mentioned in the previous paragraph (network planning, capacity management, etc). Moreover, these are some of the examples of a particular realization of the requirement for guarantees; other mechanisms may also be used.

#### Authors' Addresses

Reza Rokui  
Nokia  
Canada

Email: reza.rokui@nokia.com

Shunsuke Homma  
NTT  
Japan

Email: shunsuke.homma.ietf@gmail.com

Kiran Makhijani  
Futurewei  
USA

Email: kiranm@futurewei.com

Luis M. Contreras  
Telefonica  
Spain

Email: luismiguel.contrerasmurillo@telefonica.com

Jeff Tantsura  
Apstra, Inc.

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

TEAS  
Internet-Draft  
Intended status: Standards Track  
Expires: August 19, 2020

Shaofu. Peng  
Ran. Chen  
Gregory. Mirsky  
ZTE Corporation  
Fengwei. Qin  
China Mobile  
February 16, 2020

Packet Network Slicing using Segment Routing  
draft-peng-teas-network-slicing-03

Abstract

This document presents a mechanism aimed at providing a solution for network slicing in the transport network for 5G services. The proposed mechanism uses a unified administrative instance identifier to distinguish different virtual network resources for both intra-domain and inter-domain network slicing scenarios. Combined with the segment routing technology, the mechanism could be used for both best-effort and traffic engineered services for tenants.

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 August 19, 2020.

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 . . . . .	3
2. Architecture of TN Slicing . . . . .	3
2.1. Key Technologies of Transport slice . . . . .	5
3. Slicing Requirements . . . . .	6
3.1. Dedicated Virtual Networks . . . . .	6
3.2. End-to-End Slicing . . . . .	6
3.3. Unified NSI . . . . .	6
3.4. Traffic Engineering . . . . .	7
3.5. Summarized Requirements . . . . .	7
4. Conventions Used in This Document . . . . .	8
5. Overview of Existing Identifiers . . . . .	8
5.1. AG and EAG Bit . . . . .	8
5.2. Multi-Topology Identifier . . . . .	9
5.3. SR Policy Color . . . . .	9
5.4. Flex-algorithm Identifier . . . . .	9
5.5. New Slice-based Identifier Introduced . . . . .	10
6. Overview of AII-based Mechanism . . . . .	10
6.1. Physical Network Partition by AII . . . . .	11
6.2. Path within AII specific Slice . . . . .	11
6.2.1. SR-BE Path within AII specific Slice . . . . .	11
6.2.2. SR-TE Path within AII specific Slice . . . . .	12
6.3. Traffic Steering to SR policy within Slice . . . . .	12
6.4. Simple Variant of AII-based Slicing Scheme . . . . .	13
7. Resource Allocation per AII . . . . .	13
7.1. L3 Link Resource AII Configuration . . . . .	13
7.2. L2 Link Resource AII Configuration . . . . .	14
7.3. Node Resource AII Configuration . . . . .	14
7.4. Service Function Resource AII Configuration . . . . .	15
8. E2E Slicing with Centralized Mode . . . . .	15
9. E2E Slicing with Distributed Mode . . . . .	16
10. Combined with SR Flex-algorithm for Stack Depth Optimization	16
10.1. Flex-algo Using AII Criteria . . . . .	17
10.2. Best-effort Color Template Mapping to Flex-algo . . . . .	17
10.3. Traffic Engineering Color Template Mapping to Flex-algo	17
11. Network Slicing Examples . . . . .	17
11.1. Intra-domain Network Slicing Example . . . . .	18
11.1.1. Best-effort Service over Network Slice Example . . . . .	18
11.1.2. TE Service over Network Slice Example . . . . .	18
11.1.3. TE Service over Network Slice with Flex-algo Example	19
11.2. Inter-domain Network Slicing via BGP-LS Example . . . . .	19

11.2.1.	Best-effort Service Example . . . . .	19
11.2.2.	TE Service Example . . . . .	20
11.2.3.	TE Service Using Flex-algo Example . . . . .	20
11.3.	Inter-domain Network Slicing via BGP-LU Example . . . . .	21
12.	Implementation Suggestions . . . . .	21
12.1.	SR-MPLS . . . . .	21
12.2.	SRv6 . . . . .	22
13.	IANA Considerations . . . . .	23
14.	Security Considerations . . . . .	24
15.	Acknowledgements . . . . .	24
16.	Normative references . . . . .	24
	Authors' Addresses . . . . .	26

## 1. Introduction

According to 5G context, network slicing is the collection of a set of technologies to create specialized, dedicated logical networks as a service (NaaS) in support of network service differentiation and meeting the diversified requirements from vertical industries. Through the flexible and customized design of functions, isolation mechanisms, and operation and management (O&M) tools, network slicing is capable of providing dedicated virtual networks over a shared infrastructure. A Network Slice Instance (NSI) is the realization of network slicing concept. It is an E2E logical network, which comprises of a group of network functions, resources, and connection relationships. An NSI typically covers multiple technical domains, which include a terminal, access network (AN), transport network (TN) and a core network (CN), as well as a DC domain that hosts third-party applications from vertical industries. Different NSIs may have different network functions and resources. They may also share some of the network functions and resources.

For a transport network, network slicing requires the underlying network to support partitioning of the network resources to provide the client with dedicated (private) networking, computing, and storage resources drawn from a shared pool. The slices may be seen as virtual networks.

## 2. Architecture of TN Slicing

Relationship with NS Design Team:

The current scope of NS design team will focus on the framework of the TN Slice. We would like to make some contributions of it, and will sent this section to the NS Design Team for dicussion.

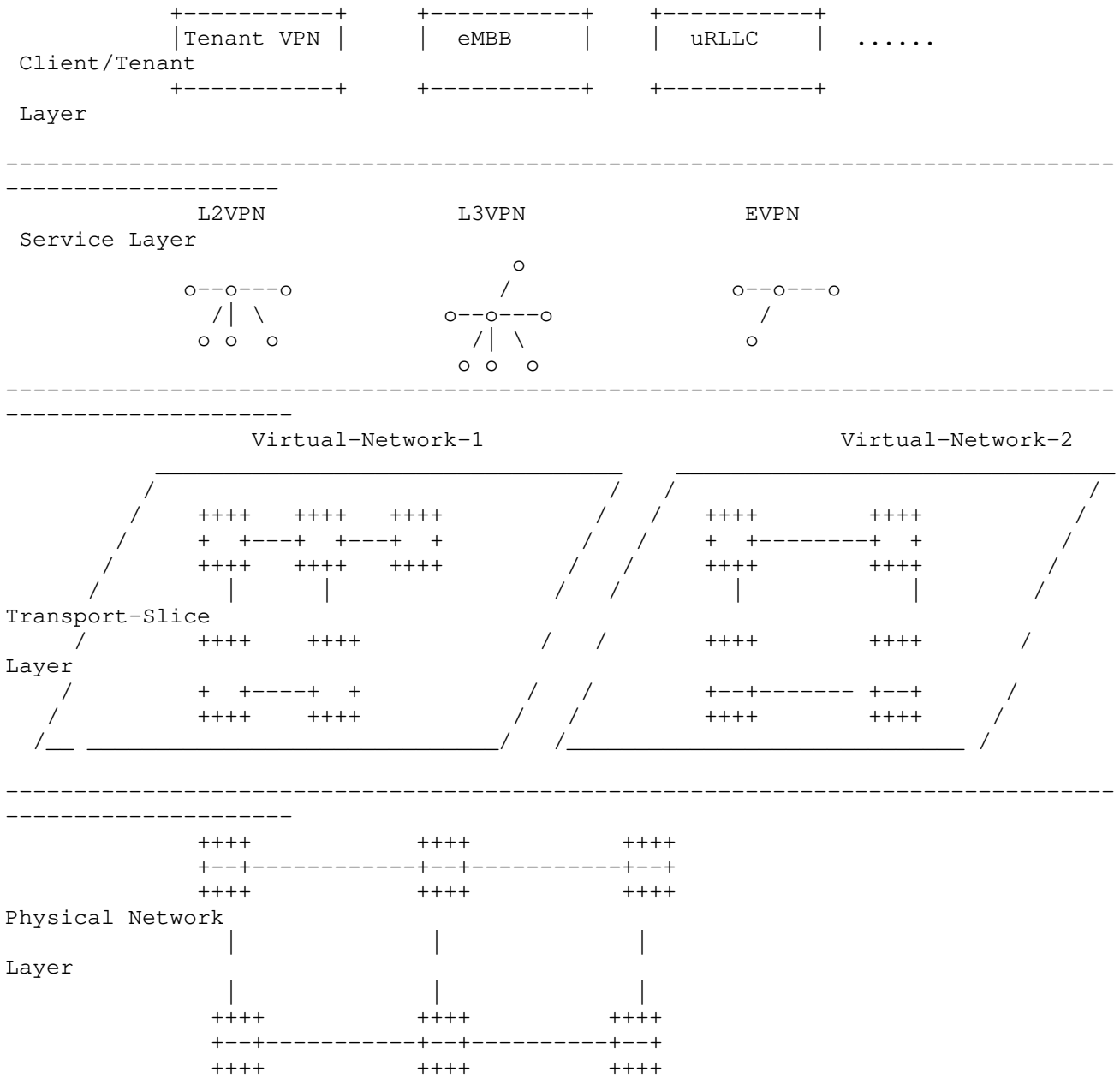


Figure 1 Architecture of TN Slicing

Based on the concept and architecture of Transport slice, the basic requirements and features of Transport slice are as following:

- o On-Demand network reconstitution: The slice network can be reconstituted in network topology and node capability to meet service needs. Each slice network has its own specific bandwidth, latency and lifecycle. Different Transport Slice networks are isolated from each other, and have independent topology and network resources.



