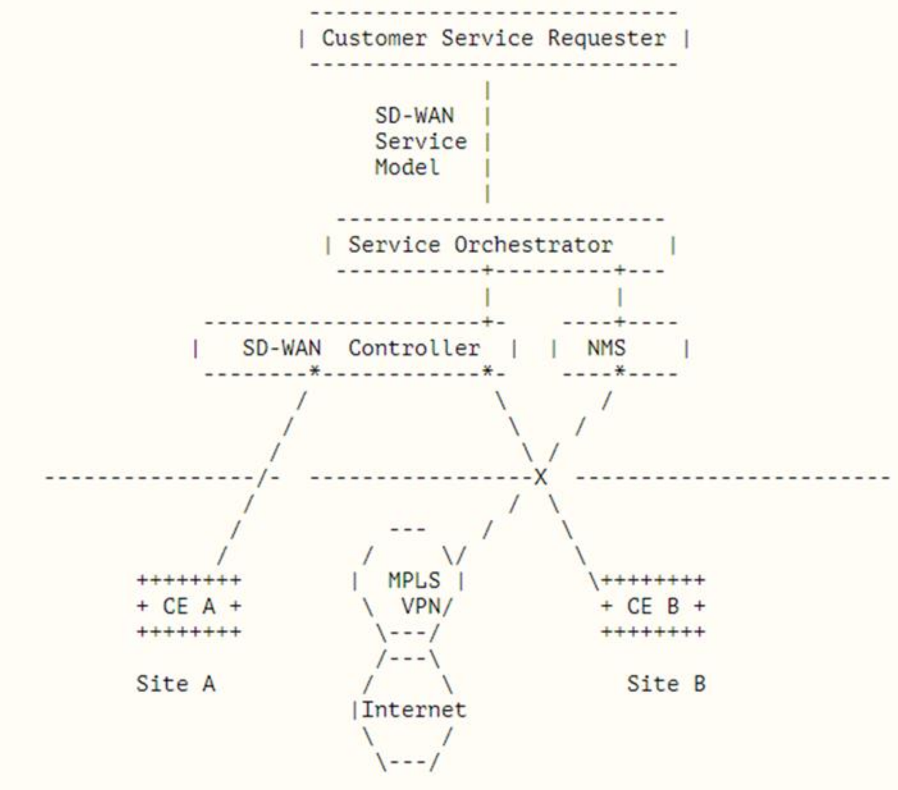
A YANG Data Model for SD-WAN service delivery

draft-sun-opsawg-sdwan-service-model-04

Qiong Sun, China Telecom
Bo Wu(Presenting), Qin Wu, Huawei
Charles Eckel, Cisco

What is SD-WAN service model

- SD-WAN service is a connectivity service offered by a service provider network to provide connectivity across different customer sites using one of more underlay networks
- A service providers can use this model to dynamically create, modify, and manage SD-WAN service components



