TEAS Working Group
Internet Draft
Intended Status: Standard Track

Y. Lee (Editor)
Dhruv Dhody
Huawei
D. Ceccarelli
Ericsson
Takuya Miyasaka
KDDI
Peter Park
KT
Bin Young Yoon
ETRI

Expires: September 9, 2017 March 9, 2017

A Yang Data Model for ACTN VN Operation

draft-lee-teas-actn-vn-yang-03

Abstract

This document provides a YANG data model for the Abstraction and Control of Traffic Engineered (TE) networks (ACTN) Virtual Network (VN) operation.

Copyright Notice

Table of Contents

1. Introduction

This document provides a YANG data model for the Abstraction and Control of Traffic Engineered (TE) networks (ACTN) Virtual Network (VN) operation that is going to be 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 [Service-YANG]. The YANG model discussed in this document is used to operate customer-driven VNs during the VN computation, VN instantiation and its life-cycle operations stages.

Note that the YANG model presented in this draft has two aspects:

- VN pre-instantiation mode of operation (also known as VN compute);
- VN instantiation mode of operation.

The VN pre-instantiation mode of operation is concerned about service inquiry before making a formal request for VN instantiation. This operation is important for a customer to make sure the network can provide VN services it desires.

The VN instantiation mode of operation is concerned about instantiating VNs. In the VN instantiation mode, the CNC provides the VN service definition that includes VN members, VN service objective, VN service policy and preferences, etc. Upon receipt of a VN instantiation request, the MDSC (in coordination with PNCs) executes service request into network operation that include creating tunnels/paths and securing network resources/slices for VNs.

The YANG model discussed in this document basically provides the characteristics of VNs such as VN level parameters (e.g., VN ID, VN member, VN objective function, VN service preference, etc.), customer's end point characteristics (e.g., Customer Interface Capability, Access Points Interface characteristics, etc.), and other relevant VN information that needs to be known to the MDSC to facilitate ACTN VN operation.

1.1. Terminology

Refer to [ACTN-Frame] and [RFC7926] for the key terms used in this document.

2. ACTN CMI context

The model presented in this document has the following ACTN context.

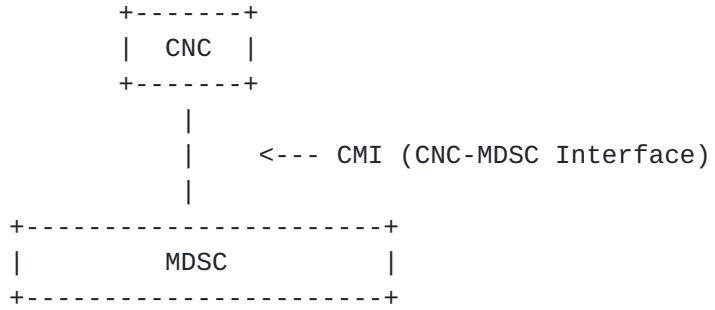


Figure 1. ACTN CMI

As defined in [ACTN-FW], a Virtual Network is a client view (typically a network slice) of the transport network. It can be presented by the provider as a set of physical and/or abstracted resources. Depending on the agreement between client and provider various VN operations and VN views are possible.

a) VN can be seen as an (or set of) e2e tunnel(s) from a customer point of view where an e2e tunnel is referred as a VN member. Each VN member (i.e., e2e tunnel) can then be formed by recursive aggregation of lower level paths at a provider level. Such end to end tunnels may comprise of customer end points, access links, intra domain paths and inter-domain link. In this view VN is thus a list of VN members.

b) VN can also be seen as a terms of topology comprising of physical and abstracted nodes and links. The nodes in this case include physical customer end points, border nodes, and internal nodes as well as abstracted nodes. Similarly the links includes physical access, inter-domain and intra-domain links as well as abstracted links. The abstracted nodes and links in this view can be pre-negotiated or created dynamically.

For both cases, the CNC can dynamically add VN elements. For case 1, the VN element is an end-to-end tunnel and for case 2, the VN element can be virtual nodes or virtual links.

In the subsequent discussion, the first form of VN will be discussed.

The following figure describes a VN that comprises three VN members forming a full mesh for the VN as an illustration.

