IETF Network Slice NBI YANG

draft-wd-teas-ietf-network-slice-nbi-yang

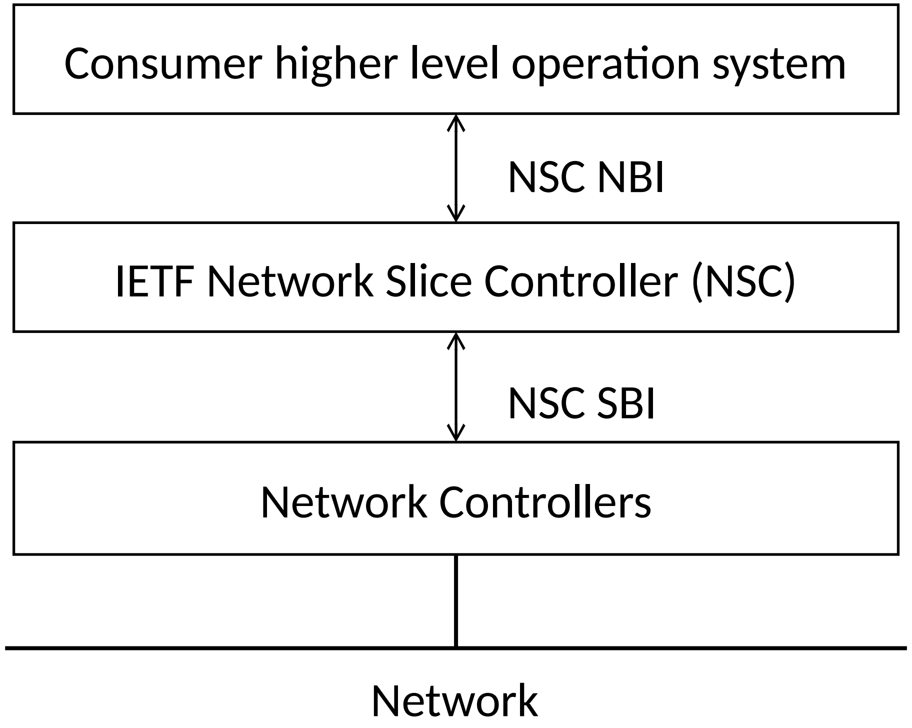
TEAS WG

Nov 2020

Bo Wu (Presenting), Dhruv Dhody Huawei
Liuyan Han China Mobile
Reza Rokui Nokia

Draft Status

- This draft is intended to define IETF Network Slice NBI (Northbound) YANG model. And has been discussed in the NS-DT and WG several times. This update aims to keep this draft consistent with the IETF Network Slice definition draft.
  - Notice the name change
- To support IETF Network Slice service and network management automation, this IETF Network Slice NBI YANG module and other YANG modules at different levels (or at the same level) need to be integrated.


Consumer higher level operation system
NSC NBI
IETF Network Slice Controller (NSC)
NSC SBI
Network Controllers
Network

NSC NBI: Service Model
(consumer view)
NS (IETF Network Slice), NS SLO,
NSE (IETF Network Slice Endpoint)

