A Yang Data Model for ACTN VN Operation

draft-lee-teas-actn-vn-yang-12

Young Lee, Dhruv Dhody, Igor Bryskin  Huawei
Daniele Ceccarelli  Ericsson
Bin Yeong Yoon  ETRI
Haomian, Xian, Sergio, Qin, Takuya, Peter  Contributors
Introduction

• A YANG data model for the ACTN Virtual Network Service (VNS) operation that is going to be implemented for CMI (between CNC and MDSC).
  • Aligned to Customer Service Model
  • VN Instantiation, VN Computation, VN Lifecycle
  • Access Points (AP) and Virtual Network Access Points (VNAP)
  • Virtual Network (VN)
    • List of VN members
  • Other VN Operations
    • Multi-src/Multi-dest
  • As per the ACTN Informational Model
Major Change

- Coupling with the TE Topology Yang Model
  - Removed any duplicated information that can be found in TE Topology
    - Constraints
    - Explicit Path
    - Underlay path
    - Etc.
- Simplification of model for both VN Type 1 & VN Type 2
- JSON examples are added
Overview

ACTN VN Yang Model

AP/VNAP

VN

VN-Member

For both VN Type 1 or Type 2 VN Yang model rely on a single node in the abstract TE Topology.

The abstract node has

- connectivity-matrices
- connectivity-matrix [id]

The attributes directly under container connectivity-matrices are the default attributes for all connectivity-matrix entries when the per entry corresponding attribute is not specified. When a per entry attribute is specified, it overrides the corresponding attribute directly under the container connectivity-matrices.
Overview

- ACTN VN
  - AP
    - VNAP
      - VN-Member
        - From
          - Constraints?
            - Optimization?
          - To
  - VN

- /nw:networks/nw:network/nw:node/te-node-id

- Node
  - connectivity-matrices
    - underlay
      - path-constraints
      - computed-path-properties
      - From
      - To
      - primary-path
        - tunnel-termination-points
        - tunnel
        - Optimizations
          - path-constraints
          - underlay
            - Optimizations
              - tunnel

- global
  - Per entry

- TEAS WG @ IETF 101 5
### How: Reference to TE Topology Yang Model

| Access                                      | • AP/ VNAP -> LTP  
|                                            | • Ltp of type te-types:te-tp-id |
| Customer’s Virtual Network                 | • VN -> Abstract Node  
|                                            | • vn-topology-id of type te-types:te-topology-id  
|                                            | • abstract-node -> /nw:networks/network/node/tet:te-node-id (reference) |
| Customer Site to Site connection           | • VN-Member -> Entry in the connectivity matrix of the abstract node  

All parameters which are can be set as global attributes to VN are set in the connectivity-matrices (such as bandwidth) and an attribute for a particular VN-member is set in the connectivity-matrix [id] (such as explicit path)! Duplicate parameters in ACTN VN Yang model are removed!
VN Type 1

• VN 1
  • VN-Member 1  L1-L4
  • VN-Member 2  L1-L7
  • VN-Member 3  L2-L4
  • VN-Member 4  L3-L8
  • This VN has following properties
    • Bandwidth 500
    • Optimize by delay

• VN is seen as edge to edge links (VN-members) setup as tunnels across underlying networks!

Abstract Topology with a Single Node

These properties are set in TE Topo

Connectivity Matrix

<table>
<thead>
<tr>
<th>ID</th>
<th>Connectivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>L1-L4</td>
</tr>
<tr>
<td>17</td>
<td>L1-L7</td>
</tr>
<tr>
<td>24</td>
<td>L2-L4</td>
</tr>
<tr>
<td>38</td>
<td>L3-L8</td>
</tr>
</tbody>
</table>
VN Type 2

- VN is seen as a topology of virtual nodes and links
- To ease mapping between VN Yang Model and TE models, an abstract single node topology is created with VN topology as the underlay!
- The same mapping as VN Type 1 is reused.
- VN 2
  - VN-Member 1: L1-L8 via S3, S67110, S11
    - Set via the underlay path in connectivity-matrix[id]
Multi-Src / Multi-Dest

