

CCAMP Working Group
Internet Draft
Intended status: Informational

I. Busi
Huawei
D. King
Lancaster University
H. Zheng
Huawei
Y. Xu
CAICT

Expires: January 2019

July 2, 2018

Transport Northbound Interface Applicability Statement
draft-ietf-ccamp-transport-nbi-app-statement-02

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

This Internet-Draft will expire on January 2, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the [Trust Legal Provisions](#) and are provided without warranty as described in the Simplified BSD License.

Abstract

Transport network domains, including Optical Transport Network (OTN) and Wavelength Division Multiplexing (WDM) networks, are typically deployed based on a single vendor or technology platforms. They are often managed using proprietary interfaces to dedicated Element Management Systems (EMS), Network Management Systems (NMS) and increasingly Software Defined Network (SDN) controllers.

A well-defined open interface to each domain management system or controller is required for network operators to facilitate control automation and orchestrate end-to-end services across multi-domain networks. These functions may be enabled using standardized data models (e.g. YANG), and appropriate protocol (e.g., RESTCONF).

This document analyses the applicability of the YANG models being defined by IETF (TEAS and CCAMP WGs in particular) to support OTN single and multi-domain scenarios.

Table of Contents

1.	Introduction.....	4
1.1.	Scope of this document.....	4
1.2.	Assumptions.....	5
2.	Terminology.....	6
3.	Conventions used in this document.....	6
3.1.	Topology and traffic flow processing.....	6
3.2.	JSON code.....	7
4.	Scenarios Description.....	8
4.1.	Reference Network.....	8
4.1.1.	Single-Domain Scenario.....	11
4.1.2.	Multi-Domain Scenario.....	11
4.2.	Topology Abstractions.....	11
4.3.	Service Configuration.....	13
4.3.1.	ODU Transit.....	14

4.3.2.	EPL over ODU.....	15
4.3.3.	Other OTN Clients Services.....	16
4.3.4.	EVPL over ODU.....	17
4.3.5.	EVPLAN and EVPTree Services.....	18
4.3.6.	Dynamic Service Configuration.....	19
4.4.	Multi-function Access Links.....	20
4.5.	Protection and Restoration Configuration.....	21
4.5.1.	Linear Protection (end-to-end).....	21
4.5.2.	Segmented Protection.....	23
4.5.3.	End-to-End Dynamic restoration.....	23
4.5.4.	Segmented Dynamic Restoration.....	24
4.6.	Service Modification and Deletion.....	24
4.7.	Notification.....	25
4.8.	Path Computation with Constraint.....	25
5.	YANG Model Analysis.....	26
5.1.	YANG Models for Topology Abstraction.....	26
5.1.1.	Domain 1 Topology Abstraction.....	27
5.1.2.	Domain 2 Grey (Type A) Topology Abstraction.....	28
5.1.3.	Domain 3 Grey (Type B) Topology Abstraction.....	28
5.1.4.	Multi-domain Topology Stitching.....	28
5.1.5.	Access Links.....	29
5.2.	YANG Models for Service Configuration.....	31
5.2.1.	ODU Transit Service.....	33
5.2.1.1.	Single Domain Example.....	35
5.2.2.	EPL over ODU Service.....	36
5.2.3.	Other OTN Client Services.....	38
5.2.4.	EVPL over ODU Service.....	38
5.3.	YANG Models for Protection Configuration.....	39
5.3.1.	Linear Protection (end-to-end).....	39
5.3.2.	Segmented Protection.....	39
6.	Security Considerations.....	39
7.	IANA Considerations.....	39
8.	References.....	39
8.1.	Normative References.....	39
8.2.	Informative References.....	41
9.	Acknowledgments.....	41
	Appendix A Validating a JSON fragment against a YANG Model....	43
A.1.	Manipulation of JSON fragments.....	43
A.2.	Comments in JSON fragments.....	44
A.3.	Validation of JSON fragments: DSDL-based approach.....	44
A.4.	Validation of JSON fragments: why not using a XSD-based approach.....	45
	Appendix B Detailed JSON Examples.....	46
B.1.	JSON Examples for Topology Abstractions.....	46
B.1.1.	JSON Code: mpi1-otn-topology.json.....	46

B.2. JSON Examples for Service Configuration.....	73
B.2.1. JSON Code: mpi1-odu2-service-config.json.....	73
B.2.2. JSON Code: mpi1-odu2-tunnel-config.json.....	77
B.2.3. JSON Code: mpi1-epl-service-config.json.....	80

[1. Introduction](#)

Transport of packet services are critical for a wide-range of applications and services, including: data center and LAN interconnects, Internet service backhauling, mobile backhaul and enterprise Carrier Ethernet Services. These services are typically setup using stovepipe NMS and EMS platforms, often requiring propriety management platforms and legacy management interfaces. A clear goal of operators will be to automate setup of transport services across multiple transport technology domains.

A common open interface (API) to each domain controller and or management system is pre-requisite for network operators to control multi-vendor and multi-domain networks and enable also service provisioning coordination/automation. This can be achieved by using standardized YANG models, used together with an appropriate protocol (e.g., [RESTCONF]).

This document analyses the applicability of the YANG models being defined by IETF (TEAS and CCAMP WGs in particular) to support OTN single and multi-domain scenarios.

[1.1. Scope of this document](#)

This document assumes a reference architecture, including interfaces, based on the Abstraction and Control of Traffic-Engineered Networks (ACTN), defined in [ACTN-Frame].

The focus of this document is on the MPI (interface between the Multi Domain Service Coordinator (MDSC) and a Physical Network Controller (PNC), controlling a transport network domain).

It is worth noting that the same MPI analyzed in this document could be used between hierarchical MDSC controllers, as shown in Figure 4 of [ACTN-Frame].

Detailed analysis of the CMI (interface between the Customer Network Controller (CNC) and the MDSC) as well as of the interface between service and network orchestrators are outside the scope of this

document. However, some considerations and assumptions about the information could be described when needed.

The relationship between the current IETF YANG models and the type of ACTN interfaces can be found in [\[ACTN-YANG\]](#). Therefore, it considers the TE Topology YANG model defined in [\[TE-TOPO\]](#), with the OTN Topology augmentation defined in [\[OTN-TOPO\]](#) and the TE Tunnel YANG model defined in [\[TE-TUNNEL\]](#), with the OTN Tunnel augmentation defined in [\[OTN-TUNNEL\]](#).

The analysis of how to use the attributes in the I2RS Topology YANG model, defined in [\[I2RS-TOPO\]](#), is for further study.

The ONF Technical Recommendations for Functional Requirements for the transport API in [\[ONF TR-527\]](#) and the ONF transport API multi-domain examples in [\[ONF GitHub\]](#) have been considered as an input for defining the reference scenarios analyzed in this document.

