IETF Network Slice Topology YANG Data Model

draft-liu-teas-transport-network-slice-yang-06

Co-authors:

Xufeng Liu (IBM)
Jeff Tantsura (Microsoft)
Igor Bryskin (Individual)
Luis M. Contreras (Telefonica)
Qin Wu (Huawei)
Sergio Belotti (Nokia)
Reza Rokui (Ciena)
Aihua Guo (Futurewei)
Italo Busi (Huawei)

Motivation

- A network slice may require an additional level of control to customize its connections, as described in Section 2 of draft-ietf-teas-ietf-network-slices.
  - "*Additionally, the IETF Network Slice service customer might ask for some level of control of, e.g., to customize the service paths in a network slice.*"

- A network slice can benefit from using a customized topology to allow better resource reservation and resource sharing among connections
  - We call it a "network slice topology"

Network Slice Topology

- A network slice topology is a customized topology used to expresses a customer's intent for reserving topology resources and expressing connectivity constraints
  - A network slice topology is defined by the customer("customized"), not one exposed by the provider("abstract" or "native" topology)
  - Existing topology models, e.g. network topology or TE topology is not designed for expressing the slice intent, e.g. no customer SLO/SLEs defined
- Service paths for connections in a network slice can be specified on top of a network slice topology
- The topology model defined in early revisions of draft-liu-teas-transport-network-slice-yang is a good candidate for defining the network slice topology

Updates in Rev-06

- Imported common SLO/SLE attributes from ns-nbi & applied them to the topology constructs
- Removed the augments from the TE topology model for model connectivity constructs as they are already covered by the ns-nbi YANG model
- Text & diagram updates

Updated Model Relationships

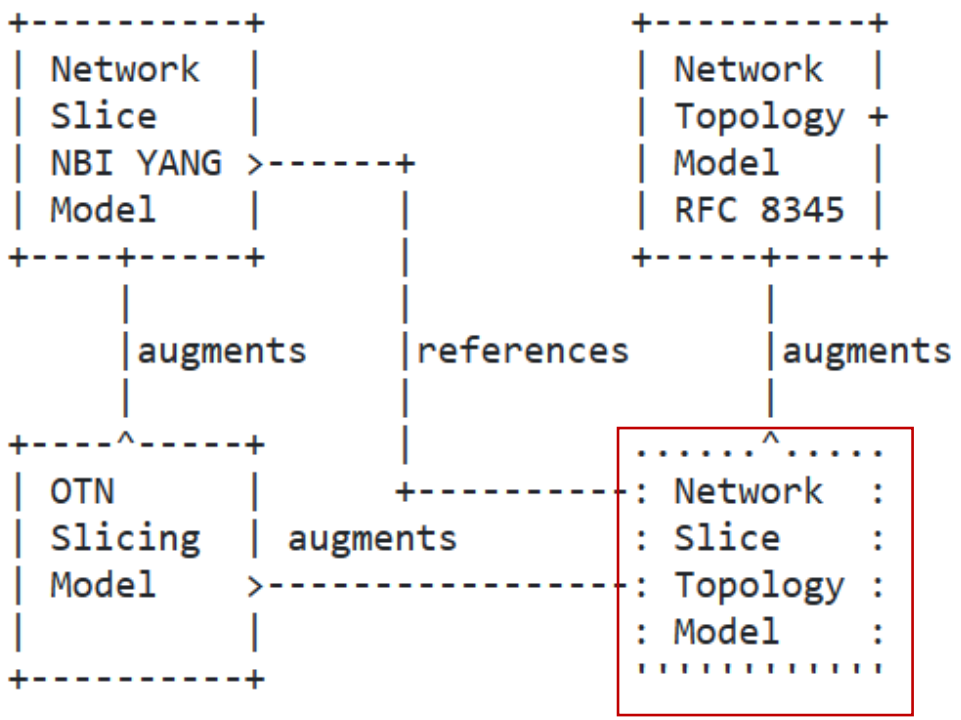

Figure 1: Model Relationships

Model Tree

```
augment /nw:networks/nw:network/nw:network-types:
  +--rw network-slice!
augment /nw:networks/nw:network:
  +--rw (slo-sle-policy)?
     +--:(standard)
     |  +--rw slo-sle-template?         leafref
     +--:(custom)
        +--rw service-slo-sle-policy
           +--rw description?              string
           +--rw metric-bounds
           |  +--rw metric-bound* [metric-type]
           |     +--rw metric-type          identityref
           |     +--rw metric-unit          string
           |     +--rw value-description?   string
           |     +--rw percentile-value?    percentile
           |     +--rw bound?               uint64
           +--rw security*                  identityref
           +--rw isolation?                 identityref
           +--rw max-occupancy-level?       uint8
           +--rw mtu?                       uint16
           +--rw steering-constraints
           |  +--rw path-constraints
           |  +--rw service-function
           |  +--rw disjointness?
           |          te-types:te-path-disjointness
           +--rw optimization-criterion?   identityref
           +--rw resize-requirement?       identityref
           +--rw service-info?             string
```