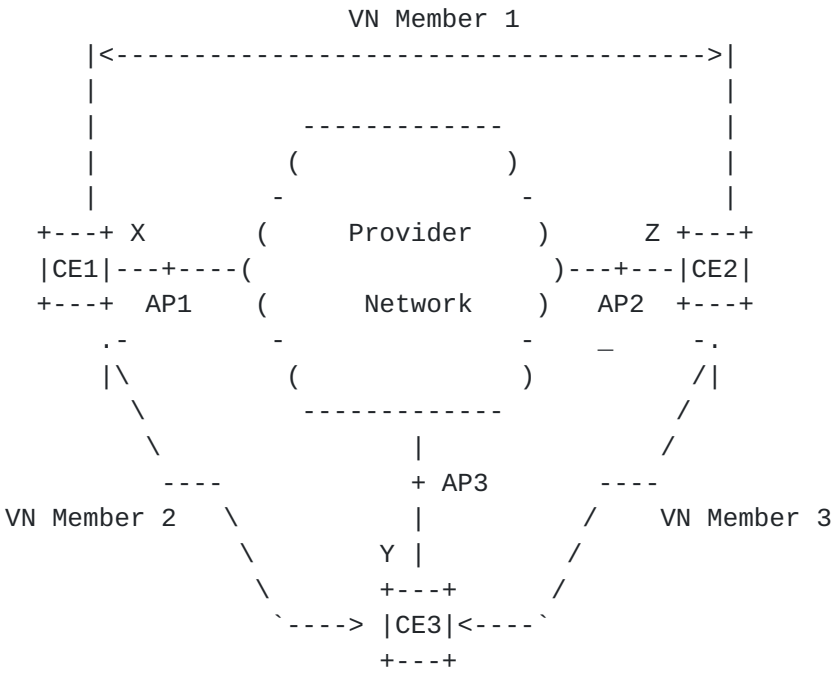


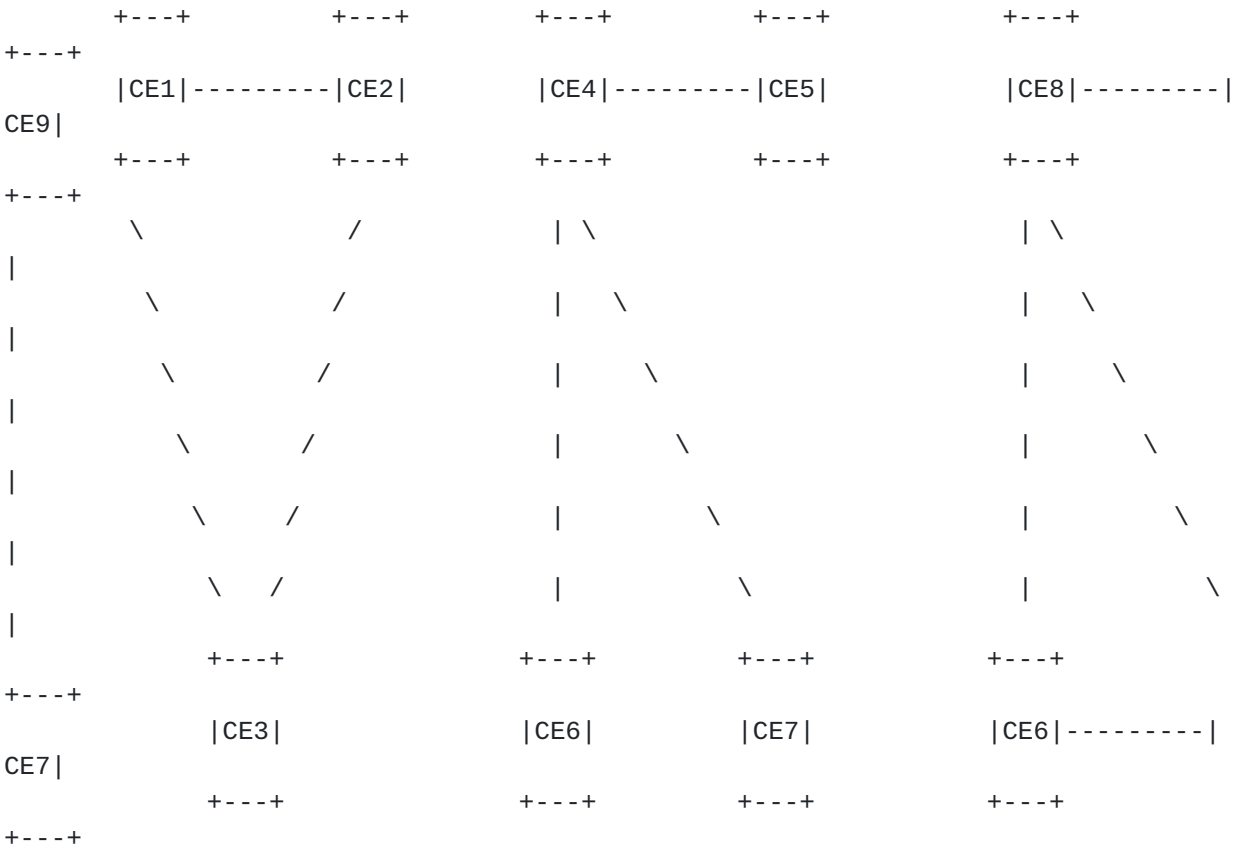
Figure 2. Full Mesh Example for a VN

In Figure 2, a VN has three members, namely, VN Member 1, VN member 2, and VN member 3. VN Member 1 is an end-to-end tunnel identified by CE1-AP1 (source) and CE2-AP2 (destination). Similarly, VN Member 2 by CE1-AP1 and CE3-AP3 and VN Member 3 by CE3-AP3 and CE2-AP2. This particular VN shown in Figure 2 is a full mesh connectivity across these three customer end-points.

The set of assumptions that applies to this document is the following:

- CNC is responsible for providing necessary Customer End-Points information to the MDSC via the CMI.
- The access links (between Customer Edge (CE) Devices and the Provider Edge (PE) Devices) are assumed to have been provisioned prior to the VN instantiation request. Access point identifiers have been configured and therefore are known in both the CNC and the MDSC.

It is also possible for the customer to create a VN which can be a hub and spoke or any other form of connectivity depending on its connectivity requirement. Each end-to-end tunnel may be unidirectional or bidirectional which is also depending on its connectivity requirements. The following figure shows some examples of a VN that can be represented in a different connectivity form depending on the customer's connectivity requirements.



(a) Full Mesh (b) Hub and Spoke (c) partial Mesh

