A YANG Data Model for Service Path Computation

CCAMP WG, IETF120

draft-ybb-ccamp-service-path-computation-00

Author:

Chaode Yu (Huawei)

Sergio Belotti (Nokia)

Italo Busi (Huawei)

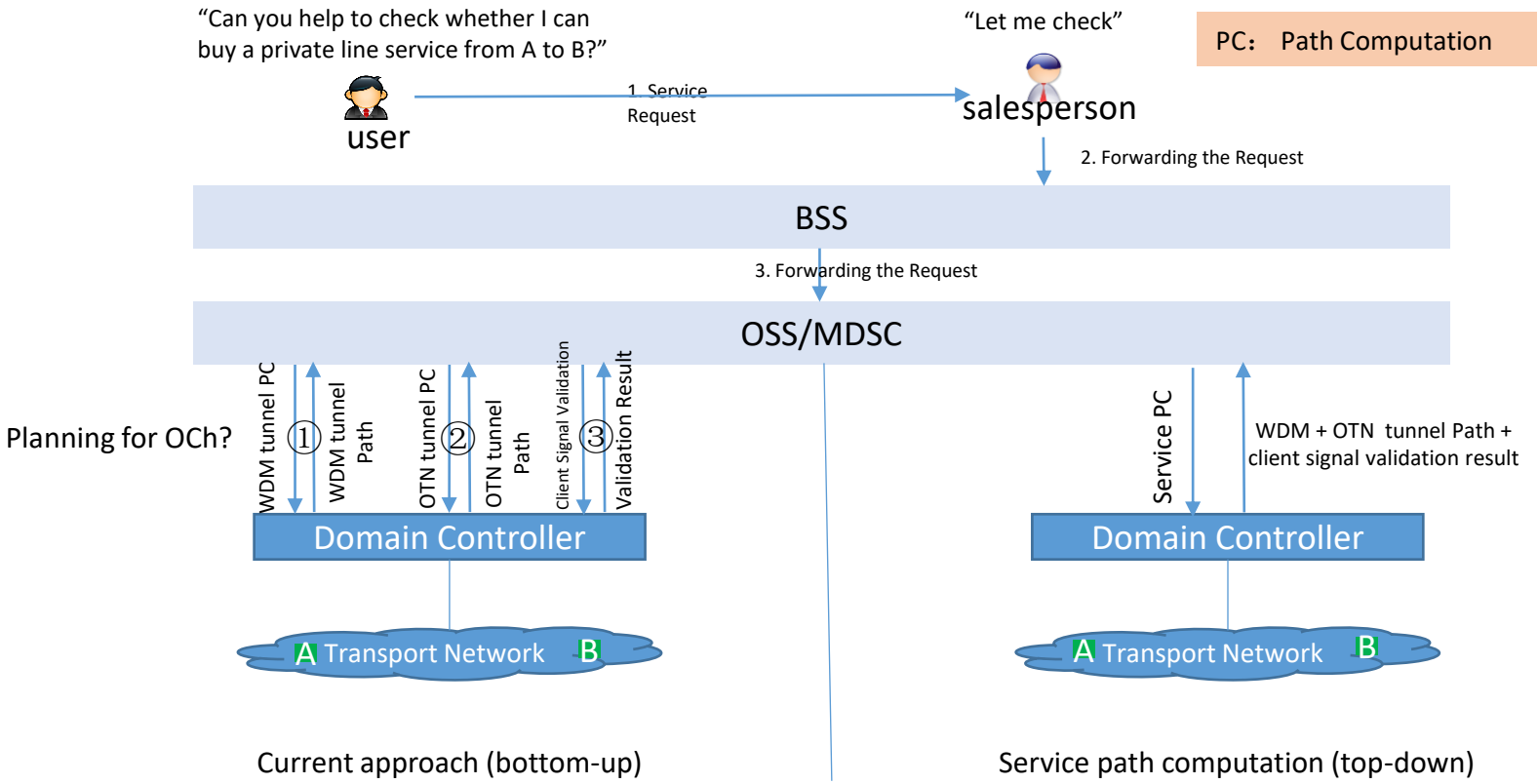
Aihua Guo (Futurewei)

Dieter Beller (Nokia)