- o Decoupling of Service Slice Layer and Physical Network Layer: The Service Slice Layer and the Physical Network Layer are decoupled, and unaware of the details of each other, which simplifies the deployment of services.
- o Similarity of Transport Slice Network and Physical Network for Service Layer: A Transport Slice Network Layer provides network resources to the upper layer (Service Layer) which is the same as the resources provided directly by a physical network from the point view of the upper layer. Services such as VPN service etc. can be deployed directly on the Transport slice network just as they are deployed on the physical network. One Transport slice network can support the deployment of more than one services or VPNs.
- o Data Plane Isolation of Transport Slice Network: The TN provides two types of traffic isolation between different TN slices: hard isolation and soft isolation. Hard isolation is implemented by providing independent circuit switched connections for the exclusive use of one slice, such as MTN (Metro Transport Network, see ITU-T G.mtn), and ODUk. Soft isolation is implemented by using a packet technology (e.g., Ethernet VLAN, MPLS tunnel, and VPN). Services of different slices are isolated from each other.
- o Transport Slice Network: There may be multiple Sub-TN-slices in a Transport Slice Network, and those Sub-Transport slices may be nested. Different sub-TN-slices can be also combined together for an end-to-end TN slice service.

## 2.1. Key Technologies of Transport slice

For the transport network forwarding plane slicing, there are basically two kinds of isolation technology: soft isolation technology and hard isolation technology. The soft isolation is a Layer 2 or Layer 3 technology, such as SR/IP/MPLS based tunnel technology and VPN/VLAN based virtualization technology. The hard isolation is a Layer 1 or optical-layer slicing technology based on physically rigid pipelines, such as MTN, OTN and Wavelength Division Multiplexing (WDM) technologies. In applications, the hybrid hard and soft isolation solution is always used. The hard isolation ensures service isolation, and the soft isolation supports service bandwidth reuse.

So, The Key Technologies of Transport slice should include: Layer-one Data Plane, Layer-Two Data Plane, and Layer-Three Data Plane.



### 3. Slicing Requirements

#### 3.1. Dedicated Virtual Networks

An end-to-end virtual network with dedicated resources is the advantage of network slicing than traditional DiffServ QoS and VPN. For example, DiffServ QoS can distinguish VoIP traffic and other type of traffic (such as high-definition video, web browsing), but can not distinguish the same type of traffic from different tenants, nor isolation of these traffic at all.

Another example is the IoT traffic of health monitoring network which connected hospital and outpatient, it always has strict privacy and safety requirements, including where the data can be stored and who can access the data, all this can not be satisfied by DiffServ QoS as it has not any function of network computing and storage.

Dedicated VN is a distinct object purchased by a customer, and it provides specific function with predictable performance, guaranteed level of isolation and safety. It is not just as QoS.

#### 3.2. End-to-End Slicing

Only an end-to-end slice and fine-grained network can match ultra delay and safety requirements of special service. End-to-end means that it is constructed with AN-slice, TN-slice, and CN-slice part.

Although 3GPP technical specifications mainly focus on the operation and management of AN-slice and CN-slice, which include some NF (network function) components, TN-slice is also created and destroyed according to the related NSI lifecycle. In fact, the 3GPP management system will request expected link requirements related to the network slice (e.g., topology, QoS parameters) with the help of the management system that handles the TN part related to the slice.

For TN part, the link requirements are independent of the existing domain partition of the network, i.e., any intra- or inter-domain link is the candidate resource for the slice. It is also independent of the existing underlay frame or routing technologies (IGP, BGP, Segment Routing, Flex-E, etc.), i.e., any L2 or L3 link is the candidate resource.

#### 3.3. Unified NSI

An NSI is indentified by S-NSSAI (Single Network Slice Selection Assistance Information), which is allocated per PDU session and has semantic global within the AN and CN.

For the purpose of operation and management simplicity, it is also better to have a unified identifier with semantic global to distinguish different TN-slice during the whole TN. TN-slice identifier has a mapping relation with S-NSSAI, perhaps 1:1 or 1:n.

Instead, using different slice identifier across multi-domain of TN for the specific TN-slice will introduce much and unnecessary complexity, especially for case two devices belongs to different domain try to exchange slice-based information directly, without the help of SDN controller to translate the unified TN-slice identifier to an individual domain-wide identifier.

### 3.4. Traffic Engineering

5G system is expected to be able to provide optimized support for a variety of different communication services, different traffic loads, and different end-user communities. For example, the communication services using network slicing may include: vehicle-to-everything (V2X) services, 5G seamless enhanced Mobile BroadBand (eMBB) service with FMC (fixed-mobile convergence), massive IoT connections. Among these service types, high data rates, high traffic densities, low-latency, high-reliability are highlighted requirements.

Traffic engineering mechanism in TN must support the above requirements, bandwidth and delay are two primary TE constraints.

### 3.5. Summarized Requirements

In summary, the following requirements would be satisfied:

REQ1: Provide a distinct virtual network, including dedicated topology, computation, and storage resource, not only traditional QoS;

REQ2: Unified NSI for easy operation and maintenance;

REQ3: E2E network slicing, including both intra-domain and inter-domain case;

REQ4: Customization resource for QoS purpose, bandwidth and delay are basic constraints;

REQ5: Layer 2 as well as Layer 3 link resource partition;

#### 4. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119.

#### 5. Overview of Existing Identifiers

Currently there are multiple existing mature identifiers that could be used to identify the virtual network resource in the transport network, such as:

- o Administrative Group (AG) described in [RFC3630], [RFC5329], [RFC5305] and Extended Administrative Groups (EAGs) described in [RFC7308]
- o Multi-Topology Routing (MTR) described in [RFC5120], [RFC4915], [RFC5340]
- o SR policy color described in [I-D.ietf-spring-segment-routing-policy]
- o FA-id described in [I-D.ietf-lsr-flex-algo]

However, all these identifiers are not sufficient to meet the above requirements of TN-slice. Note that all these identifiers have use case of their own, besides the network slicing use case. Next, we will discuss each of them to determine their matching of slicing requirements.

##### 5.1. AG and EAG Bit

AG and EAG are limited to serve as a link color scheme used in TE path computation to meet the requirements of TE service for a tenant. It is difficult to use them for an NSI allocation mapping (assuming that each bit position of AG/EAG represents an NSI). Hence, they do not meet REQ1. At the same time, AG or EAG cannot be a FIB identifier for best-effort service for the same tenant.

AG and EAG are only as L3 link attribute, not appropriate for L2-bundles member, i.e., not meeting REQ5.

Note that AG and EAG have semantic global, so they meet REQ2,3.

## 5.2. Multi-Topology Identifier

MTR is limited to serve as an IGP logical topology scheme only used in the intra-domain scenario. Thus it is challenging to select inter-area link resources based on MT-ID when E2E inter-domain TE path needs to be created for a tenant. That is, it does not meet REQ3.

Different IGP domain within the same TN-slice may be configured with different MT-ID. Thus MT-ID does not meet REQ2.

MT-ID is only as L3 link attribute, not appropriate for L2-bundles member, so it does not meet REQ5.

## 5.3. SR Policy Color

The color of SR policy defines a TE purpose, which includes a set of constraints such as bandwidth, delay, TE metric, etc. Therefore color is an abstract target, and it is difficult to get a distinct virtual network according to a specific color value. In most cases, only the headend and some other border nodes need to maintain the color template, and a color-based virtual network is hard to present because of too few participants and lack of interaction scheme. That is, the color does not meet REQ1.

We can continue to define TE affinity information in color-template, but that is only appropriate for L3 link, not for L2-bundles member, so the color does not meet REQ5.

Note that the color has global semantic, so it meets REQ3.

## 5.4. Flex-algorithm Identifier

Indeed, FA-id is a short mapping of SR policy color, and it may inherit the matched-degree of the Policy Color. However, FA-id has its own characteristics. A specific FA-id can have more distributed participants and define explicit link resource so that an explicit FA plane can be created. Unfortunately, different best-effort and TE service of the same slice-tenant will define different constraints, resulting in the need to occupy more FA-id resources for one slice-tenant. The relationship between FA-id and slice is not clear. That is, FA-id does not meet REQ1.

On the other hand, FA-id, like MT-ID, is limited to serve as an IGP algorithm scheme used in the intra-domain scenario. It is challenging to select inter-area (especially inter-AS) link resources according to FA-id when the E2E inter-domain TE path needs to be created for the tenant. So, FA-id does not meet REQ3.

Different IGP domain within the same TN-slice may configure different FA-ids, so it does not meet REQ2.