[1.2. Assumptions](#)

This document is making the following assumptions, still to be validated with TEAS WG:

1. The MDSC can request, at the MPI, a PNC to setup a Transit Tunnel Segment using the TE Tunnel YANG model: in this case, since the endpoints of the E2E Tunnel are outside the domain controlled by that PNC, the MDSC would not specify any source or destination TTP (i.e., it would leave the source, destination, src-tp-id and dst-tp-id attributes empty) for the tunnel and it would use the explicit-route-object/route-object-include-exclude list to specify the ingress and egress links for each path of the Transit Tunnel Segment.
2. Each PNC provides to the MDSC, at the MPI, the list of available timeslots on the inter-domain links using the TE Topology YANG model and OTN Topology augmentation. The TE Topology YANG model in [\[TE-TOPO\]](#) is being updated to report the label set information.

This document is also making the following assumptions, still to be validated with CCAMP WG:

1. The topology information for the Ethernet access links are modelled using the YANG model defined in [\[Client-Topo\]](#).

2. The service information for Ethernet and other OTN client layer services are modelled using the YANG model defined in [Client-Signal].

2. Terminology

Domain: defined as a collection of network elements within a common realm of address space or path computation responsibility [[RFC5151](#)]

E-LINE: Ethernet Line

EPL: Ethernet Private Line

EVPL: Ethernet Virtual Private Line

OTN: Optical Transport Network

Service: A service in the context of this document can be considered as some form of connectivity between customer sites across the network operator's network [[RFC8309](#)]

Service Model: As described in [[RFC8309](#)] it describes a service and the parameters of the service in a portable way that can be used uniformly and independent of the equipment and operating environment.

UNI: User Network Interface

MDSC: Multi-Domain Service Coordinator

CNC: Customer Network Controller

PNC: Provisioning Network Controller

MAC Bridging: Virtual LANs (VLANs) on IEEE 802.3 Ethernet network

3. Conventions used in this document

3.1. Topology and traffic flow processing

The traffic flow between different nodes is specified as an ordered list of nodes, separated with commas, indicating within the brackets the processing within each node:

<node> (<processing>){, <node> (<processing>)}

The order represents the order of traffic flow being forwarded through the network.

The processing can be either an adaptation of a client layer into a server layer "(client -> server)" or switching at a given layer "([switching])". Multi-layer switching is indicated by two layer switching with client/server adaptation: "([client] -> [server])".

For example, the following traffic flow:

```
R1 ([PKT] -> ODU2), S3 ([ODU2]), S5 ([ODU2]), S6 ([ODU2]),  
R3 (ODU2 -> [PKT])
```

Node R1 is switching at the packet (PKT) layer and mapping packets into an ODU2 before transmission to node S3. Nodes S3, S5 and S6 are switching at the ODU2 layer: S3 sends the ODU2 traffic to S5 which then sends it to S6 which finally sends to R3. Node R3 terminates the ODU2 from S6 before switching at the packet (PKT) layer.

The paths of working and protection transport entities are specified as an ordered list of nodes, separated with commas:

```
<node> {, <node>}
```

The order represents the order of traffic flow being forwarded through the network in the forward direction. In case of bidirectional paths, the forward and backward directions are selected arbitrarily, but the convention is consistent between working/protection path pairs as well as across multiple domains.

3.2. JSON code

This document provides some detailed JSON code examples to describe how the YANG models being developed by IETF (TEAS and CCAMP WG in particular) can be used.

The examples are provided using JSON because JSON code is easier for humans to read and write.

Different objects need to have an identifier. The convention used to create mnemonic identifiers is to use the object name (e.g., S3 for node S3), followed by its type (e.g., NODE), separated by an "-", followed by "-ID". For example, the mnemonic identifier for node S3 would be S3-NODE-ID.

JSON language does not support the insertion of comments that have been instead found to be useful when writing the examples. This document inserts comments into the JSON code as JSON name/value pair with the JSON name string starting with the "://" characters. For example, when describing the example of a TE Topology instance representing the ODU Abstract Topology exposed by the Transport PNC, the following comment has been added to the JSON code:

```
"// comment": "ODU Abstract Topology @ MPI",
```

The JSON code examples provided in this document have been validated against the YANG models following the validation process described in [Appendix A](#), which would not consider the comments.

In order to have successful validation of the examples, some numbering scheme has been defined to assign identifiers to the different entities which would pass the syntax checks. In that case, to simplify the reading, another JSON name/value pair, formatted as a comment and using the mnemonic identifiers is also provided. For example, the identifier of node S3 (S3-NODE-ID) has been assumed to be "10.0.0.3" and would be shown in the JSON code example using the two JSON name/value pair:

```
"// te-node-id": "S3-NODE-ID",
```

```
"te-node-id": "10.0.0.3",
```

The first JSON name/value pair will be automatically removed in the first step of the validation process while the second JSON name/value pair will be validate against the YANG model definitions.

4. Scenarios Description

4.1. Reference Network

The physical topology of the reference network is shown in Figure 1. It represents an OTN network composed of three transport network domains providing transport services to an IP customer network through eight access links:

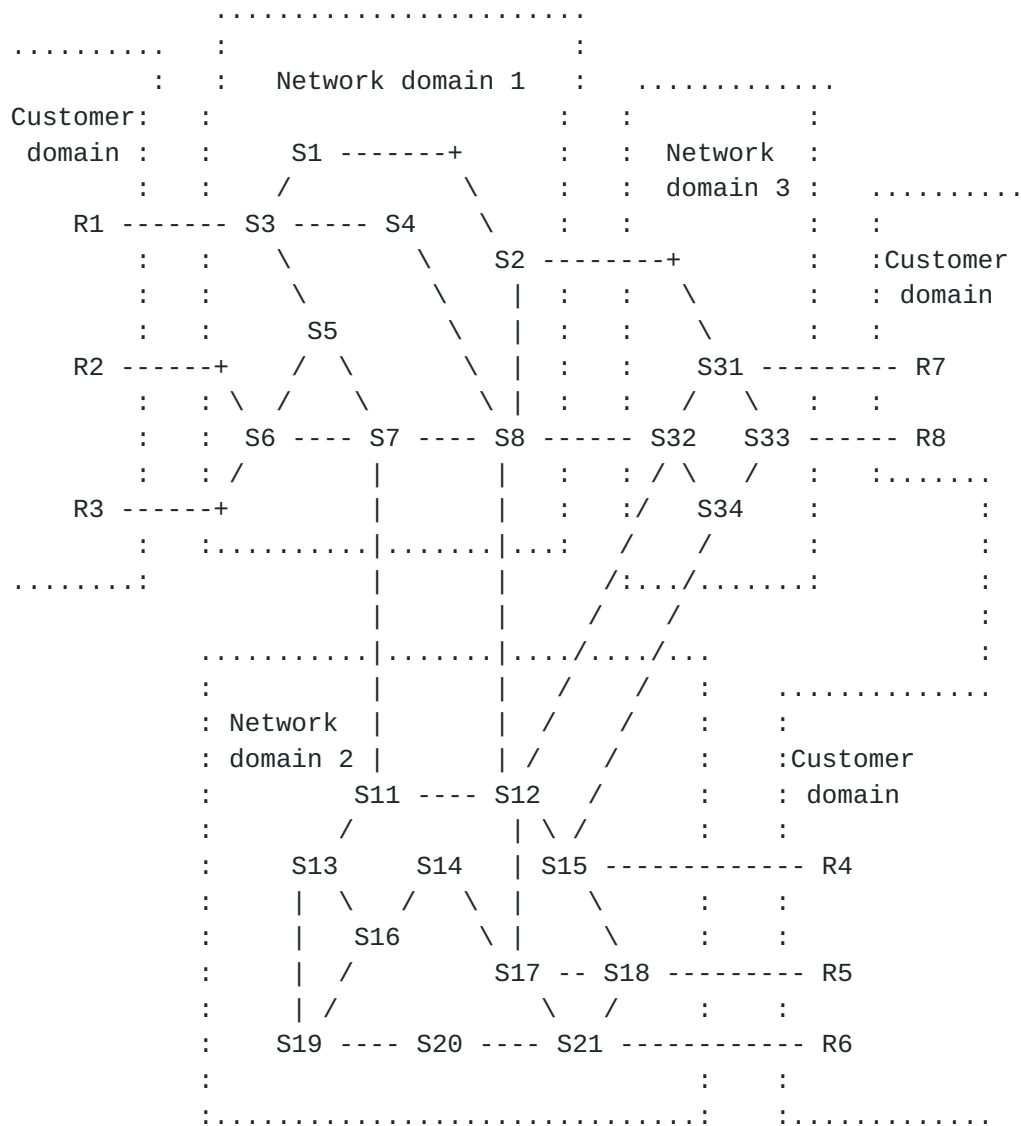


Figure 1 Reference network

This document assumes that all the transport network switching nodes S_i are OTN switching nodes capable to switch only in the electrical domain (ODU switching only) and that all the S_i - S_j OTN links within the transport network (intra-domain or inter-domain) are 100G links while the access R_i - S_j links are 10G links. Different technologies can be used at the access links (e.g., Ethernet, STM-n, OTN).

It is also assumed that, within the transport network, the physical/optical interconnections supporting the S_i - S_j OTN links (up to the OTU4 trail), are pre-configured using mechanisms which are

outside the scope of this document and are not exposed at the MPIs to the MDSC.

The transport domain control architecture, shown in Figure 2, follows the ACTN architecture and framework document [[ACTN-Frame](#)], and functional components:

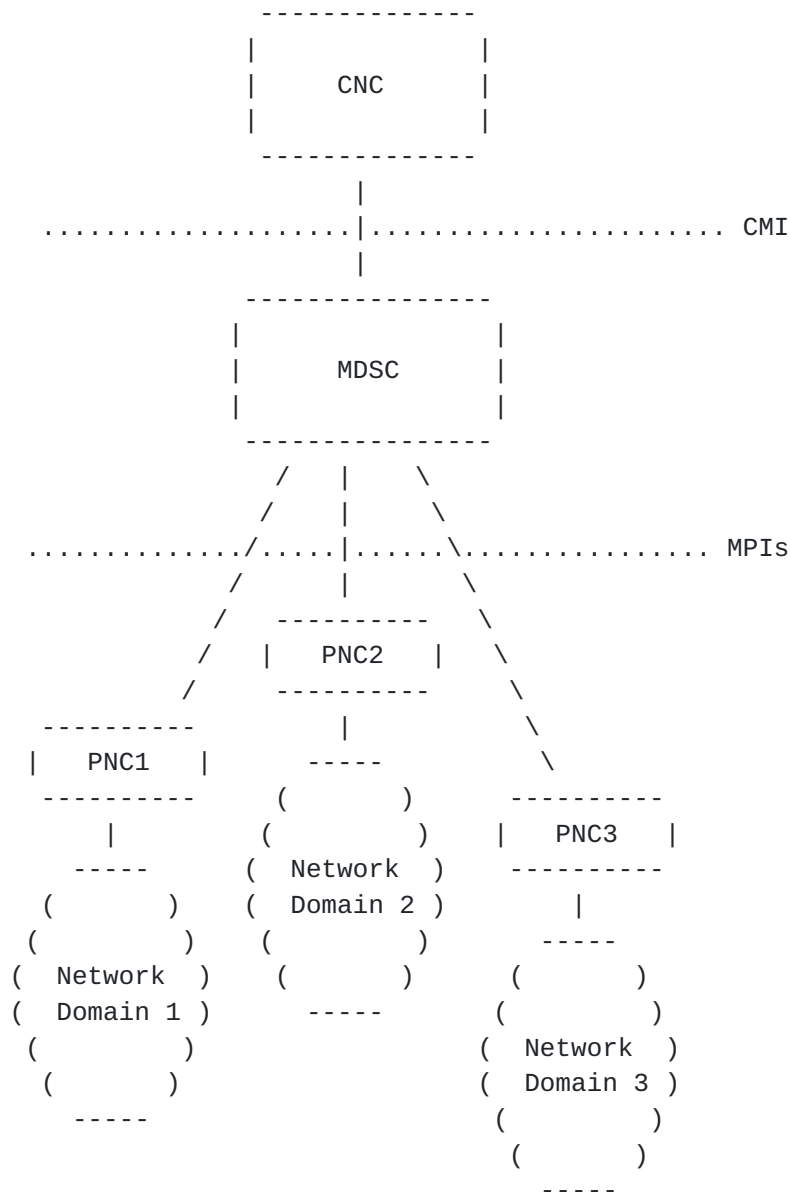


Figure 2 Controlling Hierarchy

The ACTN framework facilitates the detachment of the network and service control from the underlying technology and help the customer express the network as desired by business needs. Therefore, care must be taken to keep minimal dependency on the CMI (or no dependency at all) with respect to the network domain technologies. The MPI instead requires some specialization according to the domain technology.

This document assumes that the CNC controls the customer IP network and requests, at the CMI, transport connectivity between IP routers. The MDSC coordinates, via three MPIs, the control of a multi-domain transport network through three PNCs.

The control interfaces within scope of this document are the three MPIs, while the control interface(s) between the CNC and the IP routers is outside the scope of this document. It is also assumed that the CMI allows the CNC to provide all the information that is required by the MDSC to properly configure the transport connectivity requested by the customer.

4.1.1. Single-Domain Scenario

In case the CNC requests transport connectivity between IP routers attached to the same transport domain (e.g., between R1 and R3 in Figure 1), the MDSC can just pass the service request to the PNC controlling that domain (e.g., PNC1 in Figure 2) and let the PNC take decisions about how to implement the service (e.g., setting up the intra-domain end-to-end OTN connection).