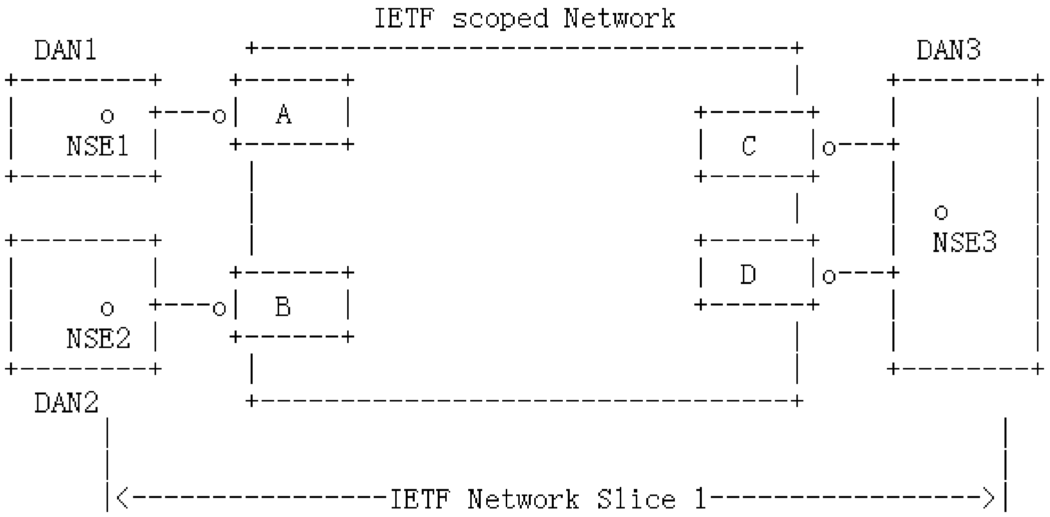
Network Model
(Operator view)
Network, topology, node, link, etc

Device Model
Interface, Tunnel, QoS/TE, …

Based on **RFC8199, RFC8309**

IETF Network Slice NBI YANG Key components

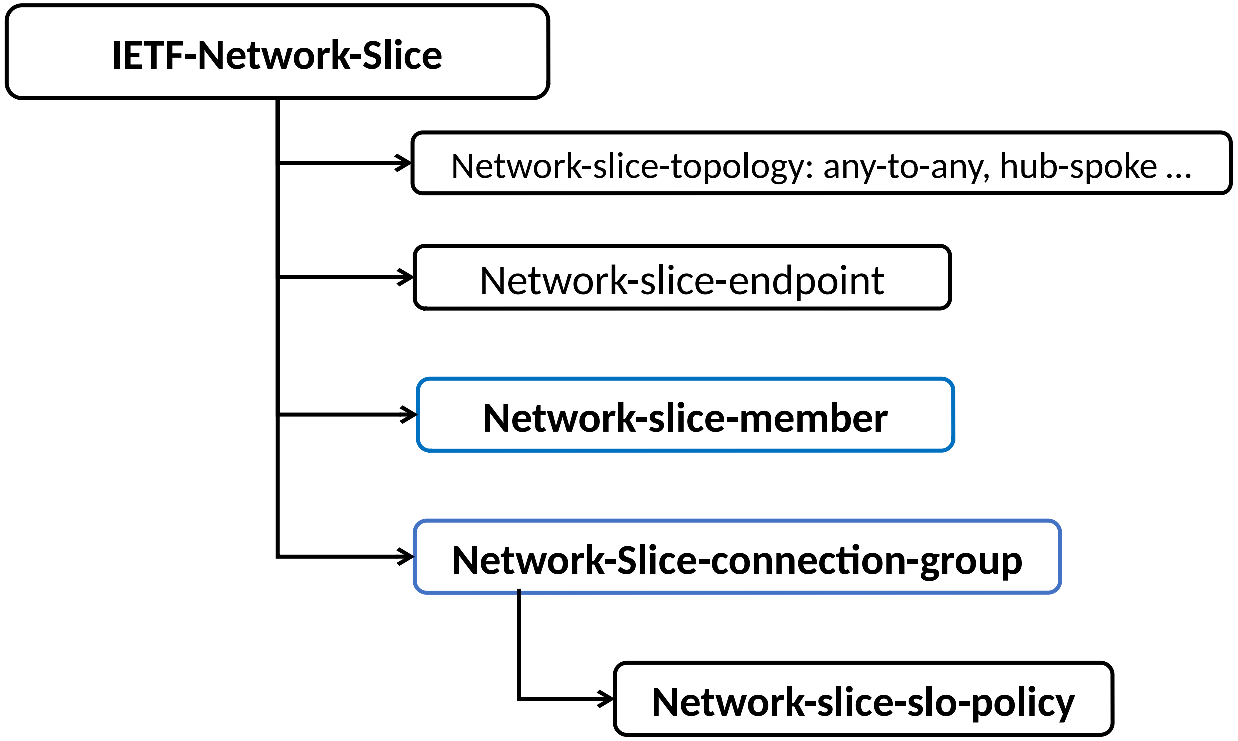
- Modelling consideration: support IETF Network Slice configuration and monitoring
- YANG components, with the ones defined in the IETF network slice definition draft
  - Network-slice-connection-group: a set of Network-slice-members with same SLOs in one IETF network slice
  - Network-slice-member: a particular connection between a pair of Network-slice-endpoint (NSE)