Figure 3. Different Connectivity Forms of a VN

It is important to note that a VN can associate a multiple number of end-to-end tunnels (i.e., VN members) with one unique identifier. From a customer standpoint, this simplifies its VN operation significantly.

The MDSC interacts with the CNC for the VN operation. Once the customer VN is requested by the CNC to the MDSC, the MDSC shall be responsible for translating and mapping the VN request into specific network centric-models (e.g., TE-tunnels [TE-Tunnel], TE-topology [TE-TOPO], etc.) to coordinate the multi-domain network operations with PNCs.

2.1. Justification of the ACTN VN Model on the CMI.

1. It binds multiple e2e tunnels as a VN. VN is a unit for concurrency. TE-tunnel model deals each VN member as a separate entity, so it loses concurrent allocation of TE resources. TE-tunnel model is a sequential provisioning approach.

2. It is easier for some customers to work on VN level/Network slicing rather than individual TE tunnels.

3. ACTN VN supports multi-source and multi-destination which TE-tunnel model does not support. (See Section 2.2 for details)

4. ACTN VN supports VN compute (pre-instantiation mode) which TE-tunnel model does not.

5. There are certain advantages to keep a set of TE-tunnels as one VN unit for applying policy, reroute, protection, restoration, etc. rather than treating each TE-tunnel as individual unit.