4.1.2. Multi-Domain Scenario

In case the CNC requests transport connectivity between IP routers attached to different transport domains (e.g., between R1 and R5), the MDSC needs to coordinate the setup of a multi-domain end-to-end OTN connection across multiple PNCs (e.g., PNC1, PNC2 and PNC3 in Figure 2) as well as to coordinate the configuration of the service with the PNCs controlling the edge domains (e.g., PNC1 and PNC2 in Figure 2).

4.2. Topology Abstractions

Abstraction provides a selective method for representing connectivity information within a domain. There are multiple methods to abstract a network topology. This document assumes the abstraction method defined in [[RFC7926](#)]:

"Abstraction is the process of applying policy to the available TE information within a domain, to produce selective information that represents the potential ability to connect across the domain. Thus, abstraction does not necessarily offer all possible connectivity options, but presents a general view of potential connectivity according to the policies that determine how the domain's administrator wants to allow the domain resources to be used."

[ACTN-Frame] Provides the context of topology abstraction in the ACTN architecture and discusses a few alternatives for the abstraction methods for both packet and optical networks. This is an important consideration since the choice of the abstraction method impacts protocol design and the information it carries. According to [\[ACTN-Frame\]](#), there are three types of topology:

- o White topology: This is a case where the PNC provides the actual network topology to the MDSC without any hiding or filtering. In this case, the MDSC has the full knowledge of the underlying network topology;
- o Black topology: The entire domain network is abstracted as a single virtual node with the access/egress links without disclosing any node internal connectivity information;
- o Grey topology: This abstraction level is between black topology and white topology from a granularity point of view. This is abstraction of TE tunnels for all pairs of border nodes. We may further differentiate from a perspective of how to abstract internal TE resources between the pairs of border nodes:
 - Grey topology type A: border nodes with a TE links between them in a full mesh fashion;
 - Grey topology type B: border nodes with some internal abstracted nodes and abstracted links.

Each PNC should provide the MDSC a topology abstraction of the domain's network topology.

Each PNC provides topology abstraction of its own domain topology independently from each other and therefore it is possible that different PNCs provide different types of topology abstractions.

The MPI operates on the abstract topology regardless on the type of abstraction provided by the PNC.

To analyze how the MPI operates on abstract topologies independently from the topology abstraction provided by each PNC and, therefore, that that different PNCs can provide different topology abstractions, it is assumed that:

- o PNC1 provides a topology abstraction which exposes at MPI1 an abstract node and an abstract link for each physical node and link within network domain 1
- o PNC2 provides a topology abstraction which exposes at MPI2 a single abstract node (representing the whole network domain) with abstract links representing only the inter-domain physical links
- o PNC3 provides a topology abstraction which exposes at MPI3 two abstract nodes (called AN31 and AN32). They abstract respectively nodes S31+S33 and nodes S32+S34. At MPI3, only the abstract nodes should be reported: the mapping between the abstract nodes (AN31 and AN32) and the physical nodes (S31, S32, S33 and S34) should be done internally by PNC3.

The MDSC should be capable to stitch together each abstracted topology to build its own view of the multi-domain network topology. The process may require suitable oversight, including administrative configuration and trust models, but this is out of scope for this document.

The MDSC can also provide topology abstraction of its own view of the multi-domain network topology at its CMIs depending on the customers' needs: it can provide different types of topology abstractions at different CMIs.

4.3. Service Configuration

In the following scenarios, it is assumed that the CNC is capable to request service connectivity from the MDSC to support IP routers connectivity.

The type of services could depend of the type of physical links (e.g. OTN link, ETH link or SDH link) between the routers and transport network.

The control of different adaptations inside IP routers, R_i (PKT \rightarrow foo) and R_j (foo \rightarrow PKT), are assumed to be performed by means that are not under the control of, and not visible to, the MDSC nor to the PNCs. Therefore, these mechanisms are outside the scope of this document.

It is just assumed that the CNC is capable to request the proper configuration of the different adaptation functions inside the customer's IP routers, by means which are outside the scope of this document.

4.3.1. ODU Transit

The physical links interconnecting the IP routers and the transport network can be OTN links. In this case, it is assumed that the physical/optical interconnections below the ODU layer (up to the OTU2 trail) are pre-configured using mechanisms which are outside the scope of this document and not exposed at the MPIs to the MDSC.

To setup a 10Gb IP link between $R1$ and $R5$, an ODU2 end-to-end data plane connection needs be created between $R1$ and $R5$, crossing transport nodes $S3$, $S1$, $S2$, $S31$, $S33$, $S34$, $S15$ and $S18$ which belong to different PNC domains.

The traffic flow between $R1$ and $R5$ can be summarized as:

```
R1 ([PKT]  $\rightarrow$  ODU2), S3 ([ODU2]), S1 ([ODU2]), S2 ([ODU2]),  
S31 ([ODU2]), S33 ([ODU2]), S34 ([ODU2]),  
S15 ([ODU2]), S18 ([ODU2]), R5 (ODU2  $\rightarrow$  [PKT])
```

It is assumed that the CNC requests, via the CMI, the setup of an ODU2 transit service, providing all the information that the MDSC needs to understand that it shall setup a multi-domain ODU2 segment connection between nodes $S3$ and $S18$.

In case the CNC needs the setup of a 10Gb IP link between $R1$ and $R3$ (single-domain service request), the traffic flow between $R1$ and $R3$ can be summarized as:

```
R1 ([PKT]  $\rightarrow$  ODU2), S3 ([ODU2]), S5 ([ODU2]), S6 ([ODU2]),  
R3 (ODU2  $\rightarrow$  [PKT])
```

Since the CNC is unaware of the transport network domains, it requests the setup of an ODU2 transit service in the same way as

before, regardless the fact the fact that this is a single-domain service.

It is assumed that the information provided at the CMI is sufficient for the MDSC to understand that this is a single-domain service request.

The MDSC can then just request PNC1 to setup a single-domain ODU2 data plane segment connection between nodes S3 and S6.

4.3.2. EPL over ODU

The physical links interconnecting the IP routers and the transport network can be Ethernet links. In this case, it is assumed that the Ethernet physical interconnections below the MAC layer (up to the OTU2 trail) are pre-configured using mechanisms which are outside the scope of this document and not exposed at the MPIs to the MDSC.

To setup a 10Gb IP link between R1 and R5, an EPL service needs to be created between R1 and R5, supported by an ODU2 end-to-end data plane connection between transport nodes S3 and S18, crossing transport nodes S1, S2, S31, S33, S34 and S15 which belong to different PNC domains.