What is more important, than the path in FA plane identified by FA-id is MP2P LSP, so it is hard to define bandwidth reservation for service. So, FA-id does not meet REQ4. Unless each link is totally dedicated to a single FA plane, i.e., link resources are not shared among multiple FA plane.

The link include/exclude rules defined by FA-id is only appropriate for the L3 link, not for L2-bundles member, so FA-id does not meet REQ5.

#### 5.5. New Slice-based Identifier Introduced

Thus, there needs to introduce a new characteristic of NSI that meets the above-listed requirements to isolate underlay resources, and it is a slice-based identifier.

Firstly, it could serve as TE criteria for TE service, this aspect is like AG/EAG; and secondly, as a FIB table identifier for best-effort service, this aspect is like MT-ID or FA-id.

This document introduces a new property of NSI called "Administrative Instance Identifier" (AII) and corresponding method of how to instantiate it in the underlay network to match the above-listed requirements.

#### 6. Overview of AII-based Mechanism

[I-D.ali-spring-network-slicing-building-blocks] described how SR policy [I-D.ietf-spring-segment-routing-policy] can be used to create service slice. This document continues to discuss AII-based mechanism to enhance SR policy to support tenant slice as well as service slice. It will signal the association of AII and shared resources required to create and manage an NSI, and steer the packets to the path within the specific NSI according to SR policy color.

SR policy color has semantic global in order to be conveniently exchanged between two PE routers. They configure the same color template information for the same color value. AII, also with global semantic, can be contained in color template to enhance SR policy to create a TE path within global TN-slice identified by AII. Besides TE service served by explicit SR policy instance, best-effort service is served by AII-specific FIB that is created by default once AII configured.

The following is how AII-based mechanism works.

## 6.1. Physical Network Partition by AII

At the initial stage, each link in a physical network can be colored to conform with network slicing requirements. As previously mentioned, AII can be used to color links to partition underlay resources. Also, we may continue to use AG or EAG to color links for traditional TE within a virtual network specified by an AII. A single or multiple AIIs could be configured on each intra-domain or inter-domain link regardless of IGP instance configuration. At the minimum, a link always belongs to default AII (the value is 0). The number of AIIs configured on a node's links determines the number of virtual networks the node belongs to.

The extension of the existing IGP-TE mechanisms [RFC3630] and [RFC5305] to distribute AII information in an AS as a new TE parameter of a link will be defined in another document.

An SDN controller, using BGP-LS [RFC7752] or another interface, will have a distinct view of each virtual network specified by AII. The extension of BGP-LS will also be defined in another document.

## 6.2. Path within AII specific Slice

Using the CSPF algorithm, a TE path for any best-effort (BE) or traffic-engineered (TE) service can be calculated within a virtual network specified by the AII. The computation criteria could be <AII, min igp-metric> or <AII, traditional TE criteria> for the BE and TE respectively. Combined with segment routing, the TE path could be represented as:

- o a single node-SID of the destination node, for the best-effort service in the domain;
- o node-SIDs of the border node and the destination node, adjacency-SID of inter-domain link, for the inter-domain best-effort service;
- o an explicit adjacency-SID list or compressed with several loose node-SID, for P2P traffic engineered service.

### 6.2.1. SR-BE Path within AII specific Slice

Because packets of the best-effort service could be transported over an MP2P LSP without congestion control, SR best-effort FIB for each virtual network specified by AII to forward best-effort packets may be created in the IGP domain. Thus, CSPF computation with criteria <AII, min igp-metric> is distributed on each node in the IGP domain.

That is similar to the behavior in [I-D.ietf-lsr-flex-algo], but the distributed CSPF computation is triggered by AII.

Besides the best-effort service, SR best-effort FIB entry for specific AII also provide an escape way for traffic engineering service within the same slice when the expected TE purpose can not be meet.

To distinguish forwarding behavior of different virtual networks, prefix-SID need to be allocated per AII and advertised in the IGP domain.

For inter-domain case, in addition to the destination node-SID, several node-SIDs of the domain border node and adjacency-SID of inter-domain link are also needed to construct the E2E segment list. The segment list could be computed with the help of the SDN controller, which needs to take account of AII information during the computation. Even for best-effort service, the head-end has to maintain the corresponding SR-TE tunnel or SR policy.

As same as the prefix-SID, adjacency-SID needs to be allocated per AII to distinguish the forwarding behavior of different virtual networks.

#### 6.2.2. SR-TE Path within AII specific Slice

For P2P traffic engineering service, especially such as the ultra-reliable low-latency communication service, it SHOULD not transfer over an MP2P LSP to avoid the risk of traffic congestion. The segment list could consist of pure adjacency-SID per AII specific. The segment list could be computed by headend or SDN controller. The head-end of the segment list maintains the corresponding SR-TE tunnel or SR policy.

However, label stack depth of the segment list MAY be optimized at a later time based on local policies.

#### 6.3. Traffic Steering to SR policy within Slice

At this moment, we can steer traffic of overlay service to the above SR best-effort FIB, SR-TE tunnel, or SR policy instance for the specific virtual network. The overlay service could specify a color for TE purposes.

For example, color 1000 means <AII=10, min igp-metric> to say "I need best-effort forwarding within AII 10 resource", color 1001 means <AII=10, delay=10ms, AG=0x1> to say "I need traffic engineering

forwarding within AII 10 resource, and only using link with AG equal to 0x1 to reach guarantee of not exceeding 10ms delay time".

Service with color 1000 will be steered to an SR best-effort FIB entry, or an SR-TE tunnel/policy in case of inter-domain.

Service with color 1001 will be steered to an SR-TE tunnel/policy.

#### 6.4. Simple Variant of AII-based Slicing Scheme

There is a simple variant of AII-based slicing scheme for initial slicing requirement of service, where the SDN controller in management partition the whole E2E network topology to multiple strictly isolated VNs identified by AII in local, but let the forwarding equipments be totally unaware of that.

The overlay service is steered to the SR policy whose path is limited within specific VN using a pure adjacency-segment list.

This variant need not introduce any complex virtual network technologies to forwarding equipments, however only for limited scenes.

### 7. Resource Allocation per AII

#### 7.1. L3 Link Resource AII Configuration

In IGP domain, each numbered or unnumbered L3 link could be configured with AII information and synchronized among IGP neighbors. The IGP link-state database will contain L3 links with AII information to support TE path computation taking account of AII criteria. For a numbered L3 link, it could be represented as a tuple <local node-id, remote node-id, local ip-address, remote ip-address>, for unnumbered it could be <local node-id, remote node-id, local interface-id, remote interface-id>. Each L3 link could be configured to belong to a single AII or multiple AII. Note that an L3 link always belongs to default AII(0).

For different <L3 link, AII> tuple it would allocate a different adjacency-SID, as well as advertising with different resource portion such as bandwidth occupied.

Note that AII is independent of IGP instance. An L3 link that is not part of the IGP domain, such as the special purpose for a static route, or an inter-domain link, can also be configured with AII information and allocate adjacency-SID per AII as the same as IGP links. BGP-LS could be used to collect link state data with AII information to the controller, BGP-LS has already provided a



mechanism to collect link state data from many source protocols, such as IGP, Direct, Static configuration, etc., to cover network slicing requirements.

## 7.2. L2 Link Resource AII Configuration

[I-D.ietf-isis-l2bundles] described how to encode adjacency-SID for each L2 member link of an L3 parent link. In the network slicing scenario, it is beneficial to deploy LAG or another virtual aggregation interface between two nodes. If that, the dedicated link resources belong to different virtual networks could be added or removed on demand, they are treated as L2 member links of a single L3 virtual interface. It is the single L3 virtual interface which needs to occupy IP resource and join the IGP instance. Creating a new slice-specific link on demand or removing the old one is likely to affect little configurations.

For network slicing purpose, [I-D.ietf-isis-l2bundles] need to be extended to advertise the AII attribute for each L2 member link. For different <L2 link, AII> tuple it would allocate a different adjacency-SID, as well as advertising with different resource portion such as bandwidth occupied.

In practice, for hard isolation purpose, different L2 member link of the same L3 parent link SUGGESTED to be configured to belong to different AII, with different adjacency-SID. Note that in this case, the L3 parent link belongs to default AII(0), but each L2 member link belongs to the specific non-default AII. An L2 member link maybe a Flex-E channel or ODUK tunnel created/destroyed on demand.

In the control plane, routing protocol packets following the L3 parent link will select the L2 member link with the highest priority.

In the forwarding plane, data packets that belong to the specific virtual network will pass along the L2 member link with the specific AII value.

TE path computation based on link-state database need inspect the detailed L2 members of an L3 adjacency to select the expected L2 link resource.

## 7.3. Node Resource AII Configuration

For topology resource, each node needs to allocate node-SID per AII when it joins the related virtual network. All nodes in the IGP domain can run the CSPF algorithm with criteria <AII, min IGP metric> to compute best-effort next-hop to any other destination nodes for a virtual network AII-specific based on the link-state database that

containing AII information, so that SR best-effort FIB can be constructed for each AII. Static routes could also be added to the AII-specific FIB.

An intra-domain overlay best-effort service belongs to a virtual network could be directly matched in the SR best-effort FIB for the specific AII.