2.2. 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. See details in Section 4.

```
   +--rw actn
         |  +--rw vn
         |     +--rw vn-list* [vn-id]
         |           ...
         |        |  +--rw src
         |        |  |  +--rw src?            -> /actn/ap/access-point-list/
access-
   point-id
         |        |  |  +--rw src-vn-ap-id?   uint32
         |        |  |  +--rw multi-src?      boolean
         |        |  +--rw dest
         |        |     +--rw dest?            -> /actn/ap/access-point-list/
access-
   point-id
         |        |     +--rw dest-vn-ap-id?   uint32
         |        |     +--rw multi-dest?      boolean

         +--ro actn-state
            +--ro vn
               +--ro vn-list* [vn-id]
                  |  +--ro src
                  |  |  +--ro src?            -> /actn/ap/access-point-list/
access-
   point-id
                  |  |  +--ro src-vn-ap-id?   uint32
                  |  |  +--ro multi-src?      boolean
                  |  +--ro dest
                  |  |  +--ro dest?            -> /actn/ap/access-point-list/
access-
   point-id
                  |  |  +--ro dest-vn-ap-id?   uint32
                  |  |  +--ro multi-dest?      boolean
                   ...
                  +--ro multi-src-dest
                  |  +--ro selected-vn-member?   -> /actn-state/vn/vn-list/vn-
id
```

3. ACTN VN YANG Model (Tree Structure)

```
module: ietf-actn-vn
    +--rw actn
    |  +--rw ap
    |  |  +--rw access-point-list* [access-point-id]
    |  |     +--rw access-point-id      uint32
```

```
    |  |     +--rw access-point-name?   string
    |  |     +--rw src-tp-id?           binary
    |  |     +--rw dst-tp-id?           binary
    |  |     +--rw max-bandwidth?       tet:te-bandwidth
```

```
    |  |     +--rw avl-bandwidth?       tet:te-bandwidth
    |  +--rw vn
    |     +--rw vn-list* [vn-id]
    |        +--rw vn-id                uint32
    |        +--rw vn-name?             string
    |        +--rw vn-member-list* [vn-member-id]
    |        |  +--rw vn-member-id    uint32
    |        |  +--rw src
    |        |  |  +--rw src?           -> /actn/ap/access-point-list/access-
point-id
    |        |  |  +--rw src-vn-ap-id?   uint32
    |        |  |  +--rw multi-src?      boolean
    |        |  +--rw dest
    |        |     +--rw dest?            -> /actn/ap/access-point-list/access-
point-
id
    |        |     +--rw dest-vn-ap-id?   uint32
    |        |     +--rw multi-dest?      boolean
    |        +--rw objective-function?   pcep:objective-function
    |        +--rw metric* [metric-type]
    |        |  +--rw metric-type    identityref
    |        |  +--rw limit
    |        |  |  +--rw enabled?   boolean
    |        |  |  +--rw value?     uint32
    |        |  +--rw optimize
    |        |     +--rw enabled?   boolean
    |        |     +--rw value?     uint32
    |        +--rw bandwidth?            tet:te-bandwidth
    |        +--rw protection?           identityref
    |        +--rw local-reroute?        boolean
    |        +--rw push-allowed?         boolean
    |        +--rw incremental-update?   boolean
    |        +--rw admin-status?         identityref
    +--ro actn-state
       +--ro ap
       |  +--ro access-point-list* [access-point-id]
       |     +--ro access-point-id      uint32
       |     +--ro access-point-name?   string
       |     +--ro src-tp-id?           binary
       |     +--ro dst-tp-id?           binary
       |     +--ro max-bandwidth?       tet:te-bandwidth
       |     +--ro avl-bandwidth?       tet:te-bandwidth
       +--ro vn
          +--ro vn-list* [vn-id]
             +--ro vn-id                uint32
             +--ro vn-name?             string
             +--ro vn-member-list* [vn-member-id]
             |  +--ro vn-member-id    uint32
```

```
             |  +--ro src