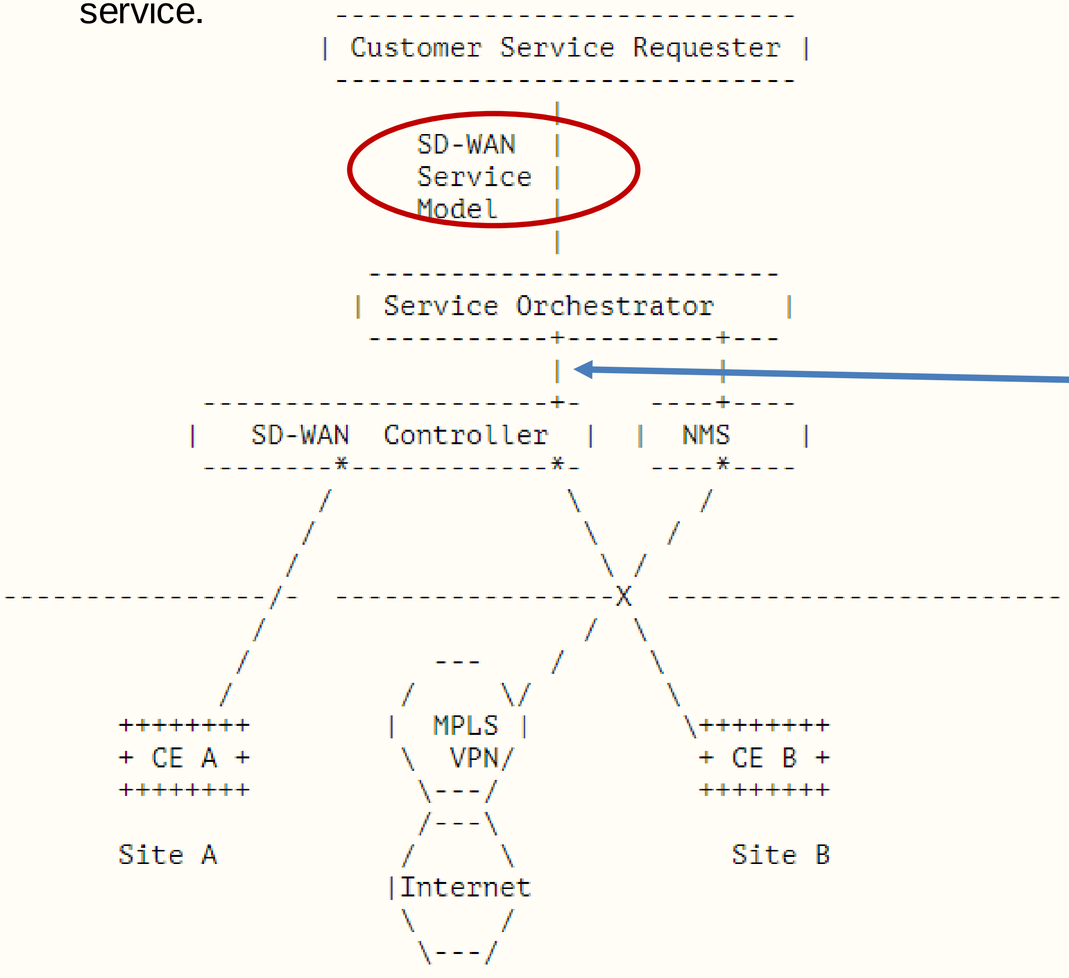
Reference Architecture for the Use of SD-WAN Service Model Usage

Changes since -02 (IETF 104)

- New co-author: Charles Eckel
- MEF SD-WAN project alignment
  - Added MEF-related references and terminology comparisons, MEF 70 Draft (R1) (SD-WAN Service Attributes and Services) has published
  - Made editorial changes to the entire draft to better align with the MEF SD-WAN project and improve readability
  - Highlight SD-WAN application based policy service, because SD-WAN has **application-based multipath selection** features
- draft-wood-rtgwg-sdwan-ose-yang (YANG Data Model for SD-WAN OSE service delivery) comparisons