The traffic flow between R1 and R5 can be summarized as:

```
R1 ([PKT] -> ETH), S3 (ETH -> [ODU2]), S1 ([ODU2]),  
S2 ([ODU2]), S31 ([ODU2]), S33 ([ODU2]), S34 ([ODU2]),  
S15 ([ODU2]), S18 ([ODU2] -> ETH), R5 (ETH -> [PKT])
```

It is assumed that the CNC requests, via the CMI, the setup of an EPL service, providing all the information that the MDSC needs to understand that it shall coordinate the three PNCs to setup a multi-domain ODU2 end-to-end connection between nodes S3 and S18 as well as the configuration of the adaptation functions inside nodes S3 and S18: S3 (ETH -> [ODU2]), S18 ([ODU2] -> ETH), S18 (ETH -> [ODU2]) and S3 ([ODU2] -> ETH).

In case the CNC needs the setup of a 10Gb IP link between R1 and R3 (single-domain service request), the traffic flow between R1 and R3 can be summarized as:

```
R1 ([PKT] -> ETH), S3 (ETH -> [ODU2]), S5 ([ODU2]),  
S6 ([ODU2] -> ETH), R3 (ETH-> [PKT])
```

As described in [section 4.3.1](#), the CNC requests the setup of an EPL service in the same way as before and the information provided at the CMI is sufficient for the MDSC to understand that this is a single-domain service request.

The MDSC can then just request PNC1 to setup a single-domain EPL service between nodes S3 and S6. PNC1 can take care of setting up the single-domain ODU2 end-to-end connection between nodes S3 and S6 as well as of configuring the adaptation functions on these edge nodes.

[4.3.3. Other OTN Clients Services](#)

[ITU-T G.709] defines mappings of different client layers into ODU. Most of them are used to provide Private Line services over an OTN transport network supporting a variety of types of physical access links (e.g., Ethernet, SDH STM-N, Fibre Channel, InfiniBand, etc.).

The physical links interconnecting the IP routers and the transport network can be any of these types.

In order to setup a 10Gb IP link between R1 and R5 using, for example SDH physical links between the IP routers and the transport network, an STM-64 Private Line service needs to be created between R1 and R5, supported by ODU2 end-to-end data plane connection between transport nodes S3 and S18, crossing transport nodes S1, S2, S31, S33, S34 and S15 which belong to different PNC domains.

The traffic flow between R1 and R5 can be summarized as:

```
R1 ([PKT] -> STM-64), S3 (STM-64 -> [ODU2]), S1 ([ODU2]),  
S2 ([ODU2]), S31 ([ODU2]), S33 ([ODU2]), S34 ([ODU2]),  
S15 ([ODU2]), S18 ([ODU2] -> STM-64), R5 (STM-64 -> [PKT])
```

As described in [section 4.3.2](#), it is assumed that the CNC is capable, via the CMI, to request the setup of an STM-64 Private Line service, providing all the information that the MDSC needs to coordinate the setup of a multi-domain ODU2 connection as well as the adaptation functions on the edge nodes.

In the single-domain case (10Gb IP link between R1 and R3), the traffic flow between R1 and R3 can be summarized as:

R1 ([PKT] -> STM-64), S3 (STM-64 -> [ODU2]), S5 ([ODU2]),
S6 ([ODU2] -> STM-64), R3 (STM-64 -> [PKT])

As described in [section 4.3.1](#), the CNC requests the setup of an STM-64 Private Line service in the same way as before and the information provided at the CMI is sufficient for the MDSC to understand that this is a single-domain service request.

As described in [section 4.3.2](#), the MDSC could just request PNC1 to setup a single-domain STM-64 Private Line service between nodes S3 and S6.

[4.3.4. EVPL over ODU](#)

When the physical links interconnecting the IP routers and the transport network are Ethernet links, it is also possible that different Ethernet services (e.g., EVPL) can share the same physical link using different VLANs.

To setup two 1Gb IP links between R1 to R3 and between R1 and R5, two EVPL services need to be created, supported by two ODU0 end-to-end connections respectively between S3 and S6, crossing transport node S5, and between S3 and S18, crossing transport nodes S1, S2, S31, S33, S34 and S15 which belong to different PNC domains.

Since the two EVPL services are sharing the same Ethernet physical link between R1 and S3, different VLAN IDs are associated with different EVPL services: for example, VLAN IDs 10 and 20 respectively.

The traffic flow between R1 and R5 can be summarized as:

R1 ([PKT] -> VLAN), S3 (VLAN -> [ODU0]), S1 ([ODU0]),
S2 ([ODU0]), S31 ([ODU0]), S33 ([ODU0]), S34 ([ODU0]),
S15 ([ODU0]), S18 ([ODU0] -> VLAN), R5 (VLAN -> [PKT])

The traffic flow between R1 and R3 can be summarized as:

R1 ([PKT] -> VLAN), S3 (VLAN -> [ODU0]), S5 ([ODU0]),
S6 ([ODU0] -> VLAN), R3 (VLAN -> [PKT])

As described in [section 4.3.2](#), it is assumed that the CNC is capable, via the CMI, to request the setup of these EVPL services, providing all the information that the MDSC needs to understand that it need to request PNC1 to setup an EVPL service between nodes S3

and S6 (single-domain service request) and it also needs to coordinate the setup of a multi-domain ODU0 connection between nodes S3 and S16 as well as the adaptation functions on these edge nodes.

4.3.5. EVPLAN and EVPTree Services

When the physical links interconnecting the IP routers and the transport network are Ethernet links, multipoint Ethernet services (e.g, EPLAN and EPTree) can also be supported. It is also possible that multiple Ethernet services (e.g, EVPL, EVPLAN and EVPTree) share the same physical link using different VLANs.

Note - it is assumed that EPLAN and EPTree services can be supported by configuring EVPLAN and EVPTree with port mapping.

Since this EVPLAN/EVPTree service can share the same Ethernet physical links between IP routers and transport nodes (e.g., with the EVPL services described in [section 4.3.4](#)), a different VLAN ID (e.g., 30) can be associated with this EVPLAN/EVPTree service.

In order to setup an IP subnet between R1, R2, R3 and R5, an EVPLAN/EVPTree service needs to be created, supported by two ODUFlex end-to-end connections respectively between S3 and S6, crossing transport node S5, and between S3 and S18, crossing transport nodes S1, S2, S31, S33, S34 and S15 which belong to different PNC domains.

Some MAC Bridging capabilities are also required on some nodes at the edge of the transport network: for example Ethernet Bridging capabilities can be configured in nodes S3 and S6:

- o MAC Bridging in node S3 is needed to select, based on the MAC Destination Address, whether received Ethernet frames should be forwarded to R1 or to the ODUFlex terminating on node S6 or to the other ODUFlex terminating on node S18;
- o MAC bridging function in node S6 is needed to select, based on the MAC Destination Address, whether received Ethernet frames should be sent to R2 or to R3 or to the ODUFlex terminating on node S3.