- Multiple VN members are configured and marked with multi-src or multi-dest option
- The MDSC selects the VN member to actually setup and create a connectivity matrix entry in the single node abstract topology based on the selected.
- MDSC is free to change the selected VN-member in coordination with CNC

<table>
<thead>
<tr>
<th>VN-Member 1</th>
<th>L1-L4(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VN-Member 2 (selected)</td>
<td>L1-L7(*)</td>
</tr>
<tr>
<td>VN-Member 3</td>
<td>L2(*)-L4</td>
</tr>
<tr>
<td>VN-Member 4 (selected)</td>
<td>L3(*)-L4</td>
</tr>
<tr>
<td>(*)</td>
<td>Multi-src or Multi-destination enabled</td>
</tr>
</tbody>
</table>
## Role of ACTN VN Model

### Customer view of VN
- VN as a single entity (as per the ACTN info model)
- ~ Service Model

### Innovative Services
- VN Compute
- Multi-Src / Multi-Dest

### Service Mapping

### AP / VNAP
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 max-bandwidth?      te-types:te-bandwidth
      |     +--rw avl-bandwidth?      te-types:te-bandwidth
      |     +--rw vn-ap* [vn-ap-id]
      |        +--rw vn-ap-id        uint32
      |        +--rw vn?            -->/actn/vn/vn-list/vn-id
      |        +--rw ltp?           te-types:te-ltp-id
    +--rw vn
      +--rw vn-list* [vn-id]
        +--rw vn-id        uint32
        +--rw vn-name?     string
        +--rw vn-topology-id?  te-types:te-topology-id
      +--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?    -->/actn/ap/access-point-list/vn-ap/vn-ap-id
      |     +--rw multi-src?       boolean (multi-src-dest)!
      |     +--rw dest
      |     |     +--rw dest?         -->/actn/ap/access-point-list/access-point-id
      |     |     +--rw dest-vn-ap-id?  -->/actn/ap/access-point-list/vn-ap/vn-ap-id
      |     |     +--rw multi-dest?     boolean (multi-src-dest)!
      |     |     +--ro oper-status?     identityref
      |     |     +--ro if-selected?     boolean (multi-src-dest)!
      |     |     +--rw admin-status?    identityref
      |     |     +--ro oper-status?     identityref
      |     +--rw vn-level-diversity? vn-disjointness
Next Step

• This update of draft is based on feedback received during the last IETF to reuse existing TE yang model as much as possible.
• This update does just that and works well with abstract TE topology model
• Request for WG Adoption!
Thank You!
Backup Slides
Typical Interactions: VN Type 1

CNC

Set Attributes, Connectivity Matrix, via Single Node Abstract TE Topology

200 OK

MDSC

Create VN with VN members, VNAP and map it to the TE Topology model
POST /actn/vn/vn-list/vn-id

200 OK

Get VN Status, selected VN-members in case of multi-src/multi-dest
GET /actn/vn/vn-list/vn-id
Typical Interactions: VN Type 2

Set Attributes, Connectivity Matrix, via Single Node Abstract TE Topology

Explicitly creating a single node topology

200 OK
Create VN with VN members, VNAP and map it to the TE Topology model
POST /actn/vn/vn-list/vn-id

200 OK
Get VN Status, selected VN-members in case of multi-src/multi-dest
GET /actn/vn/vn-list/vn-id

The CNC creates a single node topology by itself, in this way VN Type 1 and VN Type 2 would have similar interactions
Typical Interactions: VN Type 2

**CNC**
- Create VN with VN members, VNAP first
  - POST /actn/vn/vn-list/vn-id

  - 200 OK

  - Get VN Status and information related to the single node abstract topology and mapping
    - GET /actn/vn/vn-list/vn-id

  - Set any changes in attributes, Connectivity Matrix, via Single Node Abstract TE Topology

  - 200 OK

**MDSC**
- MDSC populates a single node abstract node topology by itself

The MDSC creates a single node topology by itself and maps it to the ACTN VN Yang model. Any attribute changes are done next via the te topology model.