Legend:DAN (Device, Application, Network Function)

Connection Group Green
Topology type: Hub & Spoke NSE1-NSE3, NSE2-NSE3
SLO: Green

Connection Group Red
Topology type: Any to Any NSE1-NSE2
SLO: Red



Open issues

- IETF Network-slice-connection-group definition
  - Our modelling assumption is multiple SLOs for one IETF Network Slice, which means different SLOs for different set of connections within an IETF NS. Is this a right approach?

- Monitoring parameters
  - We are currently maintaining SLO monitoring status at different granularity, including per connection, and per connection-group. Is there any objection to this approach?

Next Step

- Solicit comments and reviews from WG

Backup

```
module: ietf-network-slice
  +--rw ietf-network-slices
     +--rw slice-templates
     |  +--rw slo-template* [id]
     |     +--rw id                     string
     |     +--rw template-description?   string
     +--rw ietf-network-slice* [network-slice-id]
        +--rw network-slice-id                 uint32
        +--rw network-slice-name?              string
        +--rw network-slice-tag*               string
        +--rw network-slice-topology*          identityref
        +--rw network-slice-connection-group* [connection-group-name]
        |  +--rw connection-group-name             string
        |  +--rw default-connection-group?         boolean
        |  +--rw (slo-template)?
        |  |  +--:(standard)
        |  |  |  +--rw template?                   leafref
        |  |  +--:(custom)
        |  |     +--rw network-slice-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 network-slice-metric-bounds
        |  |           +--rw network-slice-metric-bound*
        |  |                   [metric-type]
        |  |              +--rw metric-type    identityref
        |  |              +--rw upper-bound?   uint64
        |  +--rw network-slice-member-group*
        |  |      [network-slice-member-id]
        |  |  +--rw network-slice-member-id    leafref
        |  +--ro connection-group-monitoring
        |     +--ro latency?   uint32
        |     +--ro jitter?    uint32
        |     +--ro loss?      decimal64
        +--rw status
        |  +--rw admin-enabled?   boolean
        |  +--ro oper-status?     operational-type
```

```
+--rw network-slice-endpoint* [endpoint-id]
|  +--rw endpoint-id                     uint32
|  +--rw endpoint-name?                  string
|  +--rw endpoint-role*                  identityref
|  +--rw geolocation
|  |  +--rw altitude?    int64
|  |  +--rw latitude?    decimal64
|  |  +--rw longitude?   decimal64
|  +--rw node-id?                        string
|  +--rw port-id?                        string
|  +--rw network-slice-match-criteria
|  |  +--rw network-slice-match-criteria* [match-type]
|  |     +--rw match-type    identityref
|  |     +--rw value?        string
|  +--rw endpoint-ip?                    inet:host
|  +--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 routing
|  |  +--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 endpoint-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 network-slice-member* [network-slice-member-id]
   +--rw network-slice-member-id            uint32
   +--rw src
   |  +--rw src-network-slice-endpoint-id?   leafref
   +--rw dest
   |  +--rw dest-network-slice-endpoint-id?   leafref
   +--rw monitoring-type?
   |       network-slice-monitoring-type
   +--ro network-slice-member-monitoring
      +--ro latency?   uint32
      +--ro jitter?    uint32
      +--ro loss?      decimal64
```