In order to support an EVPTree service instead of an EVPLAN, additional configuration of the Ethernet Bridging capabilities on the nodes at the edge of the transport network is required.

The traffic flows between R1 and R3, between R3 and R5 and between R1 and R5 can be summarized as:

```
R1 ([PKT] -> VLAN), S3 (VLAN -> [MAC] -> [ODUflex]),
S5 ([ODUflex]), S6 ([ODUflex] -> [MAC] -> VLAN),
R3 (VLAN -> [PKT])
```

```
R3 ([PKT] -> VLAN), S6 (VLAN -> [MAC] -> [ODUflex]),
S5 ([ODUflex]), S3 ([ODUflex] -> [MAC] -> [ODUflex]),
S1 ([ODUflex]), S2 ([ODUflex]), S31 ([ODUflex]),
S33 ([ODUflex]), S34 ([ODUflex]),
S15 ([ODUflex]), S18 ([ODUflex] -> VLAN), R5 (VLAN -> [PKT])
```

```
R1 ([PKT] -> VLAN), S3 (VLAN -> [MAC] -> [ODUflex]),
S1 ([ODUflex]), S2 ([ODUflex]), S31 ([ODUflex]),
S33 ([ODUflex]), S34 ([ODUflex]),
S15 ([ODUflex]), S18 ([ODUflex] -> VLAN), R5 (VLAN -> [PKT])
```

As described in [section 4.3.2](#), it is assumed that the CNC is capable, via the CMI, to request the setup of this EVPLAN/EVPTree service, providing all the information that the MDSC needs to understand that it need to request PNC1 to setup an ODUflex connection between nodes S3 and S6 (single-domain service request) and it also needs to coordinate the setup of a multi-domain ODUflex connection between nodes S3 and S16 as well as the MAC bridging and the adaptation functions on these edge nodes.

In case the CNC needs the setup of an EVPLAN/EVPTree service only between R1, R2 and R3 (single-domain service request), it would request the setup of this service in the same way as before and the information provided at the CMI is sufficient for the MDSC to understand that this is a single-domain service request.

The MDSC can then just request PNC1 to setup a single-domain EVPLAN/EVPTree service between nodes S3 and S6. PNC1 can take care of setting up the single-domain ODUflex end-to-end connection between nodes S3 and S6 as well as of configuring the MAC bridging and the adaptation functions on these edge nodes.

4.3.6. Dynamic Service Configuration

Given the service established in the previous sections, there is a demand for an update of some service characteristics. A straightforward approach would be terminate the current service and replace with a new one. Another more advanced approach would be

dynamic configuration, in which case there will be no interruption for the connection.

An example application would be updating the SLA information for a certain connection. For example, an ODU transit connection is set up according to [section 4.3.1](#), with the corresponding SLA level of 'no protection'. After the establishment of this connection, the user would like to enhance this service by providing a restoration after potential failure, and a request is generated on the CMI. In this case, after receiving the request, the MDSC would need to send an update message to the PNC, changing the SLA parameters in TE Tunnel model. Then the connection characteristic would be changed by PNC, and a notification would be sent to MDSC for acknowledgement.

[4.4. Multi-function Access Links](#)

Some physical links interconnecting the IP routers and the transport network can be configured in different modes, e.g., as OTU2 or STM-64 or 10GE.

This configuration can be done a-priori by means outside the scope of this document. In this case, these links will appear at the MPI either as an ODU Link or as a STM-64 Link or as a 10GE Link (depending on the a-priori configuration) and will be controlled at the MPI as discussed in [section 4.3](#).

It is also possible not to configure these links a-priori and give the control to the MPI to decide, based on the service configuration, how to configure it.

For example, if the physical link between R1 and S3 is a multi-functional access link while the physical links between R7 and S31 and between R5 and S18 are STM-64 and 10GE physical links respectively, it is possible to configure either an STM-64 Private Line service between R1 and R7 or an EPL service between R1 and R5.

The traffic flow between R1 and R7 can be summarized as:

R1 ([PKT] -> STM-64), S3 (STM-64 -> [ODU2]), S1 ([ODU2]),
S2 ([ODU2]), S31 ([ODU2] -> STM-64), R3 (STM-64 -> [PKT])

The traffic flow between R1 and R5 can be summarized as:

```
R1 ([PKT] -> ETH), S3 (ETH -> [ODU2]), S1 ([ODU2]),  
S2 ([ODU2]), S31 ([ODU2]), S33 ([ODU2]), S34 ([ODU2]),  
S15 ([ODU2]), S18 ([ODU2] -> ETH), R5 (ETH -> [PKT])
```

As described in [section 4.3.2](#), it is assumed that the CNC is capable, via the CMI, to request the setup either an STM-64 Private Line service between R1 and R7 or an EPL service between R1 and R5, providing all the information that the MDSC needs to understand that it need to coordinate the setup of a multi-domain ODU2 connection, either between nodes S3 and S31, or between nodes S3 and S18, as well as the adaptation functions on these edge nodes, and in particular whether the multi-function access link on between R1 and S3 should operate as an STM-64 or as a 10GE link.

[4.5. Protection and Restoration Configuration](#)

Protection switching provides a pre-allocated survivability mechanism, typically provided via linear protection methods and would be configured to operate as 1+1 unidirectional (the most common OTN protection method), 1+1 bidirectional or 1:n bidirectional. This ensures fast and simple service survivability.

Restoration methods would provide capability to reroute and restore connectivity traffic around network faults, without the network penalty imposed with dedicated 1+1 protection schemes.

This section describes only services which are protected with linear protection and with dynamic restoration.

The MDSC needs to be capable to coordinate different PNCs to configure protection switching when requesting the setup of the protected connectivity services described in [section 4.3](#).

Since in these service examples, switching within the transport network domain is performed only in the OTN ODU layer, also protection switching within the transport network domain can only be provided at the OTN ODU layer.

[4.5.1. Linear Protection \(end-to-end\)](#)

In order to protect any service defined in [section 4.3](#) from failures within the OTN multi-domain transport network, the MDSC should be capable to coordinate different PNCs to configure and control OTN linear protection in the data plane between nodes S3 and node S18.

It is assumed that the OTN linear protection is configured to with 1+1 unidirectional protection switching type, as defined in [ITU-T G.808.1] and [ITU-T G.873.1], as well as in [[RFC4427](#)].

In these scenarios, a working transport entity and a protection transport entity, as defined in [ITU-T G.808.1], (or a working LSP and a protection LSP, as defined in [[RFC4427](#)]) should be configured in the data plane.

Two cases can be considered:

- o In one case, the working and protection transport entities pass through the same PNC domains:

Working transport entity: S3, S1, S2,
S31, S33, S34,
S15, S18

Protection transport entity: S3, S4, S8,
S32,
S12, S17, S18

- o In another case, the working and protection transport entities can pass through different PNC domains:

Working transport entity: S3, S5, S7,
S11, S12, S17, S18

Protection transport entity: S3, S1, S2,
S31, S33, S34,
S15, S18

The PNCs should be capable to report to the MDSC which is the active transport entity, as defined in [ITU-T G.808.1], in the data plane.