Motivations of this Draft

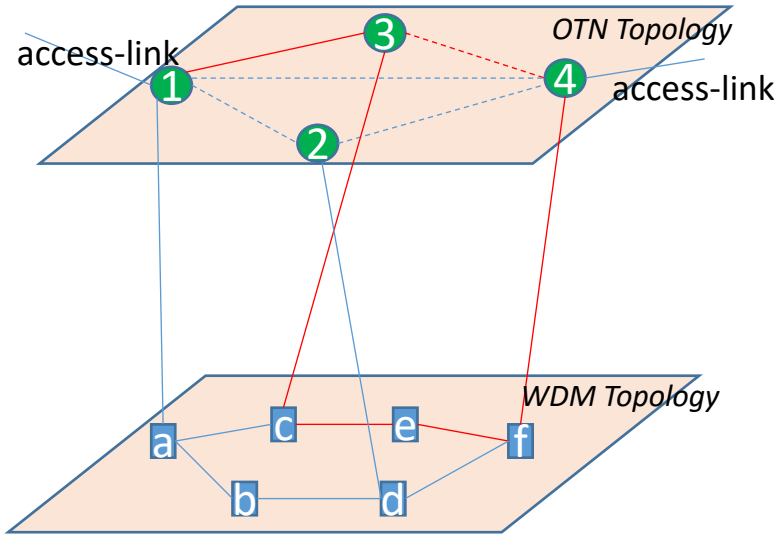
- **Service Driven (top-down) Approach:** User and the engineers of Operators only need to provide the source and destination of service and some intention preference (SLA?), domain controller can help to compute all the underlay paths' computation, and provide comprehensive optimal result;
- **Multi-layers' Path Display:** Provide a generic & nested structure to describe all the layers' information in the response;
- **Path Management:** Provide a data structure to store the path computation result on the purpose to reference it when provisioning.
- **More kinds of path constrain:** Includes specifying cross-layer resource, finer granularity (TS/wavelength) objects, include/exclude service/tunnel/path/SRLG;

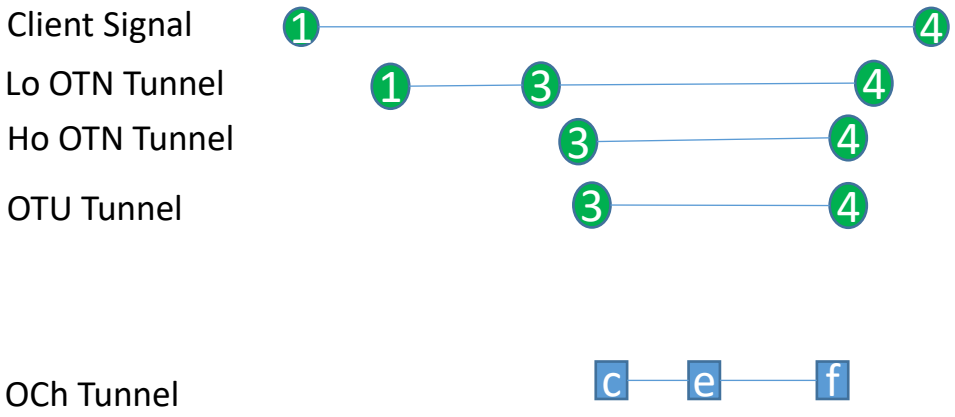
What is a Top-down Approach?



- For bottom-up approach, the OSS/MDSC needs to interact with domain controller multiple times which may increase TTM (Time To Market);
- Some operators would like to delegate to controller configuration of service. And this option is not supported in the current model.
- The current approach requires configuration of the tunnel before the service.

Multi-Layers Path Display






```
rpcs:
  +---x client-service-precompute
    +--ro input
    +--ro output
      +--ro result* [request-id]
        +--ro (result-detail)?
          +--:(success)
            +--ro computed-paths* [path-id]
            |  +--ro path-id       yang:uuid
            |  +--ro path-number?   uint8
              +--ro src-access-ports
              +--ro dst-access-ports
              +--ro underlay-tunnels
                +--ro underlay-tunnel* [index]
                  +--ro underlay-tunnel* [index]
                    +--ro index                 uint8
                    +--ro tunnel-name?          leafref
                    +--ro te-topology-identifier
                    +--ro computed-lsp* [lsp-id]
                      +--ro lsp-route-objects* [index]
                        +--ro server-tunnel?   leafref
```

- The tunnels underlay of the client signal service will ~~all~~ be presented in the underlay-tunnels structure and will be ordered hierarchically, following their supporting relationship.
- The detailed path information will be described by the computed-lsp structure.
- If the hierarchical link is supported by its server layer, then the server-tunnel identifier referenced under the lsp-route-object can be used to indicate the supporting relationship. By this nested structure, we can have all the layers' path information.