             |  |  +--ro src?            -> /actn/ap/access-point-list/access-
point-id
             |  |  +--ro src-vn-ap-id?   uint32
```

```
            |  |  +--ro multi-src?      boolean
            |  +--ro dest
            |  |  +--ro dest?            -> /actn/ap/access-point-list/access-
point-
id
            |  |  +--ro dest-vn-ap-id?   uint32
            |  |  +--ro multi-dest?      boolean
            |  +--ro metric* [metric-type]
            |  |  +--ro metric-type    identityref
            |  |  +--ro limit
            |  |  |  +--ro enabled?   boolean
            |  |  |  +--ro value?     uint32
            |  |  +--ro optimize
            |  |     +--ro enabled?   boolean
            |  |     +--ro value?     uint32
            |  +--ro oper-status?    identityref
            |  +--ro tunnel-ref?     te:tunnel-ref
            +--ro multi-src-dest
            |  +--ro selected-vn-member?   -> /actn-state/vn/vn-list/vn-id
            +--ro vn-topology-ref
            |  +--ro network-ref?   -> /nw:networks/network/network-id
            +--ro objective-function?   pcep:objective-function
            +--ro metric* [metric-type]
            |  +--ro metric-type    identityref
            |  +--ro limit
            |  |  +--ro enabled?   boolean
            |  |  +--ro value?     uint32
            |  +--ro optimize
            |     +--ro enabled?   boolean
            |     +--ro value?     uint32
            +--ro bandwidth?            tet:te-bandwidth
            +--ro protection?           identityref
            +--ro local-reroute?        boolean
            +--ro push-allowed?         boolean
            +--ro incremental-update?   boolean
            +--ro admin-status?         identityref
            +--ro oper-status?          identityref

  rpcs:
    +---x vn-compute
       +---w input
       |  +---w vn-member-list* [vn-member-id]
       |  |  +---w vn-member-id    uint32
       |  |  +---w src
       |  |  |  +---w src?           -> /actn/ap/access-point-list/access-
point-id
       |  |  |  +---w src-vn-ap-id?   uint32
       |  |  |  +---w multi-src?      boolean
```

```
       |  |  +---w dest
       |  |     +---w dest?            -> /actn/ap/access-point-list/access-
point-id
       |  |     +---w dest-vn-ap-id?   uint32
```

```
       |  |     +---w multi-dest?      boolean
       |  +---w objective-function?   pcep:objective-function
       |  +---w metric* [metric-type]
       |  |  +---w metric-type    identityref
       |  |  +---w limit
       |  |  |  +---w enabled?   boolean
       |  |  |  +---w value?     uint32
       |  |  +---w optimize
       |  |     +---w enabled?   boolean
       |  |     +---w value?     uint32
       |  +---w bandwidth?            tet:te-bandwidth
       |  +---w protection?           identityref
       |  +---w local-reroute?        boolean
       |  +---w push-allowed?         boolean
       |  +---w incremental-update?   boolean
      +--ro output
         +--ro vn-member-list* [vn-member-id]
         |  +--ro vn-member-id    uint32
         |  +--ro src
         |  |  +--ro src?            -> /actn/ap/access-point-list/access-
point-id
         |  |  +--ro src-vn-ap-id?   uint32
         |  |  +--ro multi-src?      boolean
         |  +--ro dest
         |  |  +--ro dest?            -> /actn/ap/access-point-list/access-
point-id
         |  |  +--ro dest-vn-ap-id?   uint32
         |  |  +--ro multi-dest?      boolean
         |  +--ro metric* [metric-type]
         |  |  +--ro metric-type    identityref
         |  |  +--ro limit
         |  |  |  +--ro enabled?   boolean
         |  |  |  +--ro value?     uint32
         |  |  +--ro optimize
         |  |     +--ro enabled?   boolean
         |  |     +--ro value?     uint32
         |  +--ro oper-status?    identityref
         +--ro multi-src-dest
            +--ro selected-vn-member-id?   uint32
```