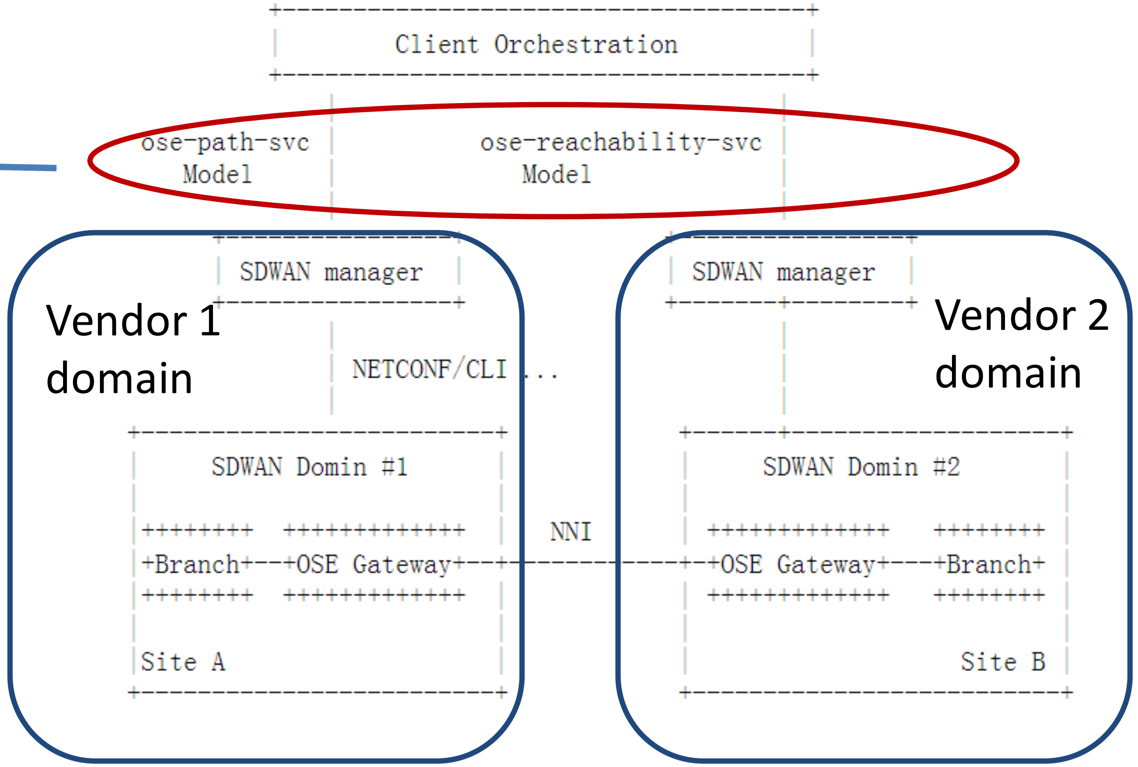
Issue : OSE model difference

- SD-WAN service model: high-level interface to the customer, upon user's request, service orchestrator can add a new site, VPN or application policy in real time
- OSE service model: The assumptions of the orchestrator function are different. The responsibility for SD-WAN infrastructure service configuration lies entirely within the domain SDWAN manager. The OSE draft defines OSE Gateway service and inter-domain path service.



```
              ----------------------------
             | Customer Service Requester |
              ----------------------------
                           |
                 SD-WAN    |
                 Service   |
                 Model     |
                           |
               -------------------------
              | Service Orchestrator    |
               -----------+---------+---
                          |         |
      --------------------+-    ----+----
     |   SD-WAN  Controller  |  |  NMS    |
      --------*-----------*-    ----*----
             /             \       /
            /               \     /
           /                 \   /
 ---------------/-   ----------------X   ----------------------
         /                  /   \
        /         ---      /     \
       /         /   \/         \
  ++++++++      |  MPLS |         \++++++++
  + CE A +      \  VPN/            + CE B +
  ++++++++       \---/             ++++++++
                 /---\
  Site A        /     \             Site B
               |Internet
                \     /
                 \---/
```

Reference Architecture for the Use of SD-WAN Service Model Usage



```
+------------------------------------+
|        Client Orchestration        |
+------------------------------------+
        |                      |
ose-path-svc |        ose-reachability-svc |
   Model     |              Model          |
        |                      |
   +-------------------+     +------------------+
   | SDWAN manager |         | SDWAN manager |
   +-------------------+     +------------------+
        |                      |
        | NETCONF/CLI ...      |
        |                      |
+------------------------+   +------------------------+
|    SDWAN Domin #1      |   |    SDWAN Domin #2      |
|                        |NNI|                        |
|++++++++  +++++++++++++ |   | +++++++++++++  ++++++++|
|+Branch+--+OSE Gateway+-+---+-+OSE Gateway+--+Branch+|
|++++++++  +++++++++++++ |   | +++++++++++++  ++++++++|
|                        |   |                        |
|Site A                  |   |                 Site B |
+------------------------+   +------------------------+
```