How to Do the Path Management?

```
rpcs:
  +---x client-service-precompute
    +--ro input
    +--ro output
      +--ro result* [request-id]
        +--ro (result-detail)?
          +--:(success)
            +--ro computed-paths* [path-id]
              |  +--ro path-id        yang:uuid
              |  +--ro path-number?   uint8
                 +--ro src-access-ports
                 +--ro dst-access-ports
                 +--ro underlay-tunnels
                   +--ro underlay-tunnel* [index]
                      +--ro index                uint8
                      +--ro tunnel-name?         leafref
                      +--ro te-topology-identifier
                      +--ro computed-lsp* [lsp-id]
                        +--ro lsp-route-objects* [index]
                          +--ro server-tunnel?   leafref
```

```
module: ietf-trans-client-service
  +--rw client-svc
  +--rw path-management
    +--rw path* [path-id]
       +--rw path-id          yang:uuid
       +--rw creation-time?   yang:date-and-time
       +--rw validity-period? uint8
       +--ro src-access-ports
       +--ro dst-access-ports
       +--ro underlay-tunnels
         +--ro underlay-tunnel* [index]
            +--ro index                uint8
            +--ro tunnel-name?         leafref
            +--ro te-topology-identifier
            +--ro computed-lsp* [lsp-id]
            +--ro lsp-route-objects* [index]
            +--ro server-tunnel?   leafref
```

```
module: ietf-trans-client-service
  +--rw client-svc
  |  +--rw client-svc-instances* [client-svc-name]
  |     +--rw client-svc-name          string
  |     +--rw src-access-ports
  |     +--rw dst-access-ports
  |     +--rw (creation-manner)?
  |     |  +--:(bottom-up)
  |     |  |  +--rw svc-tunnels* [tunnel-name]
  |     |  |     +--rw tunnel-name    string
  |     |  +--:(top-down)
  |     |     +--rw path-id?              leafref
```

- The service path computation result will be saved on the domain controller and be kept for a while. (How long the computation result can be saved depends on the server's implementation.)
- If the path is adopted, e.g. it's referenced when service provisioning or its referenced in path constrain, it cannot be deleted.
- When the HCO/OSS want to reference the path existing on the server, it can reference the path-id directly.

Path Constraint

```
rpcs:
  +---x client-service-precompute
    +--ro input
    |  +--ro request* [request-id]
    |     +--ro request-id                    string
    |     +--ro explicit-route-exclude-objects
    |     |  +--ro route-object-exclude-object* [index]
    |     |     +--ro index              uint8
    |     |     +--ro node-id?           te-types:te-node-id
    |     |     +--ro node-uri-id?       yang:uuid
    |     |     +--ro link-tp-id?        te-types:te-tp-id
    |     |     +--ro ltp-uri-id?        yang:uuid
    |     |     +--ro te-label
    |     |     |  +--ro (technology)?
    |     |     |  |  +--:(wson)
    |     |     |  |  |  +--ro (grid-type)?
    |     |     |  |  |     +--:(dwdm)
    |     |     |  |  |     |  +--ro (single-or-super-channel)?
    |     |     |  |  |     |     +--:(single)
    |     |     |  |  |     |     |  +--ro dwdm-n?          l0-types:dwdm-n
    |     |     |  |  |     |     +--:(super)
    |     |     |  |  |     |        +--ro subcarrier-dwdm-n*   l0-types:dwdm-n
    |     |     |  |  |     +--:(cwdm)
    |     |     |  |  |        +--ro cwdm-n?          l0-types:cwdm-n
    |     |     |  |  +--:(otn)
    |     |     |  |     +--ro otn
    |     |     |  |        +--ro tpn?      otn-tpn
    |     |     |  |        +--ro tsg?      identityref
    |     |     |  |        +--ro ts-list?  string
    |     |     |  +--ro direction?         te-types:te-label-direction
    |     |     +--ro server-tunnel-name?   leafref
    |     |     +--ro lsp-type?             enumeration
    |     +--ro explicit-route-include-objects
```

It is supported to specify:

- Cross-layer's resource: resource on L0.
- Finer granularity resource: TPN/TS, wavelength
- Disjointness: service/tunnel/path/SRLG;

Next Step

- Fix the remained issues
- Discuss the modeling of Ethernet service path computation

- Call for interest & joint contribution
  - Github: https://github.com/YuChaode/draft-ybb-ccamp-service-path-computation
  - Bi-Weekly Call: https://ietf.webex.com/ietf/j.php?MTID=m524f3ae9ab0337f2e50c6e1ae06551e1 (UTC 3pm-4pm)