Given the fast dynamic of protection switching operations in the data plane (50ms recovery time), this reporting is not expected to be in real-time.

It is also worth noting that with unidirectional protection switching, e.g., 1+1 unidirectional protection switching, the active transport entity may be different in the two directions.

4.5.2. Segmented Protection

To protect any service defined in [section 4.3](#) from failures within the OTN multi-domain transport network, the MDSC should be capable to request each PNC to configure OTN intra-domain protection when requesting the setup of the ODU2 data plane connection segment.

If PNC1 provides linear protection, the working and protection transport entities could be:

Working transport entity: S3, S1, S2

Protection transport entity: S3, S4, S8, S2

If PNC2 provides linear protection, the working and protection transport entities could be:

Working transport entity: S15, S18

Protection transport entity: S15, S12, S17, S18

If PNC3 provides linear protection, the working and protection transport entities could be:

Working transport entity: S31, S33, S34

Protection transport entity: S31, S32, S34

4.5.3. End-to-End Dynamic restoration

To restore any service defined in [section 4.3](#) from failures within the OTN multi-domain transport network, the MDSC should be capable to coordinate different PNCs to configure and control OTN end-to-end dynamic Restoration in the data plane between nodes S3 and node S18. For example, the MDSC can request the PNC1, PNC2 and PNC3 to create a service with no-protection, MDSC set the end-to-end service with the dynamic restoration.

Working transport entity: S3, S1, S2,
S31, S33, S34,
S15, S18

When a link failure between S1 and S2 occurred in network domain 1, PNC1 does not restore the tunnel and send the alarm notification to the MDSC, MDSC will perform the end-to-end restoration.

Restored transport entity: S3, S4, S8,
S12, S15, S18

4.5.4. Segmented Dynamic Restoration

To restore any service defined in [section 4.3](#) from failures within the OTN multi-domain transport network, the MDSC should be capable to coordinate different PNCs to configure and control OTN segmented dynamic Restoration in the data plane between nodes S3 and node S18.

Working transport entity: S3, S1, S2,
S31, S33, S34,
S15, S18

When a link failure between S1 and S2 occurred in network domain 1, PNC1 will restore the tunnel and send the alarm or tunnel update notification to the MDSC, MDSC will update the restored tunnel.

Restored transport entity: S3, S4, S8, S2
S31, S33, S34,
S15, S18

When a link failure between network domain 1 and network domain 2 occurred, PNC1 and PNC2 will send the alarm notification to the MDSC, MDSC will update the restored tunnel.

Restored transport entity: S3, S4, S8,
S12, S15, S18

In order to improve the efficiency of recovery, the controller can establish a recovery path in a concurrent way. When the recovery fails in one domain or one network element, the rollback operation should be supported.

The creation of the recovery path by the controller can use the method of "make-before-break", in order to reduce the impact of the recovery operation on the services.

4.6. Service Modification and Deletion

To be discussed in future versions of this document.

4.7. Notification

To realize the topology update, service update and restoration function, following notification type should be supported.

1. Object create
2. Object delete
3. Object state change
4. Alarm

Because there are three types of topology abstraction type defined in [section 4.2](#), the notification should also be abstracted. The PNC and MDSC should coordinate together to determine the notification policy, such as when an intra-domain alarm occurred, the PNC may not report the alarm but the service state change notification to the MDSC.

4.8. Path Computation with Constraint

It is possible to have constraint during path computation procedure, typical cases include IRO/XRO and so on. This information is carried in the TE Tunnel model and used when there is a request with constraint. Consider the example in [section 4.3.1](#), the request can be a Tunnel from R1 to R5 with an IRO from S2 to S31, then a qualified feedback would become:

```
R1 ([PKT] -> ODU2), S3 ([ODU2]), S1 ([ODU2]), S2 ([ODU2]),  
S31 ([ODU2]), S33 ([ODU2]), S34 ([ODU2]),  
S15 ([ODU2]), S18 ([ODU2]), R5 (ODU2 -> [PKT])
```

If the request covers the IRO from S8 to S12, then the above path would not be qualified, while a possible computation result may be:

```
R1 ([PKT] -> ODU2), S3 ([ODU2]), S1 ([ODU2]), S2 ([ODU2]),  
S8 ([ODU2]), S12 ([ODU2]), S15 ([ODU2]), S18 ([ODU2]), R5 (ODU2 ->  
[PKT])
```

Similarly, the XRO can be represented by TE tunnel model as well.

When there is a technology specific network (e.g, OTN), the corresponding technology (OTN) model should also be used to specify

the tunnel information on MPI, with the constraint included in TE Tunnel model.

5. YANG Model Analysis

This section provides a high-level overview of how IETF YANG models can be used at the MPIs, between the MDSC and the PNCs, to support the scenarios described in [section 4](#).

[Section 5.1](#) describes the different topology abstractions provided to the MDSC by each PNC via its own MPI.

[Section 5.2](#) describes how the MDSC can coordinate different requests to different PNCs, via their own MPIs, to setup the different services described in [section 4.3](#).

[Section 5.3](#) describes how the protection scenarios can be deployed, including end-to-end protection and segment protection, for both intra-domain and inter-domain scenario.

5.1. YANG Models for Topology Abstraction

Each PNC reports its respective abstract topology to the MDSC, as described in [section 4.2](#).

5.1.1.1. Domain 1 Topology Abstraction

PNC1 provides the required topology abstraction to expose at its MPI toward the MDSC (called "MPI1") one TE Topology instance for the ODU layer (called "MPI1 ODU Topology"), containing one TE Node (called "ODU Node") for each physical node, as shown in Figure 3 below.