Vendor 1 domain

Vendor 2 domain

Open discussion

1. **MEF SD-WAN project coordination approach**

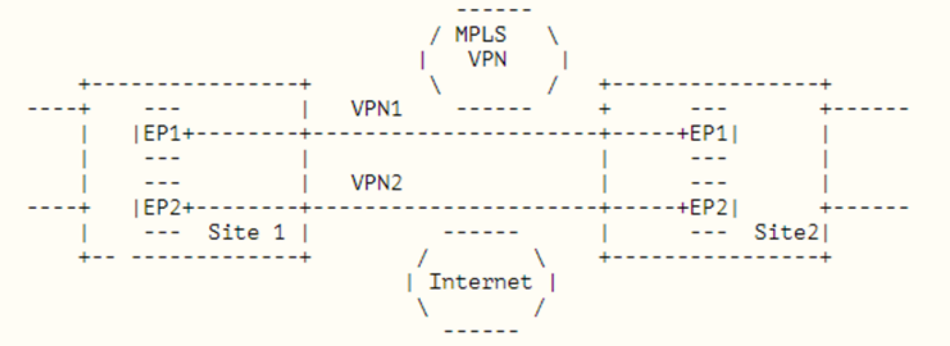
2. **draft-wood-rtgwg-sdwan-ose-yang-01 coordination approach**
   - ➢Align terminology of the two drafts
   - ➢Use the Grouping statement on the model component to allow reuse, such as site, vpn, application policy grouping

Next Steps

- Seeking WG adoption
- Solicit more comments
  - Your comments and suggestions are welcome!

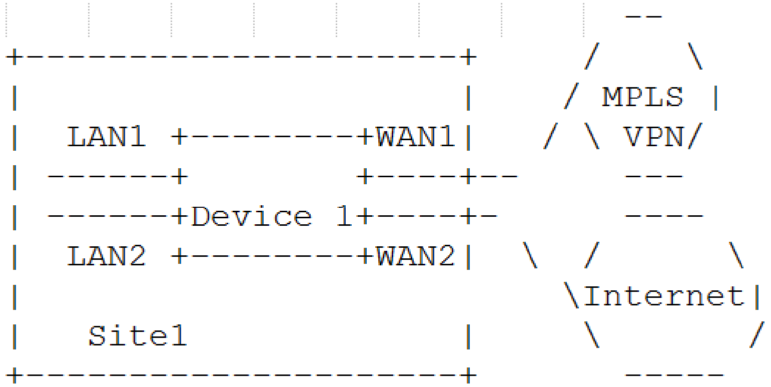
SD-WAN service component – “vpn” and “sites”

- The service orchestrator of a service provider can request, configure, and manage “vpn-service” and “sites”.



```
                           ------
                          / MPLS    \
                         |    VPN     |
    +----------------+    \          /   +----------------+
----+    ---         |  VPN1    ------   +      ---       +------
    |   |EP1+--------+-------------------+-----+EP1|      |
    |    ---         |                   |      ---       |
    |    ---         |  VPN2             |      ---       |
----+   |EP2+--------+-------------------+-----+EP2|      +------
    |    ---  Site 1 |        ------     |      ---  Site2|
    +-- -------------+       /      \    +----------------+
                            | Internet |
                             \      /
                              ------
```

```
module: ietf-sdwan-svc
 +--rw sdwan-svc
   +--rw vpn-services
    | +--rw vpn-service* [vpn-id]
    |   +--rw vpn-id              svc-id
    |   +--rw topology?            identityref
...
    |   +--rw endpoints* [endpoint-id]
    |     +--rw endpoint-id          svc-id
    |     +--rw site-role?          identityref
    |     +--rw site-attachment
    |     | +--rw site-id?  -> /sdwan-svc/sites/site/site-id
    |     +--rw endpoint-policy-map
    |       +--rw app-group-policy* [app-group-id]
    |       | +--rw app-group-id    leafref
    |       | +--rw policy-id?      leafref
    |       +--rw app-policy* [app-id]
    |         +--rw app-id       leafref
    |         +--rw policy-id?   leafref
```