Network augments

```
augment /nw:networks/nw:network/nw:node:
  +--rw (slo-sle-policy)?
     +--:(standard)
     |  +--rw slo-sle-template?         leafref
     +--:(custom)
        +--rw service-slo-sle-policy
           +--rw description?              string
           +--rw metric-bounds
           |  +--rw metric-bound* [metric-type]
           |     +--rw metric-type          identityref
           |     +--rw metric-unit          string
           |     +--rw value-description?   string
           |     +--rw percentile-value?    percentile
           |     +--rw bound?               uint64
           +--rw security*                  identityref
           +--rw isolation?                 identityref
           +--rw max-occupancy-level?       uint8
           +--rw mtu?                       uint16
           +--rw steering-constraints
           |  +--rw path-constraints
           |  +--rw service-function
           |  +--rw disjointness?
           |          te-types:te-path-disjointness
           +--rw optimization-criterion?   identityref
           +--rw resize-requirement?       identityref
           +--rw service-info?             string
```

Node augments

Model Tree (Cont.)

```
augment /nw:networks/nw:network/nt:link:
  +--rw (slo-sle-policy)?
     +--:(standard)
     |  +--rw slo-sle-template?         leafref
     +--:(custom)
        +--rw service-slo-sle-policy
           +--rw description?              string
           +--rw metric-bounds
           |  +--rw metric-bound* [metric-type]
           |     +--rw metric-type          identityref
           |     +--rw metric-unit          string
           |     +--rw value-description?   string
           |     +--rw percentile-value?    percentile
           |     +--rw bound?               uint64
           +--rw security*                  identityref
           +--rw isolation?                 identityref
           +--rw max-occupancy-level?       uint8
           +--rw mtu?                       uint16
           +--rw steering-constraints
           |  +--rw path-constraints
           |  +--rw service-function
           |  +--rw disjointness?
           |          te-types:te-path-disjointness
           +--rw optimization-criterion?    identityref
           +--rw resize-requirement?        identityref
           +--rw service-info?              string
```

Link augments

```
augment /nw:networks/nw:network/nw:node/nt:termination-point:
  +--rw (slo-sle-policy)?
     +--:(standard)
     |  +--rw slo-sle-template?         leafref
     +--:(custom)
        +--rw service-slo-sle-policy
           +--rw description?              string
           +--rw metric-bounds
           |  +--rw metric-bound* [metric-type]
           |     +--rw metric-type          identityref
           |     +--rw metric-unit          string
           |     +--rw value-description?   string
           |     +--rw percentile-value?    percentile
           |     +--rw bound?               uint64
           +--rw security*                  identityref
           +--rw isolation?                 identityref
           +--rw max-occupancy-level?       uint8
           +--rw mtu?                       uint16
           +--rw steering-constraints
           |  +--rw path-constraints
           |  +--rw service-function
           +--rw optimization-criterion?    identityref
           +--rw resize-requirement?        identityref
           +--rw service-info?              string
```

TP augments

Open Issue

- Since the NS Topology model is a NS technology agnostic topology model, why not reference directly this model in NS-NBI for customized topology
  - Add constraint into NS-NBI that the reference topology must be of type “network-slice”

Open Issue – Relationship between SDP and LTP

- Current NS NBI model defines an SDP to reference an LTP object

```
+--rw sdps
  +--rw sdp* [id]
     +--rw id                        string
     +--rw description?              string
     +--rw location
     |     ...
     +--rw node-id?                  string
     +--rw sdp-ip-address*           inet:ip-address
     +--rw tp-ref?                   -> /nw:networks/network[nw:network-id =current()/../../../custom-
topology-ref/network-ref]/node/nt:termination-point/tp-id
```

- Should the reference be the other way around, i.e. an LTP object to reference an SDP?

```
augment /nw:networks/nw:network/nw:node/nt:termination-point:
  +--rw customer-facing-sdp-id?         Leafref
```

- Can multiple LTPs in different topologies point to the same SDP?
- Should an SDP exist before or at the time the LTP is created?

Next Steps

- Request for WG adoption
- Resolve open issues

* GitHub Repo
  https://github.com/aguoietf/ietf-network-slice-topology

Thank You!