4. ACTN-VN YANG Code

The YANG code is as follows:

```
<CODE BEGINS> file "ietf-actn-vn@2017-03-09.yang"

module ietf-actn-vn {
```

```
          namespace "urn:ietf:params:xml:ns:yang:ietf-actn-vn";
```

```
prefix "vn";

/* Import TE generic types */
import ietf-te-types {
    prefix "te-types";
}

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

import ietf-te {
    prefix "te";
}

import ietf-pcep {
   prefix "pcep";
}

organization
    "IETF Traffic Engineering Architecture and Signaling (TEAS)
    Working Group";

contact
    "Editor: Young Lee <leeyoung@huawei.com>
     Editor: Dhruv Dhody <dhruv.ietf@gmail.com>";

description
    "This module contains a YANG module for the ACTN 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.";

revision 2017-03-09 {
    description
        "initial version.";
    reference
        "TBD";
}

identity path-metric-delay {
   base te-types:path-metric-type;
   description
```

```
            "delay  path metric";
          }

          identity path-metric-delay-variation {
            base te-types:path-metric-type;
            description
              "delay-variation path metric";
          }

          identity path-metric-loss {
            base te-types:path-metric-type;
            description
              "loss path metric";
          }

         /*
          * Groupings
          */

         grouping access-point{
             description
                 "AP related information";
             leaf access-point-id {
                 type uint32;
                 description
                     "unique identifier for the referred
                     access point";
             }
             leaf access-point-name {
                 type string;
                 description
                     "ap name";
             }
             /*using direct tp-id for now as per ietf-te
               should check if reference is better*/
             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 max-bandwidth {
                type tet:te-bandwidth;
                description
                    "max bandwidth of the AP";
            }
            leaf avl-bandwidth {
                type tet:te-bandwidth;
                description
                    "available bandwidth of the AP";
            }
            /*add details and any other properties of AP,
            not associated by a VN
            CE port, PE port etc.

            */
        }//access-point

        grouping vn-member {
            description
                "vn-member is described by this container";
            leaf vn-member-id {
                type uint32;
                description
                    "vn-member identifier";
            }
            container src
            {
                description
                    "the source of VN Member";
                leaf src {
                    type leafref {
                        path "/actn/ap/access-point-list/access-point-id";
                    }
                    description
                        "reference to source AP";
                }
                leaf src-vn-ap-id{
                    type uint32;
                    description

                        "vn-ap-id";
                }
                leaf multi-src {
                    type boolean;
                    description
```

```
                    "Is 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 "/actn/ap/access-point-list/access-point-id";
                    }
                    description
                        "reference to destination AP";
                }
                leaf dest-vn-ap-id{
                    type uint32;
                    description
                        "vn-ap-id";
                }
                leaf multi-dest {
                    type boolean;
                    description
                        "Is destination part of multi-destination, where
                        only one of the destination is enabled";
                }
            }
        }//vn-member

        grouping policy {
            description
                "policy related to vn-member-id";
            leaf local-reroute {
                type boolean;
                description
                    "Policy to state if reroute
                    can be done locally";
            }
            leaf push-allowed {
                type boolean;
                description
                    "Policy to state if changes
                    can be pushed to the customer";
            }
            leaf incremental-update {
```

```
                type boolean;
                description
                    "Policy to allow only the
                    changes to be reported";
            }
        }//policy

        grouping metrics {
            description
                "metric related information";
            list metric{
                key "metric-type";
                description
                    "The list of metrics for VN";
                leaf metric-type {
                    type identityref {
                        base te-types:path-metric-type;
                    }
                    description
                        "The VN metric type.";
                }
                container limit {
                    description
                        "Limiting contraints";
                    leaf enabled{
                        type boolean;
                        description
                            "Limit contraint is enabled";
                    }
                    leaf value{
                        type uint32;
                        description
                            "The limit value";
                    }
                }
                container optimize{
                    description
                        "optimizing constraints";
                    leaf enabled{
                        type boolean;
                        description
                            "Metric to optimize";
                     }
                     leaf value{
                         type uint32;
```

```
                        description
                           "The computed value";
                    }
                }
            }

        }

        grouping service-metric {
            description
                "service-metric";
            leaf objective-function {
                type pcep:objective-function;
                    description
                        "operational state of the objective function";
            }

            uses metrics;

            leaf bandwidth {
                type tet:te-bandwidth;
                description
                    "bandwidth requested/required for
                    vn-member-id";
            }

            leaf protection {
                type identityref {
                    base te-types:lsp-prot-type;
                }
                description "protection type.";
            }

            uses policy;

        }//service-metric

        /*
         * Configuration data nodes
         */
        container actn {
            description
                "actn is described by this container";
            container ap {
                description
```

```
                "AP configurations";
             list access-point-list {
                key "access-point-id";
                description