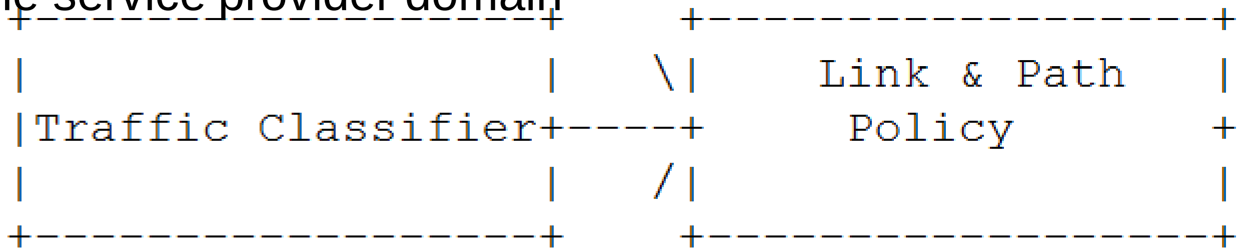


```
+---------------------+       --
|                     |     /     \
|  LAN1 +---------+WAN1|    / MPLS |
|  ------+        +----+--  / \ VPN/
|  ------+Device 1+----+-       ---
|  LAN2 +---------+WAN2|  \  /       \
|                          \Internet|
|   Site1             |     \       /
+---------------------+       -----
```

```
+--rw sites
   +--rw site* [site-id]
     +--rw site-id      svc-id
     +--rw device* [name]
     | +--rw name    string
     | +--rw type?   identityref
     +--rw lan-access* [name]
     | +--rw name            string
     | +--rw l2-technology
...
     +--rw wan-access* [name]
      +--rw name            string
      +--rw access-type?      identityref
      +--rw access-provider?  string
      +--rw bandwidth
      | +--rw input-bandwidth?   uint64
      | +--rw output-bandwidth?  uint64
...
```

SD-WAN application based policy service components

- The service orchestrator of a service provider can request, configure, and manage application policies
- The assumption is one service provider domain



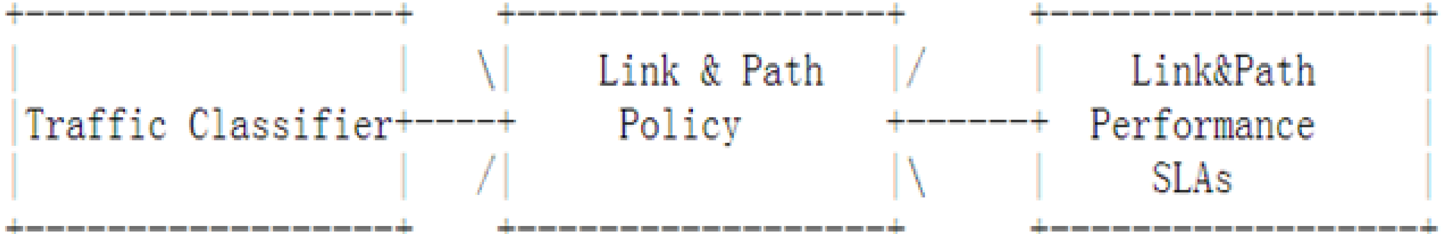
```
+--------------------+    +--------------------+
|                    |   \|     Link & Path    |
|Traffic Classifier+-----+      Policy         +
|                    |   /|                    |
+--------------------+    +--------------------+
```