An inter-domain overlay best-effort service belongs to a virtual network could be over a segment list containing domain border node-SID and destination node-SID which could be matched in the SR best-effort FIB for the specific AII.

#### 7.4. Service Function Resource AII Configuration

[I-D.ietf-spring-sr-service-programming] introduces the notion of service segments, and describes how to implement service segments and achieve stateless service programming in SR-MPLS and SRv6 networks. The ability of encoding the service segments along with the topological segment enables service providers to forward packets along a specific network path and through VNFs or physical service appliances available in the network. Typically, a Service Function may be any purposeful execution for the packet, such as DPI, firewall, NAT, etc.

The Service Function is independent of topology, it can also be instantiated per AII, each with different priority to be executed or scheduled. For example, a docker container including specific Service Function process can be generated or destroyed on demand according to the life-cycle of a particular slice. It will have a particular CPU scheduling priority.

At a node, multiple instance of the same type of Service Function for different slice will allocate different Service SID and advertise to other nodes.

#### 8. E2E Slicing with Centralized Mode

[RFC7752] BGP-LS describes the methodology that using BGP protocol to transfer the Link-State information that maybe originated from IGP instance (for intra-domain topology information) or from local direct interface or static configuration(for inter-domain topology information). [I-D.ietf-idr-bgppls-inter-as-topology-ext] also describes a method to firstly put inter-domain interconnections to IGP instance, then always import data from IGP protocol source to BGP-LS. In any case BGP-LS need extend to transfer the Link-State data with AII information.

An E2E inner-AS SR-TE instance with particular color template could be initiated on PE1, PE1 is head-end and PE2 is destination node. BGP-LS could be used to inform the SDN controller about the underlay network topology information including AII attribute. Thus the controller could calculate E2E TE path within the particular virtual network. Especially AII specific Adacency-SID of inter-domain link is included in the E2E SID list.

#### 9. E2E Slicing with Distributed Mode

In some deployments, especially the network evolution from seamless MPLS in reality, operators adopt BGP-LU to build inter-domain MPLS LSP, and overlay service will be directly over BGP-LU LSP.

In this case, the network is divided into some domains and each domain will run its own IGP process. These IGP process are isolated to each other to be simple. That means it is inconvenient to realize network slicing depending on IGP itself with inter-area route leak or redistribution.

For an E2E BGP-LU LSP, if overlay service has TE requirements that defined by a color, the BGP-LU LSP need also have a sense of color, i.e., BGP-LU label could be allocated per color.

At entry node of each domain, BGP-LU LSP generated for specific color will be over intra-domain SR-TE or SR Best-effort path generated for that color again. At exit node of each domain, BGP-LU LSP generated for specific color will select inter-domain forwarding resource per color. Especially, an ASBR will select slice-specific inter-AS link according to AII information of color template.

[RFC7911] defined that multiple paths UPDATE message for the same destination prefix can be advertised in BGP, each UPDATE can contain the Color Extended Community ([I-D.ietf-idr-tunnel-encaps]) with different color value. That is a simple existing way to realize BGP-LU color function, with needless new BGP extensions.

#### 10. Combined with SR Flex-algorithm for Stack Depth Optimization

[I-D.ietf-lsr-flex-algo] introduced a mechanism to do label stack depth optimization for an SR policy in IGP domain part. As the color of SR policy defined a TE purpose, traditionally the headend or SDN controller will compute an expected TE path to meet that purpose.

It is necessary to map a color (32 bits) to an FA-id (8 bits) when SR flex-algorithm enabled for an SR policy. Besides that, it is necessary to enable the FA-id on each node that wants to join the

same FA plane manually. The FAD could copy the TE constraints (not including bandwidth case) contained in the color template.

We need to consider the cost of losing the flexibility of color when executing the flex-algo optimization, and also consider the gap between P2P TE requirements and MP2P SR FA LSP capability, to reach the right balance when deciding which SR policy need optimization.

#### 10.1. Flex-algo Using AII Criteria

Because the first feature of AII is a TE criteria of link and node, it could be served as a parameter of Flex-algo Definition. [I-D.peng-lsr-flex-algo-opt-slicing] described how to extend IGP Flex-algo to compute constraint based paths over the AII specific network slice.

#### 10.2. Best-effort Color Template Mapping to Flex-algo

As described above, for best-effort service we have already constructed SR best-effort FIB per AII, that is mostly like Flex-algo. Thus, it is not necessary to map to FA-id again for a color template which has defined a best-effort behavior within the dedicated AII. Of course, if someone forced to remap it, there is no downside for the operation, the overlay best-effort service (with a color which defined specific AII, best-effort requirement, and mapping FA-id) in IGP domain will try to recurse over <AII, prefix> or <FA-id, prefix> FIB entry.

#### 10.3. Traffic Engineering Color Template Mapping to Flex-algo

An SR-TE tunnel/policy that served for traffic engineering service of a virtual network specified by an AII was generated and computed according to the relevant color template, which contained specific AII and some other traditional TE constraints. If we config mapping FA-id under the color template, the SR-TE tunnel/policy instance could inherit forwarding information from corresponding SR Flex-Algo FIB entry.

### 11. Network Slicing Examples

In this section, we will further illustrate the point through some examples. All examples share the same figure below.

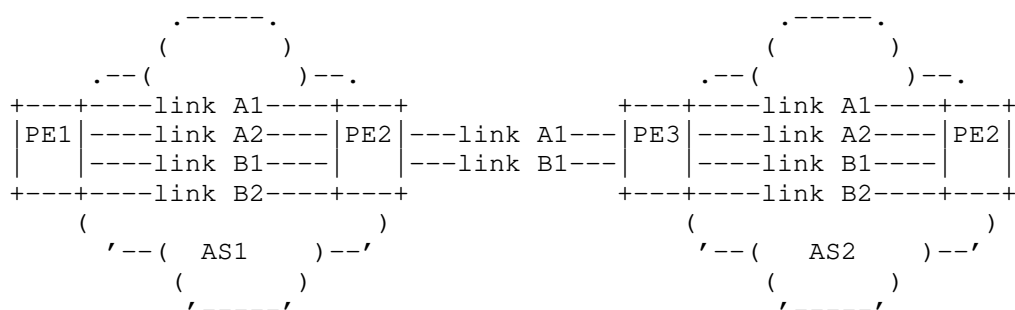


Figure 2 Network Slicing via AII

Suppose that each link belongs to separate virtual network, e.g., link Ax belongs to the virtual network colored by AII A, link Bx belongs to the virtual network colored by AII B. link x1 has an IGP metric smaller than link x2, but TE metric larger.

To simplify the use case, each AS just contained a single IGP area.

11.1. Intra-domain Network Slicing Example

11.1.1. Best-effort Service over Network Slice Example

From the perspective of node PE1 in AS1, it will calculate best-effort forwarding entry for each AII instance (including default AII) to destinations in the same IGP area. For example:

For <AII=0, destination=ASBR1> entry, forwarding information could be ECMP during link A1 and link B1, with destination node-SID 100 for <AII=0, destination=ASBR1>.

For <AII=A, destination=ASBR1> entry, forwarding information could be link A1, with destination node-SID 200 for <AII=A, destination=ASBR1>.

For <AII=B, destination=ASBR1> entry, forwarding information could be link B1, with destination node-SID 300 for <AII=B, destination=ASBR1>.

11.1.2. TE Service over Network Slice Example

It could also initiate an SR-TE instance (SR tunnel or SR policy) with the particular color template on PE1, PE1 is headend and ASBR1 is destination node. For example:

For SR-TE instance 1 with color template which defined criteria including {default AII, min TE metric}, forwarding information could be ECMP during two segment list {adjacency-SID 1002 for <AII=0, link A2> @PE1} and {adjacency-SID 1004 for <AII=0, link B2> @PE1}.

For SR-TE instance 2 with the color template which defined criteria including {AII=A, min TE metric}, forwarding information could be presented as the segment list {adjacency-SID 2002 for <AII=A, link A2> @PE1}.

For SR-TE instance 3 with the color template which defined criteria including {AII=B, min TE metric}, forwarding information could be presented as the segment list {adjacency-SID 3004 for <AII=B, link B2> @PE1}.

### 11.1.3. TE Service over Network Slice with Flex-algo Example

Furthermore, we can use SR Flex-algo to optimize the above SR-TE instance. For example, for SR-TE instance 1, we can define FA-ID 201 with FAD that contains the same information as the color template, in turn, FA-ID 202 for SR-TE instance 2, FA-ID 203 for SR-TE instance 3. Note that each FA-ID also needs to be enabled on ASBR1. So that the corresponding SR FA entry could be:

For <FA-ID=201, destination=ASBR1> entry, forwarding information could be ECMP during link A2 and link B2, with destination node-SID 600 for <FA-ID=201, destination=ASBR1>.

For <FA-ID=202, destination=ASBR1> entry, forwarding information could be link A2, with destination node-SID 700 for <FA-ID=202, destination=ASBR1>.

For <FA-ID=203, destination=ASBR1> entry, forwarding information could be link B2, with destination node-SID 800 for <FA-ID=203, destination=ASBR1>.

