A YANG Data Model for Transport Network Client Signals

CCAMP WG, IETF118

draft-ietf-ccamp-client-signal-yang-10

Author:

Haomian Zheng (Huawei)

Aihua Guo (Futurewei)

Italo Busi (Huawei)

Anton Snitser (Cisco)

Francesco Lazzeri (Erricson)

Chaode Yu (Huawei)

Contributors:

Yunbin Xu (CAICT)

Yang Zhao (China Mobile)

Xufeng Liu (Alef Edge)

Giuseppe Fioccola (Huawei)

Yanlei Zheng (China Unicom)

Zhe Liu (Huawei)

Sergio Belotti (Nokia)

Yingxi Yao (Shanghai Bell)

Henry Yu (Huawei)

Major Updates Since IETF 117

```
module: ietf-trans-client-service
  +--rw client-svc
    +--rw client-svc-instances* [client-svc-name]
      +--rw client-svc-name         string
      +--rw client-svc-title?       string
      +--rw user-label?             string                  ①
      +--rw client-svc-descr?       string
      +--rw client-svc-customer?    string
      +--rw resilience
     +--rw src-access-ports
      |  +--rw access-node-id?   te-types:te-node-id
      |  +--rw access-node-uri?   nw:node-id
      |  +--rw access-ltp-id?     te-types:te-tp-id             ②
      |  +--rw access-ltp-uri?    nt:tp-id
      |  +--rw client-signal?     identityref
      +--rw dst-access-ports
      |......................................................................
      +--ro pm-state
      |  +--ro latency?   uint32                                ③
      +--ro error-info
      |  +--ro error-code?          uint16
      |  +--ro error-description?   string                      ④
      |  +--ro error-timestamp?     yang:date-and-time
      +--rw alarm-threshold
      |  +--rw latency-threshold?   Uint32                      ⑤
      |......................................................................
```

① **user-label**: an optional alias of client signal service

② **access-node-uri & access-ltp-uri**: align with the identifiers defined in network and network topology model;

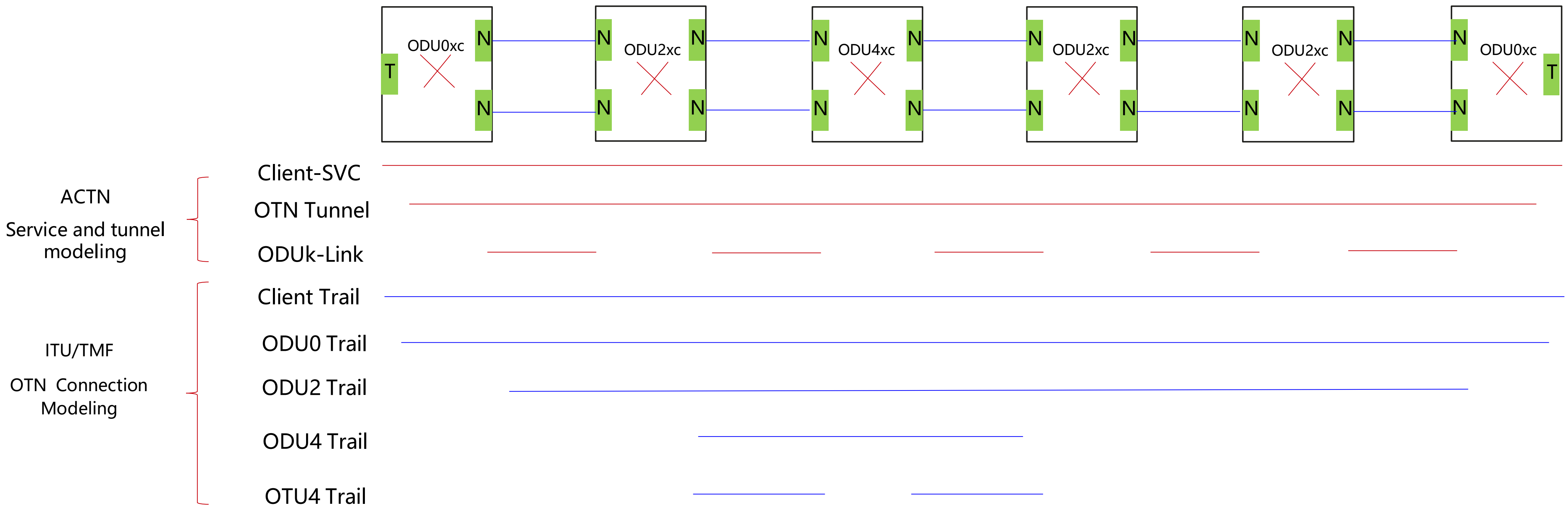
③ **pm-state**: provides a service-level performance overview;

④ **error-info**: provides mode detail of error information which is more technology and vendor specific, besides RESTCONF protocol error.