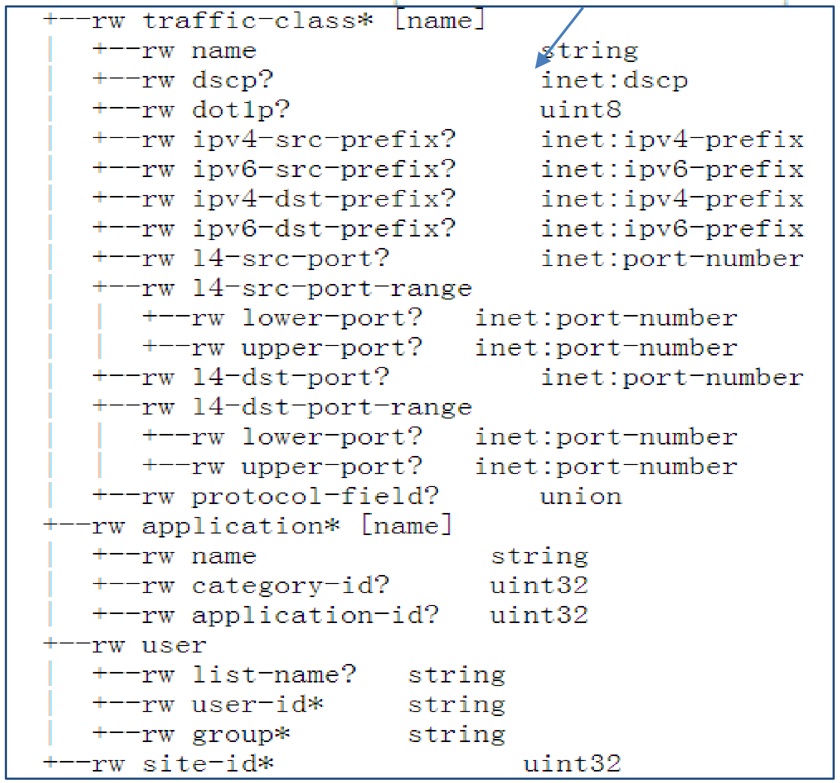
```
+--rw application* [app-id]
  +--rw app-id    svc-id
  +--rw ac* [name]  (application flow criteria)
     +--rw name                    string
     +--rw (match-type)?
        +--:(match-flow)
        |  +--rw match-flow
        |     +--rw ethertype?        uint16
        |     +--rw cvlan?            uint8
        |     +--rw ipv4-src-prefix?   inet:ipv4-prefix
        |     +--rw ipv4-dst-prefix?   inet:ipv4-prefix
        |     +--rw l4-src-port?       inet:port-number
        |     +--rw l4-dst-port?       inet:port-number
        |     +--rw ipv6-src-prefix?   inet:ipv6-prefix
        |     +--rw ipv6-dst-prefix?   inet:ipv6-prefix
        |     +--rw protocol-field?    union
        +--:(match-application)
           +--rw match-application?   identityref
```

```
+--rw policy* [policy-id]
  +--rw policy-id        svc-id
  +--rw policy-package
     +--rw encryption?       enumeration
     +--rw public-private?   enumeration
     +--rw local-breakout?   boolean
     +--rw billing-method?   enumeration
     +--rw backup-path?      enumeration
     +--rw bandwidth
        +--rw commit?   uint32
        +--rw max?      uint32
```