## 11.2. Inter-domain Network Slicing via BGP-LS Example

### 11.2.1. Best-effort Service Example

For SR-TE instance 4 with color template which defined criteria including {default AII, min IGP metric}, forwarding information could be segment list {node-SID 100 for <AII=0, destination=ASBR1> , adjacency-SID 1001 for <AII=0, link A1> @ASBR1, node-SID 400 for <AII=0, destination=PE2> }.

For SR-TE instance 5 with color template which defined criteria including {AII=A, min IGP metric}, forwarding information could be

```
segment list {node-SID 200 for <AII=A, destination=ASBR1> ,
adjacency-SID 1001 for <AII=A, link A1> @ASBR1, node-SID 500 for
<AII=A, destination=PE2> }.
```

For SR-TE instance 6 with color template which defined criteria including {AII=B, min IGP metric}, forwarding information could be segment list {node-SID 300 for <AII=B, destination=ASBR1> , adjacency-SID 1003 for <AII=B, link B1> @ASBR1, node-SID 600 for <AII=B, destination=PE2> }.

#### 11.2.2. TE Service Example

For SR-TE instance 7 with color template which defined criteria including {default AII, min TE metric}, forwarding information could be ECMP during two segment list {adjacency-SID 1002 for <AII=0, link A2> @PE1, adjacency-SID 1001 for <AII=0, link A1> @ASBR1, adjacency-SID 1002 for <AII=0, link A2> @ASBR2} and {adjacency-SID 1004 for <AII=0, link B2> @PE1, adjacency-SID 1003 for <AII=0, link B1> @ASBR1, adjacency-SID 1004 for <AII=0, link B2> @ASBR2}.

For SR-TE instance 8 with color template which defined criteria including {AII=A, min TE metric}, forwarding information could be segment list {adjacency-SID 2002 for <AII=A, link A2> @PE1, adjacency-SID 2001 for <AII=A, link A1> @ASBR1, adjacency-SID 2002 for <AII=A, link A2> @ASBR2}.

For SR-TE instance 9 with color template which defined criteria including {AII=B, min TE metric}, forwarding information could be segment list {adjacency-SID 3004 for <AII=B, link B2> @PE1, adjacency-SID 3003 for <AII=B, link B1> @ASBR1, adjacency-SID 3004 for <AII=B, link B2> @ASBR2}.

#### 11.2.3. TE Service Using Flex-algo Example

For TE service, if we use SR Flex-algo to do optimization, the above forwarding information of each TE instance could inherit the corresponding SR FA entry, it would look like this:

```
For SR-TE instance 7, forwarding information could be ECMP during two
segment list {node-SID 600 for <FA-ID=201, destination=ASBR1> ,
adjacency-SID 1001 for <AII=0, link A1> @ASBR1, node-SID 600 for <FA-
ID=201, destination=PE2> } and {adjacency-SID 1004 for <AII=0, link
B2> @PE1, adjacency-SID 1003 for <AII=0, link B1> @ASBR1, adjacency-
SID 1004 for <AII=0, link B2> @ASBR2}.
```

For SR-TE instance 8 with color template which defined criteria including {AII=A, min TE metric}, forwarding information could be segment list {node-SID 700 for <FA-ID=202, destination=ASBR1> ,

adjacency-SID 2001 for <AII=A, link A1> @ASBR1, node-SID 700 for <FA-ID=202, destination=PE2> }.

For SR-TE instance 9 with color template which defined criteria including {AII=B, min TE metric}, forwarding information could be segment list {node-SID 800 for <FA-ID=203, destination=ASBR1> , adjacency-SID 3003 for <AII=B, link B1> @ASBR1, node-SID 800 for <FA-ID=203, destination=PE2> }.

### 11.3. Inter-domain Network Slicing via BGP-LU Example

In figure 1, PE2 can allocate and advertise six labels for its loopback plus color 1, 2, 3, 4, 5, 6 respectively. Suppose color 1 defines {default AII, min IGP metric}, color 2 defines {AII=A, min IGP metric}, color 3 defines {AII=B, min IGP metric}, and color 4 defines {default AII, min TE metric}, color 5 defines {AII=A, min TE metric}, color 6 defines {AII=B, min TE metric}. PE2 will advertise these labels to ASBR2 and ASBR2 then continues to allocate six labels each for prefix PE2 plus different color. Other nodes will have the same operation. Ultimately PE1 will maintain six BGP-LU LSP.

For example, the BGP-LU LSP for color 1 will be over SR best-effort FIB entry node-SID 100 for <AII=0, destination=ASBR1> to pass through AS1, over adjacency-SID 1001 for <AII=0, link A1>@ASBR1 to pass inter-AS, over SR best-effort FIB entry node-SID 400 for <AII=0, destination=PE2> to pass through AS2.

For example, The BGP-LU LSP for color 4 will over SR-TE instance 1 (see section 10.1.2), or SR best-effort FIB entry node-SID 600 for <FA-id=201, destination=ASBR1> (see section 10.1.3) to pass through AS1, over adjacency-SID 1001 for <AII=0, link A1>@ASBR1 to pass inter-AS, over SR-TE instance 1' or corresponding SR FA entry to pass through AS2. Note that ASBR1 need also understand the meaning of a specific color and select forwarding resource between two AS.

## 12. Implementation Suggestions

The implementation cost is low by means of existing segment routing infrastructure.

### 12.1. SR-MPLS

As a node often contains control plane and forwarding plane, a suggestion is that only default AII specific FTN table, i.e, traditional FTN table, need be installed on forwarding plane, so that there are not any modification and upgrade requirement for hardware and existing MPLS forwarding mechanism. FTN entry for non-default AII instance will only be maintained on the control plane and be used



for overlay service iteration according to next-hop plus color (color will give AII information and mapping FA-id information). Note that ILM entry for all AII need be installed on forwarding plane, that does not bring any confusion because of prefix-SID allocation per AII.

SR NHLFE entry and other iteration entry such as <next-hop, color> can contain AII information for expected packet scheduling. The Slice Type value of AII can distinguish flows by coarse-grained classification, while the Instance value of AII can be used for more scheduling policy.

## 12.2. SRv6

For SRv6 case, IPv6 address resource is directly used to represent SID, so that different IPv6 block could be allocated to different slice. There are two possible ways to advertise slice specific IPv6 block:

- o Traditional prefix reachability, but only for default AII (0) specific IPv6 block.
- o New SRv6 Locator advertisement, for nonzero AII specific IPv6 block.

Forwarding entries for the default AII specific locators advertised in prefix reachability MUST be installed in the forwarding plane of receiving routers.

Forwarding entries for the nonzero AII specific locators advertised in the SRv6 Locator MUST be also installed in the forwarding plane of receiving SRv6 capable routers when the associated AII is supported by the receiving node.

The entries of both the above two cases SHOULD be installed in the unified FIB table, i.e., a single FIB table for default AII, because different IPv6 block is allocated to different slice. Instead, more FIB tables created for each VN in dataplane will bring complexity for overlay service iteration, that is why MTR has no practical deployment.

The forwarding information of FIB entry can contain AII information for expected packet scheduling.

## 13. IANA Considerations

This document requests IANA to create a new top-level registry called "Network Slicing Parameters". This registry is being defined to serve as a top-level registry for keeping all other Network Slicing sub-registries.

Additionally, a new sub-registry "AII (TN-slice Identifier) codepoint" is to be created under top-level "Network Slicing Parameters" registry. This sub-registry maintains 32-bit identifiers and has the following registrations:

Slice Type (High 8bits)	Instance (Low 24bits)	Description
0 (Normal)	0	Default Slice: the original physical network.
	nonzero	Normal Slice, for user defined.
1 (eMBB)	0	Resevered.
	nonzero	Slice suitable for the handling of 5G enhanced Mobile Broadband, for user defined.
2 (URLLC)	0	Resevered.
	nonzero	Slice suitable for the handling of ultra- reliable low latency communications, for user defined.
3 (MIoT)	0	Resevered.
	nonzero	Slice suitable for the handling of massive IoT, for user defined.
4 (V2X)	0	Resevered.
	nonzero	Slice suitable for the handling of V2X services, for user defined.
5-255	any	Unassigned.

Table 1. AII Codepoint

## 14. Security Considerations

TBD.

## 15. Acknowledgements

TBD.

## 16. Normative references

[I-D.ali-spring-network-slicing-building-blocks]

Ali, Z., Filsfils, C., Camarillo, P., and D. Voyer, "Building blocks for Slicing in Segment Routing Network", draft-ali-spring-network-slicing-building-blocks-02 (work in progress), November 2019.

[I-D.ietf-idr-bgppls-inter-as-topology-ext]

Wang, A., Chen, H., Talaulikar, K., Zhuang, S., and S. Ma, "BGP-LS Extension for Inter-AS Topology Retrieval", draft-ietf-idr-bgppls-inter-as-topology-ext-07 (work in progress), September 2019.

[I-D.ietf-idr-tunnel-encaps]

Patel, K., Velde, G., and S. Ramachandra, "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-15 (work in progress), December 2019.

[I-D.ietf-isis-l2bundles]

Ginsberg, L., Bashandy, A., Filsfils, C., Nanduri, M., and E. Aries, "Advertising L2 Bundle Member Link Attributes in IS-IS", draft-ietf-isis-l2bundles-07 (work in progress), May 2017.