⑤ **alarm-threshold**: TCA for service assurance

These updates are also applicable to ETH service model.

Open Discussion1— Architecture of Client Signal Service and Underlay Tunnel

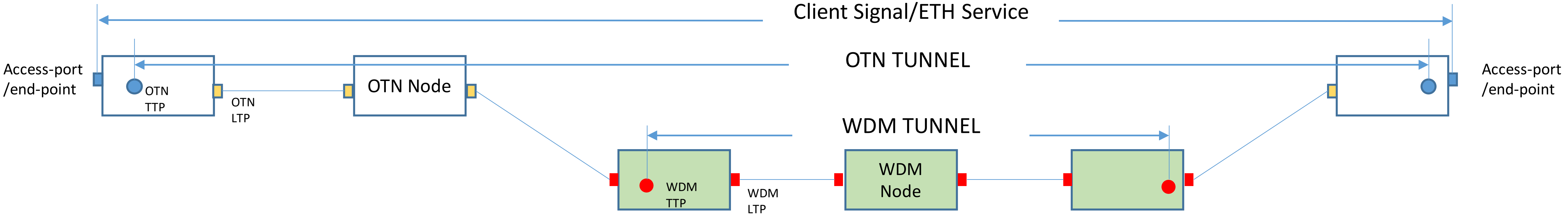


We recognized the missing definition of current draft:

- It is not clear whether the OTN tunnel in ACTN architecture includes all the ODUk and OTUk trails of ITU-T/TMF modeling;
- It is not clear whether the Client Signal service in ACTN can be mapped to the client trail object or not. Do we need to define a new client tunnel?
- It is not clear how to use the current modules to create an E2E client signal service, since a lot of attributes are defined in tunnel model rather than client signal model, e.g. protection & restoration, path computation policy and path constrain etc..
- It is not clear how to model crossing connection with current data modules.

Open Discussion2— Service Path Computation

1. Do we need a service path computation RPC?



- ➢ Currently we can support path computation on OTN &WDM tunnel layer, but OTN tunnel is not the ultimate object delivered to the user, while Client Signal/ETH service is.
- ➢ The current tunnel path computation modules can only support to be operated on a single tunnel, and there are requirements to specify different layers of resource in computation requests and get an E2E path computation result including both OTN&WDM tunnel.

2. How to quickly use the path computation result in service configuration?

An example is referenced from TAPI connectivity model：

```
module: tapi-connectivity
augment /tapi-common:context:
   +--rw connectivity-context
      +--rw connectivity-service* [uuid]
      | ………………………………………………………………
      | +--rw topology-constraint* [local-id]
      | | +--rw include-path* [path-uuid]
      | | | +--rw path-uuid   -> /tapi-common:context/tapi-path-computation:path-computation-context/path/uuid
      | | +--rw exclude-path* [path-uuid]
      | | | +--rw path-uuid   -> /tapi-common:context/tapi-path-computation:path-computation-context/path/uuid
```

- ➢ It is convenient to specify the path id to utilize the path computation result.
- ➢ Include/exclude a or several paths is a functionality tunnel path computation not support.
- ➢ Include/exclude paths can be extended to use as include/exclude services which is a common UC.
- ➢ There could be some other advanced functions supported if the path object is not limited to generate by path computation .

Next Step

- Try to figure out some solutions for the issues posted in the open discussion;
- Fixed the existing issues;
- Cooperate with the DT and design more UCs and find more gaps;

- Call for interest & joint contribution
  - Github: https://github.com/ietf-ccamp-wg/draft-ietf-ccamp-client-signal-yang
  - Weekly Call: coming soon….

Thank You ！