                    "access-point identifier";
                uses access-point{
                    description
                        "access-point information";
                }
             }
            }
            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 uint32;
                        description
                            "a unique vn identifier";
                    }
                    leaf vn-name {
                        type string;
                        description "vn name";
                    }
                    list vn-member-list{
                        key "vn-member-id";
                        description
                            "List of VN-members in a VN";
                        uses vn-member;
                    }
                    uses service-metric;

                    leaf admin-status {
                        type identityref {
                            base te-types:state-type;
                        }
                        default te-types:state-up;
                        description "VN administrative state.";
                    }

                }//vn-list
```

```
            }//vn
          }//actn

          /*
           * Operational data nodes
           */

          container actn-state{
              config false;

              description
                  "actn is described by this container";

              container ap {
                  description
                      "AP state";
                  list access-point-list {
                      key "access-point-id";
                      description
                          "access-point identifier";
                      uses access-point{
                          description
                              "access-point information";
                      }
               }
              }
              container vn {
                  description
                      "VN state";
                  list vn-list {
                      key "vn-id";
                      description
                          "a virtual network is identified by a vn-id";
                      leaf vn-id {
                          type uint32;
                          description
                              "a unique vn identifier";
                      }
                      leaf vn-name {
                          type string;
                          description "vn name";
                      }
                      list vn-member-list{
                          key "vn-member-id";
                          description
```

```
                "List of VN-members in a VN";
            uses vn-member;
            uses metrics;
            leaf oper-status {
                type identityref {
                    base te-types:state-type;
                }
                description

                    "VN-member operational state.";
            }
            leaf tunnel-ref {
                type te:tunnel-ref;
                description
                    "A reference to the TE tunnel
                    in the TE model";
            }
        }
        container multi-src-dest{
            description
                "The selected VN Member when multi-src
                and/or mult-destination is enabled.";
            leaf selected-vn-member{
                type leafref {
                    path "/actn-state/vn/vn-list/vn-id";
                }
                description
                    "The selected VN Member along the set
                    of source and destination configured
                    with multi-source and/or multi-destination";
            }
        }
        container vn-topology-ref{
            description
                "An optional reference to the TE Topology
                 Model where the abstract nodes and links
                 of the Topology can be found";
             uses tet:te-topology-ref;
        }

        uses service-metric;

        leaf admin-status {
            type identityref {
                base te-types:state-type;
```

```
                        }
                        description "VN administrative state.";
                    }
                    leaf oper-status {
                        type identityref {
                            base te-types:state-type;
                        }
                        description "VN operational state.";
                    }

                }//vn-list
            }//vn
        }//actn-state
        /*
        * Notifications - TBD
        */
        /*
        * RPC
        */
        rpc  vn-compute{
            description
                "The VN computation without actual
                instantiation";
            input {
                list vn-member-list{
                    key "vn-member-id";
                    description
                        "List of VN-members in a VN";
                    uses vn-member;
                }
                uses service-metric;
            }
            output {
                list vn-member-list{
                    key "vn-member-id";
                    description
                        "List of VN-members in a VN";
                    uses vn-member;
                    uses metrics;
                    leaf oper-status {
                        type identityref {
                            base te-types:state-type;
                        }
                        description
                            "VN-member operational state.";
```

```
                    }
                }
                container multi-src-dest{
                    description
                        "The selected VN Member when multi-src
                        and/or mult-destination is enabled.";
                    leaf selected-vn-member-id{
                        type uint32;
                        description
                            "The selected VN Member-id from the
                            input";
                    }
                }
            }
        }
    }
```

<CODE ENDS>

5. Security Considerations

TDB

6. IANA Considerations

TDB

7. Acknowledgments

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

8. References

8.1. Normative References

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

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

[Service-YANG] Q. Wu, W. Liu and A. Farrel, "Service Models Explained", draft-wu-opsawg-service-model-explained, work in progress.

8.2. Informative References

[RFC7926] A. Farrel (Ed.), "Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks", RFC 7926, July 2016.

[ACTN-REQ] Lee, et al., "Requirements for Abstraction and Control of TE Networks", work in progress: draft-ietf-teas-actn-requirements.

[ACTN-FWK] D. Ceccarelli, Y. Lee [Editors], "Framework for Abstraction and Control of Traffic Engineered Networks", work in progress: draft-ceccarelli-teas-actn-framework.

9. Contributors

Contributor's Addresses

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

Authors' Addresses

Young Lee (ed.)
Huawei Technologies
Email: leeyoung@huawei.com

Dhruv Dhody
Huawei Technologies
Email: dhruv.ietf@gmail.com

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

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

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

Bin Yeong Yoon
ETRI
Email: byyun@etri.re.kr