[I-D.ietf-lsr-flex-algo]

Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-05 (work in progress), November 2019.

[I-D.ietf-spring-segment-routing-policy]

Filsfils, C., Sivabalan, S., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-06 (work in progress), December 2019.

- [I-D.ietf-spring-sr-service-programming]  
Clad, F., Xu, X., Filsfils, C., daniel.bernier@bell.ca,  
d., Li, C., Decraene, B., Ma, S., Yadlapalli, C.,  
Henderickx, W., and S. Salsano, "Service Programming with  
Segment Routing", draft-ietf-spring-sr-service-  
programming-01 (work in progress), November 2019.
- [I-D.nsd-treas-transport-slice-definition]  
Rokui, R., Homma, S., and K. Makhijani, "IETF Definition  
of Transport Slice", draft-nsdt-teas-transport-slice-  
definition-00 (work in progress), November 2019.
- [I-D.peng-lsr-flex-algo-opt-slicing]  
Peng, S., Chen, R., and G. Mirsky, "IGP Flexible Algorithm  
Optimization for Network Slicing", draft-peng-lsr-flex-  
algo-opt-slicing-00 (work in progress), November 2019.
- [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>>.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering  
(TE) Extensions to OSPF Version 2", RFC 3630,  
DOI 10.17487/RFC3630, September 2003,  
<<https://www.rfc-editor.org/info/rfc3630>>.
- [RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P.  
Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF",  
RFC 4915, DOI 10.17487/RFC4915, June 2007,  
<<https://www.rfc-editor.org/info/rfc4915>>.
- [RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi  
Topology (MT) Routing in Intermediate System to  
Intermediate Systems (IS-IS)", RFC 5120,  
DOI 10.17487/RFC5120, February 2008,  
<<https://www.rfc-editor.org/info/rfc5120>>.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic  
Engineering", RFC 5305, DOI 10.17487/RFC5305, October  
2008, <<https://www.rfc-editor.org/info/rfc5305>>.
- [RFC5329] Ishiguro, K., Manral, V., Davey, A., and A. Lindem, Ed.,  
"Traffic Engineering Extensions to OSPF Version 3",  
RFC 5329, DOI 10.17487/RFC5329, September 2008,  
<<https://www.rfc-editor.org/info/rfc5329>>.

- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <<https://www.rfc-editor.org/info/rfc5340>>.
- [RFC7308] Osborne, E., "Extended Administrative Groups in MPLS Traffic Engineering (MPLS-TE)", RFC 7308, DOI 10.17487/RFC7308, July 2014, <<https://www.rfc-editor.org/info/rfc7308>>.
- [RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <<https://www.rfc-editor.org/info/rfc7752>>.
- [RFC7911] Walton, D., Retana, A., Chen, E., and J. Scudder, "Advertisement of Multiple Paths in BGP", RFC 7911, DOI 10.17487/RFC7911, July 2016, <<https://www.rfc-editor.org/info/rfc7911>>.

## Authors' Addresses

Shaofu Peng  
ZTE Corporation

Email: peng.shaofu@zte.com.cn

Ran Chen  
ZTE Corporation

Email: chen.ran@zte.com.cn

Gregory Mirsky  
ZTE Corporation

Email: gregimirsky@gmail.com

Fengwei Qin  
China Mobile

Email: qinfengwei@chinamobile.com

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 . . . . .	2
2. Conventions used in this document . . . . .	3
2.1. Tree Diagrams . . . . .	4
3. Transport Slice NBI Model Usage . . . . .	4
4. Transport Slice NBI Model Overview . . . . .	5
5. Transport Slice NBI Model Description . . . . .	8
5.1. Transport Slice Connection Pattern . . . . .	8
5.2. Transport Slice EndPoint (TSE) . . . . .	8
5.3. Transport Slice SLO . . . . .	9
6. Transport Slice Monitoring . . . . .	10
7. Transport Slice NBI Model Usage Example . . . . .	11
8. Transport Slice NBI Module . . . . .	11
9. Security Considerations . . . . .	26
10. IANA Considerations . . . . .	27
11. Acknowledgments . . . . .	27
12. References . . . . .	27
12.1. Normative References . . . . .	27
12.2. Informative References . . . . .	29
Appendix A. Comparison with Other Possible Design choices for Transport Slice NBI (Northbound Interface) . . . . .	30
A.1. ACTN VN Model Augmentation . . . . .	30
A.2. RFC8345 Augmentation Model . . . . .	31
Appendix B. Appendix B Transport Slice Filter Criteria . . . . .	31
Authors' Addresses . . . . .	32

## 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:

- o client
- o configuration data
- o state data

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

- o augment
- o data model
- o data node

This document also makes use of the following terminology introduced in the Transport Slice definition draft [I-D.nsd-t-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.nsd-t-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.nsd-t-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.

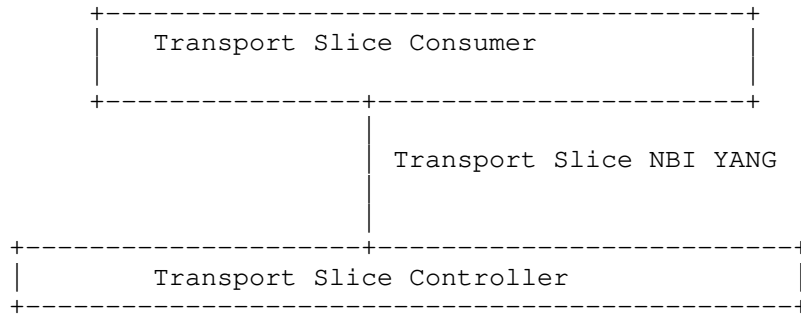
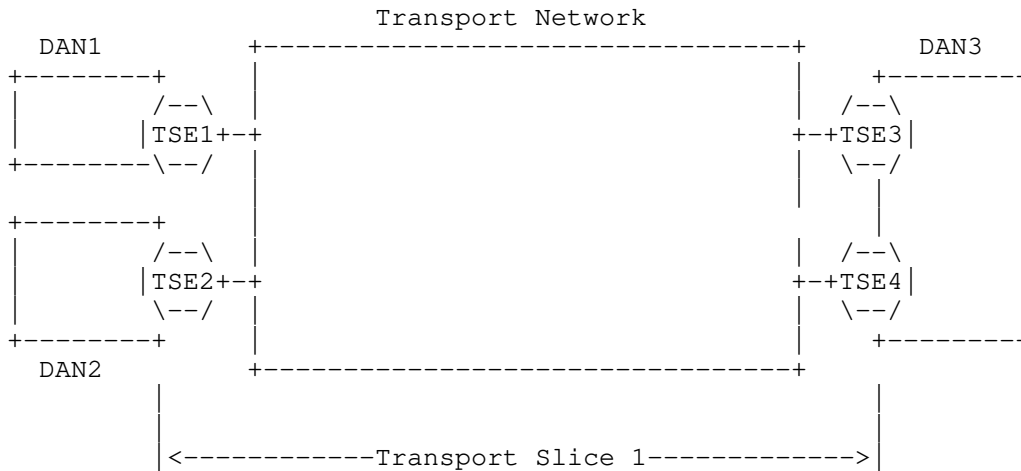


Figure 1 Transport Slice NBI Model Context

4. Transport Slice NBI Model Overview

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



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.nsd-t-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:

- o 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.

- o 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' Transport 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-group"
- 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.nsdtd-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.nsdtd-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 Interface (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:

- o VN: The VN can be seen as a set of edge-to-edge abstract links (a Type 1 VN).
- o AP"links" list and "termination points" list describe how nodes in a network are connected to each other
- o VN-AP:vertical layering relationships between transport slice networks and underlay networks
- o 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:

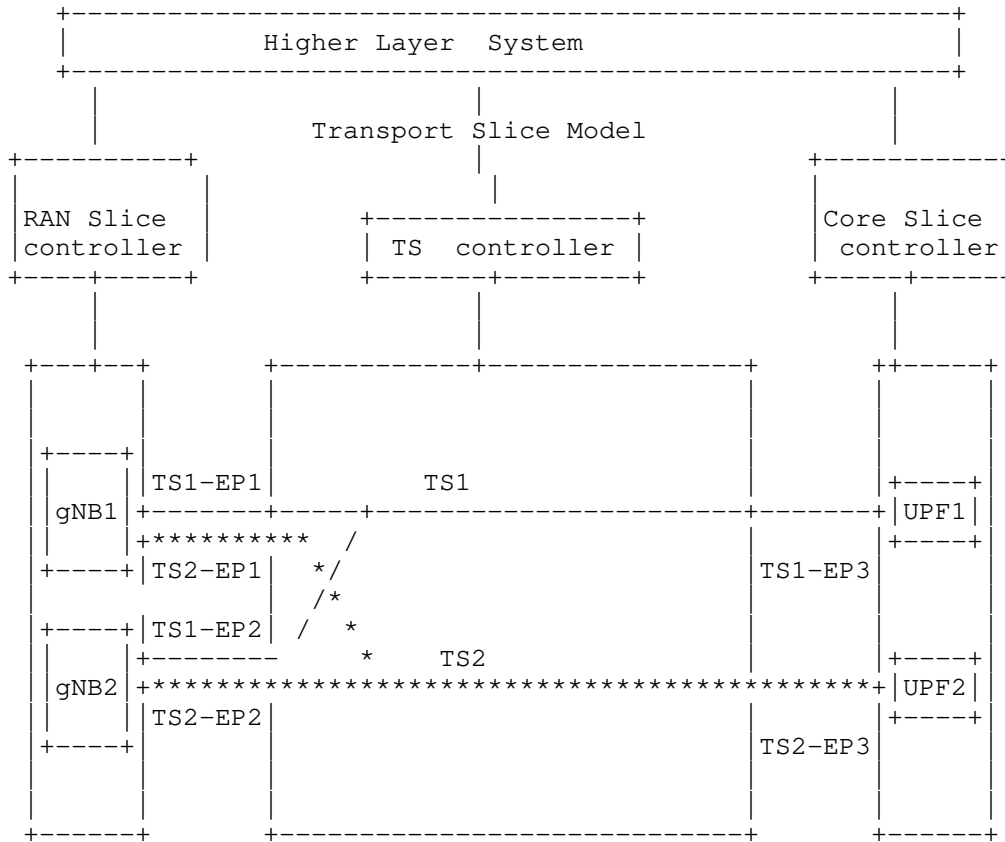
- o Transport network: a transport network list and an list of nodes contained in the transport network
- o Link: "links" list and "termination points" list describe how nodes in a network are connected to each other
- o Support network: vertical layering relationships between transport slice networks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