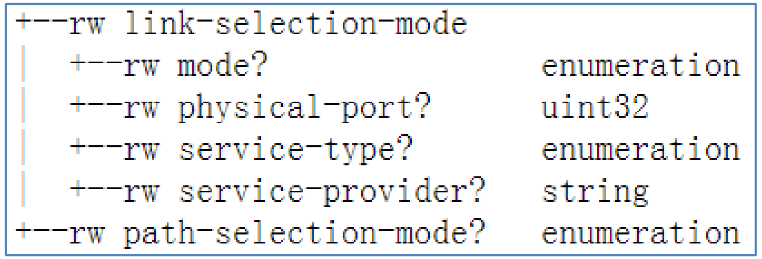
SD-WAN OSE path service

- OSE model needs to ensure consistent policy across domains





```
+--rw traffic-class* [name]
|  +--rw name                    string
|  +--rw dscp?                   inet:dscp
|  +--rw dot1p?                  uint8
|  +--rw ipv4-src-prefix?        inet:ipv4-prefix
|  +--rw ipv6-src-prefix?        inet:ipv6-prefix
|  +--rw ipv4-dst-prefix?        inet:ipv4-prefix
|  +--rw ipv6-dst-prefix?        inet:ipv6-prefix
|  +--rw l4-src-port?            inet:port-number
|  +--rw l4-src-port-range
|  |  +--rw lower-port?   inet:port-number
|  |  +--rw upper-port?   inet:port-number
|  +--rw l4-dst-port?            inet:port-number
|  +--rw l4-dst-port-range
|  |  +--rw lower-port?   inet:port-number
|  |  +--rw upper-port?   inet:port-number
|  +--rw protocol-field?         union
+--rw application* [name]
|  +--rw name              string
|  +--rw category-id?       uint32
|  +--rw application-id?    uint32
+--rw user
|  +--rw list-name?    string
|  +--rw user-id*      string
|  +--rw group*        string
+--rw site-id*                  uint32
```

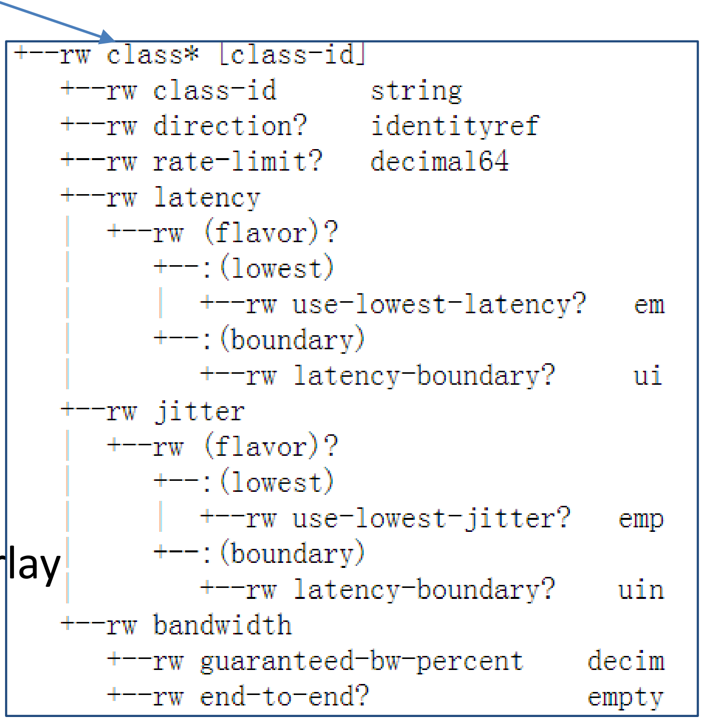


```
+--rw link-selection-mode
|  +--rw mode?                 enumeration
|  +--rw physical-port?        uint32
|  +--rw service-type?         enumeration
|  +--rw service-provider?     string
+--rw path-selection-mode?     enumeration
```

mode: automatic, preferred, lowest-cost

service-type: commodity, wireless, private

path-selection-mode: drop, underlay, overlay



```
+--rw class* [class-id]
   +--rw class-id       string
   +--rw direction?     identityref
   +--rw rate-limit?    decimal64
   +--rw latency
   |  +--rw (flavor)?
   |     +--:(lowest)
   |     |  +--rw use-lowest-latency?    em
   |     +--:(boundary)
   |        +--rw latency-boundary?      ui
   +--rw jitter
   |  +--rw (flavor)?
   |     +--:(lowest)
   |     |  +--rw use-lowest-jitter?    emp
   |     +--:(boundary)
   |        +--rw latency-boundary?     uin
   +--rw bandwidth
      +--rw guaranteed-bw-percent    decim
      +--rw end-to-end?              empty
```