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

TEAS  
Internet-Draft  
Intended status: Standards Track  
Expires: January 14, 2021

Z. Zhang  
S. Hegde  
Juniper Networks  
A. Gulko  
Refinitiv  
July 13, 2020

Network Slicing with Flexible Traffic Engineering  
draft-zzhang-teas-network-slicing-with-flex-te-00

Abstract

This document specifies procedures and signaling enhancements to Flexible Algorithm to ease provisioning and to scale it better via Flexible Traffic Engineering, which is an integration of Flexible Algorithm and Segment Routing [RFC8402] Traffic Engineering (SR-TE).

Requirements Language

The key words "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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

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 14, 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 . . . . .	2
1.1.	FlexAlgo Background . . . . .	3
1.2.	Central Provisioning and Signaling of FlexAlgo . . . . .	4
1.3.	Flexible Traffic Engineering (FlexTE) and Targeted Signaling . . . . .	4
1.4.	Traffic Isolation and FlexAlgo/FlexTE . . . . .	5
2.	Specification . . . . .	6
2.1.	Southbound BGP-LS Encoding of FAD . . . . .	6
2.2.	Southbound BGP-LS Encoding of Link Administrative Group Information . . . . .	7
2.3.	OSPF/ISIS Encoding of Link Administrative Group Information for Cetralized Advertisement . . . . .	7
2.4.	FlexAlgo and Link AG Signaling from Controllers . . . . .	8
3.	Security Considerations . . . . .	8
4.	IANA Considerations . . . . .	8
5.	Acknowledgements . . . . .	8
6.	References . . . . .	9
6.1.	Normative References . . . . .	9
6.2.	Informative References . . . . .	9
	Authors' Addresses . . . . .	10

## 1. Introduction

[dong-network-slicing-problem-statement] defines Network Slicing Problem Statement for IP/MPLS networks. While Virtual Private Networks (VPNs) have been widely deployed to provide many different virtual networks on the same physical operator network, and can be reused to provide network slicing service to applications, currently those VPNs share the same underlay operator network without any separation and isolation.

Multi-Topology Routing (MTR) [RFC4915] [RFC5120] provides a mechanism to have a set of independent IP topologies referred to as Multi-Topologies (MTs) over the same underlay network. It can be used to provide separation and isolation required by network slicing, though MTR has not been widely deployed over the years, except limited usage



of maintaining separate IGP routing domains for isolated multicast islands within the backbone.

Some reasons for MTR's lack of success are listed below:

1. Lack of strong demand for mapping traffic to different MTs
2. Lack of good mechanism for mapping traffic to different MTs
3. Lack of operating tools to ease provisioning and monitoring

In 5G era, 1) is no longer the case and 3) needs to be addressed given the network slicing requirements. 2) is addressed by SR and Flexible Algorithm (FlexAlgo) [ietf-lsr-flex-algo]. This document specifies signaling enhancements to FlexAlgo to ease provisioning and to scale it better via Flexible Traffic Engineering (FlexTE), which is an integration of FlexAlgo and Segment Routing [RFC8402] Traffic Engineering (SR-TE).

#### 1.1. FlexAlgo Background

FlexAlgo can be viewed as a more flexible and light-weighted mechanism of MTR. A Flexible Algorithm is defined as a <Calculation Type, Metric Type, Included/excluded Administrative Group> tuple. The "Included/excluded Administrative Group" defines the (sub-)topology used for the algorithm. While there are different metric types to be used, there are no per-algorithm metric values advertised for links.

Routers are configured to use certain algorithms for its SPF calculations. Definitions for the algorithms are locally configured and/or learnt through signaling.

The Administrative Groups (AG) of a link, which are often referred to as link colors, are advertised in a 32-bit AG BitMask as specified in [RFC3630] [RFC5305] or an arbitrary length EAG BitMask as specified in [RFC7308]. The advertisements are originated from the router owning the link based on local provisioning.

While not a mandatory part of FlexAlgo, Segment Routing can be integrated with FlexAlgo seamlessly to map traffic to different algorithms: prefix SIDs can optionally be associated with algorithms, so that a prefix can be reached via different SIDs or SID lists, following different paths.

## 1.2. Central Provisioning and Signaling of FlexAlgo

More and more operators use controllers for centralized orchestration, provisioning and signaling. In fact, even before the controllers were used, the network planning was centralized, albeit done manually, and then configuration information resulting from centralized planning was entered into individual routers via out of band means.

The centralized model can be applied to FlexAlgo very well - instead of provisioning FAD and link AGs on individual routers after centralized planning, the provisioning is be done centrally on the controllers and then constraints and link AG information are signaled to routers.

Given that it is very common for controllers to learn network information via northbound BGP-LS [RFC7752], this document uses southbound BGP-LS to distribute FAD and link AG information from controllers to BGP-LS speakers. This can also take advantages of inherent BGP mechanisms for optimized large scale state distribution. If not all routers but only IGP border routers run BGP-LS, the border routers will then flood received information via IGP.

## 1.3. Flexible Traffic Engineering (FlexTE) and Targeted Signaling

With current FlexAlgo mechanisms, Flexible Algorithm Definitions (FADs) and link AG information are flooded throughout an IGP area and every router will do an SPF calculation for each algorithm. This may work for a few algorithms but it will not scale for larger number of alorithms that are necessary for large number of slices in some 5G scenarios.

To address the scaling problem, the above flooded information and SPF caculation may be restricted to network edge only. The idea is first introduced in [I-D.drake-bess-enhanced-vpn] but adapted to FlexAlgo in this document.

With this scheme, the internal routers don't have per-algorithm information and do not do per-algorithm based SPF or per-algorithm prefix-SID based forwarding. The edge routers use SR-TE Adjacency SID-lists to explicitly steer traffic through the network. This is referred to as Flexible Traffic Engineering (FlexTE), an integraion of Flexible Algorithm and SR Traffic Engineering.

Specifically, with BGP Route Target [RFC4364] and Route Target Constrains [RFC4684] mechanisms, the FADs and link AG information are only propagated to and imported by edge routers that need that information. For example, if a network slice is presented to

application as a VPN and instantiated in the underlay with a Flexible Algorithm that utilizes only "red" links, then that specific FAD and "red" link AG information are advertised to and imported by only PEs for that VPN (if the same algorithm is used by many VPNs then all PEs of those VPNs will import the relevant information).

For better scalability, the link AG information is encoded in a new type of Route Target (RT) used for the control of route propagation and importation, as detailed in Section 2.2 and [I-D.zzhang-idr-bitmask-route-target].

Because a router may not be able to push too deep a label stack, per-algorithm Binding SIDs may have to be used. For example, if there are 10 hops from PE1 to PE2 while the maximum labels that PE1 can push is 5, then PE1 has to use a label stack that specifies the explicit hop-by-hop path (calculated by an algorithm) to an intermediate router P1 and a binding SID advertised by P1 for PE2. For P1 to calculate the per-algorithm explicit path to PE2, it also needs to know the information for that algorithm, and it can do so following the same way as how PE1 learns the information.

#### 1.4. Traffic Isolation and FlexAlgo/FlexTE

FlexAlgo as described in [I-D.ietf-lsr-flex-algo] separates the routing domain into different planes. The primary and backup paths are computed based on the topology that corresponds to the plane. FlexAlgo provides strict isolation of the data traffic between the different planes. Notice that FlexAlgo is suitable for slices that need complete isolation. Packet transport networks are expected to have a limited number of such isolated routing planes.

With FlexTE, traffic isolation is achieved via SR-TE Adjacency SID lists, but during local Fast ReRoute (FRR) traffic may flow through paths that don't satisfy constraints. If the SR-TE SID list is too long, Node SIDs may be used but the traffic isolation is not possible on the path between node SIDs, unless some internal routers also get targeted signaling, behave as edge routers, and advertise per-algorithm Node/Binding SIDs (targeted at the edge routers and those selected internal routers). Therefore, FlexTE is more suitable for soft slicing where traffic isolation is not critical in certain situations.

With 5G, network slicing requires high number of slices though they may not necessarily require routing plane isolation but they may need to satisfy certain constraints and have guaranteed Quality Of Service, and FlexTE as a flexible soft slicing solution allows for slice creation inside specific isolated planes or in a generic plane.

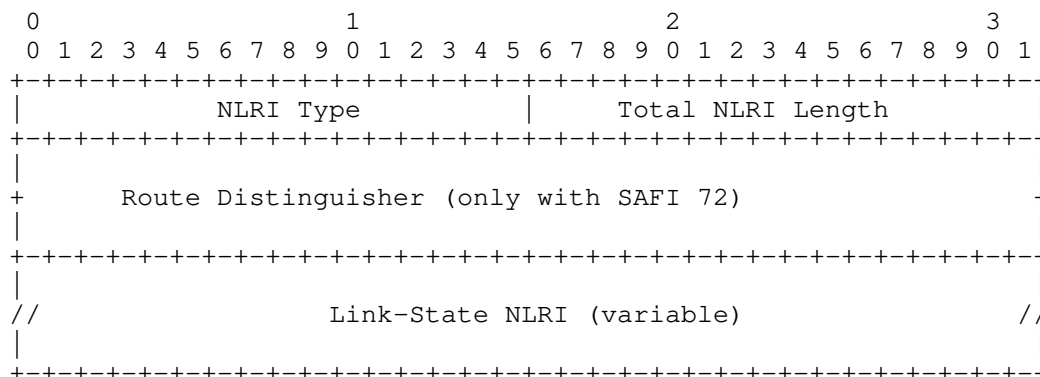
The QOS guarantees for the slices are outside the scope of this document.

2. Specification

BGP-LS [RFC7752] is for "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP". This document extends it for south-bound distribution of FlexAlgo/TE constraint related information, and specifies relevant procedures for FlexAlgo based on centralized, targeted signaling.

2.1. Southbound BGP-LS Encoding of FAD

Currently BGP-LS uses the following NLRI format with AFI 16388 and SAFI 71/72:



A new NLRI type TBD1 is added to advertise FAD. For simplicity, the variable Link-State NLRI field has the exactly same TLV format of ISIS FAD Sub-TLV as specified in [I-D.ietf-lsr-flex-algo], including the Type number and Sub-TLVs.

Type	NLRI Type
1	Node NLRI
2	Link NLRI
3	IPv4 Topology Prefix NLRI
4	IPv6 Topology Prefix NLRI
TBD1	Flexible Algorithm Definition

## 2.2. Southbound BGP-LS Encoding of Link Administrative Group Information

Currently, BGP-LS encodes link Administrative Group information as a Type 1088 Administrative Group TLV in BGP-LS attribute attached to Link NLRIs that are propagated northbound from routers to controllers. For the southbound signaling of Administrative Group information from controllers, for the purpose of targeted propagation and importation, the Administrative Group information are encoded in a new Bitmask Route Target as specified in [I-D.zzhang-idr-bitmask-route-target]. The Administrative Group TLV is omitted from the BGP-LS Attribute for the link because the information is already encoded in the BitMask RT.

Specifically, the EAG BitMask is encoded into the Bitmask field of a Bitmask RT that is attached to the Link NLRI for the link. The Global Administrator (GA) Type, GA Length, and Local Administrator fields are set according to the operator's need to provide a context.

To distinguish from Link NLRIs signaled northbound by routers, the Protocol-ID of the Link NLRI is set to BGP (to be assigned by IANA).

## 2.3. OSPF/ISIS Encoding of Link Administrative Group Information for Centralized Advertisement

When centralized provisioning and signaling is not used, an OSPF router advertises its local links' attributes in OSPFv2 Extended Link Opaque LSAs. The LSA includes OSPFv2 Extended Link TLVs, one for each link, which in turn includes sub-TLVs for specific link attributes.

The same OSPFv2 Extended Link TLVs can be used for ABRs to flood link attributes that are centrally provisioned on and signaled from controllers, but they MUST additionally carry a new sub-TLV to indicate the routers that host the links, because these Extended Link TLVs are in the Extended Link Opaque LSAs originated by the ABRs not those originated by the routers hosting the links. The sub-TLV is called Hosting Router sub-TLV, with a new TBD2 type and a 4-octet value for the Router ID of the router hosting the link.

For OSPFv3, a router advertises its local links' TE attributes in Intra-Area-TE LSAs, which contains Link TLVs with link attribute sub-TLVs. Similarly to OSPFv2, when ABRs flood the link attributes that are centrally provisioned on and signaled from controllers, the Link TLVs MUST carry the Hosting Router sub-TLV.

For ISIS, the Link Administrative Group information is signaled as sub-TLVs in Extended IS Reachability TLV [RFC5305]. Similarly, when

ABRs flood the link attributes that are centrally provisioned on and signaled from controllers, the Extended IS Reachability TLV MUST carry a new Hosting System sub-TLV. The sub-TLV has a new type TBD3 and a 7-octet value for system ID and pseudonode number.

When a router receives a OSPFv2/OSPFv3 Link TLV with Hosting Router sub-TLV or an ISIS Extended IS Reachability TLV with Hosting System sub-TLV, it MUST associate the link with the advertised hosting router/system, not with the originator of the OSPF LSA or ISIS LSP.

#### 2.4. FlexAlgo and Link AG Signaling from Controllers

With centralized provisioning and signaling, a controller signals Link AG information using BGP-LS Link NLRI with a BitMask RT attached, as specified in Section 2.2.

The controller signals FADs used in the domain using the BGP-LS NLRI type TBD1. The updates carry a Bitmask RT with the bits set for the AGs that the FADs care about.

Routers that need to learn the information MUST have a BitMask RT locally configured, with the bits set for the AGs that they care about, so that they will import corresponding NLRIs. In case of FlexTE, only edge routers and some internal routers will have the BitMask RT locally configured. Otherwise, all BGP-LS routers are configured with a BitMask RT to import all FAD and Link NLRIs.

To optimize the propagation of south-bound BGP-LS NLRIs, Route Target Constrain [RFC4684] mechanisms SHOULD be used for Bitmask RT as well, as specified in [I-D.zzhang-idr-bgp-rt-constrain-extension].

### 3. Security Considerations

To be added.

### 4. IANA Considerations

To be added.

### 5. Acknowledgements

The authors thank Jeff Haas, Srihari Sangli and Colby Barth for their comments and suggestions.

## 6. References

### 6.1. Normative References

- [I-D.ietf-lsr-flex-algo]  
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-07 (work in progress), April 2020.
- [I-D.zzhang-idr-bgp-rt-constrains-extension]  
Zhang, Z. and J. Haas, "Route Target Constrain Extension", draft-zzhang-idr-bgp-rt-constrains-extension-00 (work in progress), July 2020.
- [I-D.zzhang-idr-bitmask-route-target]  
Zhang, Z., Ramachandra, S., and J. Haas, "Bitmask Route Target", draft-zzhang-idr-bitmask-route-target-00 (work in progress), July 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>>.
- [RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <<https://www.rfc-editor.org/info/rfc7752>>.
- [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>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.

### 6.2. Informative References

- [I-D.dong-network-slicing-problem-statement]  
Dong, J. and S. Bryant, "Problem Statement of Network Slicing in IP/MPLS Networks", draft-dong-network-slicing-problem-statement-00 (work in progress), October 2016.

- [I-D.drake-bess-enhanced-vpn]  
Drake, J., Farrel, A., Jalil, L., and A. Lingala, "BGP-LS Filters : A Framework for Network Slicing and Enhanced VPNs", draft-drake-bess-enhanced-vpn-03 (work in progress), May 2020.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <<https://www.rfc-editor.org/info/rfc3630>>.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <<https://www.rfc-editor.org/info/rfc4364>>.
- [RFC4684] Marques, P., Bonica, R., Fang, L., Martini, L., Raszuk, R., Patel, K., and J. Guichard, "Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)", RFC 4684, DOI 10.17487/RFC4684, November 2006, <<https://www.rfc-editor.org/info/rfc4684>>.
- [RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC 4915, DOI 10.17487/RFC4915, June 2007, <<https://www.rfc-editor.org/info/rfc4915>>.
- [RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-IS)", RFC 5120, DOI 10.17487/RFC5120, February 2008, <<https://www.rfc-editor.org/info/rfc5120>>.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <<https://www.rfc-editor.org/info/rfc5305>>.
- [RFC7308] Osborne, E., "Extended Administrative Groups in MPLS Traffic Engineering (MPLS-TE)", RFC 7308, DOI 10.17487/RFC7308, July 2014, <<https://www.rfc-editor.org/info/rfc7308>>.

#### Authors' Addresses

Zhaohui Zhang  
Juniper Networks

EMail: [zzhang@juniper.net](mailto:zzhang@juniper.net)



Shraddha Hegde  
Juniper Networks

EMail: shraddha@juniper.net

Arkadiy Gulko  
Refinitiv

EMail: arkadiy.gulko@refinitiv.com