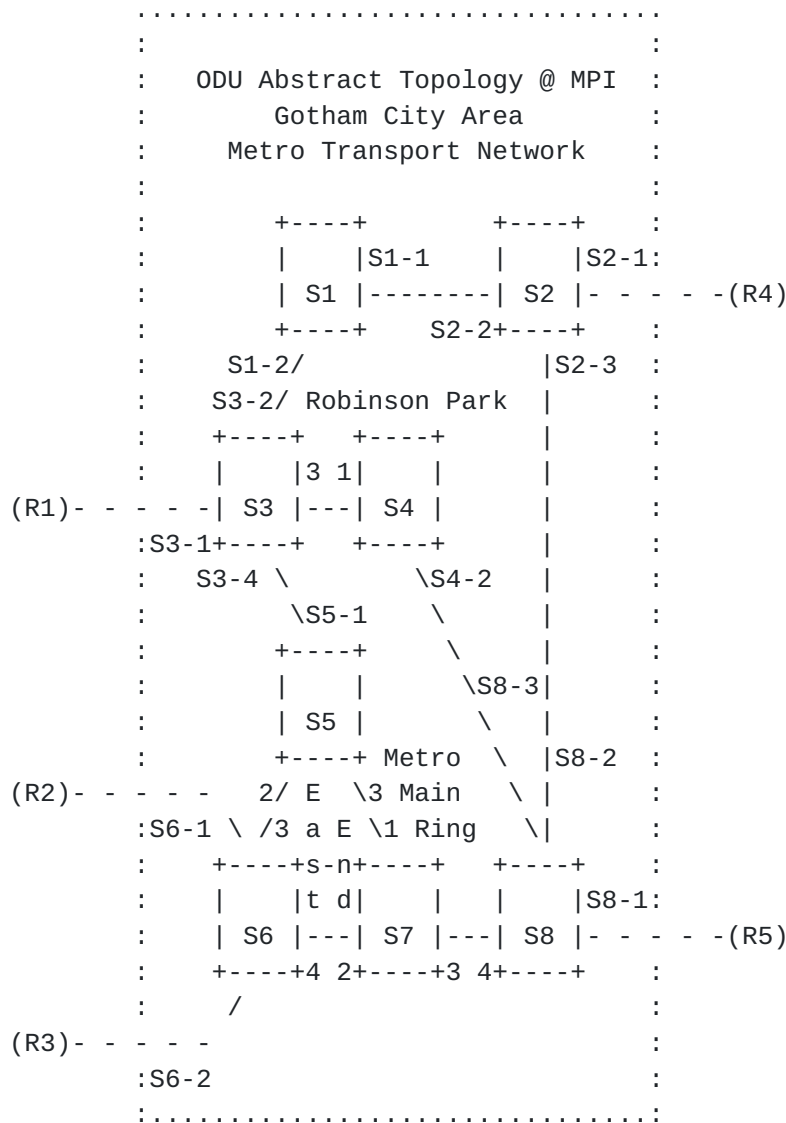


Figure 3 Abstract Topology exposed at MPI1 (MPI1 ODU Topology)

The ODU Nodes in Figure 3 are using the same names as the physical nodes to simplify the description of the mapping between the ODU Nodes exposed by the Transport PNCs at the MPI and the physical

nodes in the data plane. This does not correspond to the reality of the usage of the topology model, as described in [section 4.3](#) of [TE-TOPO], in which renaming by the client it is necessary.

As described in [section 4.1](#), it is assumed that the physical links between the physical nodes are pre-configured and therefore PNC1 exports at MPI1 one TE Link (called "ODU Link") for each of these OTU4 trails.

[Appendix B.1.1](#) provides the detailed JSON code ("mpi1-otn-topology.json") describing how this ODU Topology is reported by the PNC, using the [TE-TOPO] and [OTN-TOPO] YANG models at MPI1.

5.1.2. Domain 2 Grey (Type A) Topology Abstraction

PNC2 provides the required topology abstraction to expose at its MPI towards the MDSC (called "MPI2") only one abstract node (i.e., AN2), with only inter-domain and access links, is reported at the MPI2.

5.1.3. Domain 3 Grey (Type B) Topology Abstraction

PNC3 provides the required topology abstraction to expose at its MPI towards the MDSC (called "MPI3") only two abstract nodes (i.e., AN31 and AN32), with internal links, inter-domain links and access links.

5.1.4. Multi-domain Topology Stitching

As assumed in the beginning of this section, MDSC does not have any knowledge of the topologies of each domain until each PNC reports its own abstraction topology, so the MDSC needs to merge together the abstract topologies provided by different PNCs, at the MPIs, to build its own topology view, as described in [section 4.3](#) of [TE-TOPO].

Given the topologies reported from multiple PNCs, the MDSC need to stitch the multi-domain topology and obtain the full map of topology. The topology of each domain main be in an abstracted shape (refer to [section 5.2](#) of [ACTN-Fwk] for different level of abstraction), while the inter-domain link information MUST be complete and fully configured by the MDSC.

The inter-domain link information is reported to the MDSC by the two PNCs, controlling the two ends of the inter-domain link.

The MDSC needs to understand how to "stitch" together these inter-domain links.

One possibility is to use the plug-id information, defined in [TE-TOPO]: two inter-domain links reporting the same plug-id value can be merged as a single intra-domain link within any MDSC native topology. The value of the reported plug-id information can be either assigned by a central network authority, and configured within the two PNC domains, or it can be discovered using automatic discovery mechanisms (e.g., LMP-based, as defined in [[RFC6898](#)]).

In case the plug-id values are assigned by a central authority, it is under the central authority responsibility to assign unique values.

In case the plug-id values are automatically discovered, the information discovered by the automatic discovery mechanisms needs to be encoded as a bit string within the plug-id value. This encoding is implementation specific but the encoding rules need to be consistent across all the PNCs.

In case of co-existence within the same network of multiple sources for the plug-id (e.g., central authority and automatic discovery or even different automatic discovery mechanisms), it is RECOMMENDED that the plug-id namespace is partitioned to avoid that different sources assign the same plug-id value to different inter-domain link. The encoding of the plug-id namespace within the plug-id value is implementation specific but needs to be consistent across all the PNCs.

Another possibility is to pre-configure, either in the adjacent PNCs or in the MDSC, the association between the inter-domain link identifiers (topology-id, node-id and tp-id) assigned by the two adjacent PNCs to the same inter-domain link.

This last scenario requires further investigation and will be discussed in a future version of this document.

[5.1.5. Access Links](#)

Access links in Figure 3 are shown as ODU Links: the modeling of the access links for other access technologies is currently an open issue.

The modeling of the access link in case of non-ODU access technology has also an impact on the need to model ODU TTPs and layer transition capabilities on the edge nodes (e.g., nodes S2, S3, S6 and S8 in Figure 3).

If, for example, the physical NE S6 is implemented in a "pizza box", the data plane would have only set of ODU termination resources (where up to 2xODU4, 4xODU3, 20xODU2, 80xODU1, 160xODU0 and 160xODUflex can be terminated). The traffic coming from each of the 10GE access links can be mapped into any of these ODU terminations.

Instead if, for example, the physical NE S6 can be implemented as a multi-board system where access links reside on different/dedicated access cards with separated set of ODU termination resources (where up to 1xODU4, 2xODU3, 10xODU2, 40xODU1, 80xODU0 and 80xODUflex for each resource can be terminated). The traffic coming from one 10GE access links can be mapped only into the ODU terminations which reside on the same access card.

The more generic implementation option for a physical NE (e.g., S6) would be case is of a multi-board system with multiple access cards with separated sets of access links and ODU termination resources (where up to 1xODU4, 2xODU3, 10xODU2, 40xODU1, 80xODU0 and 80xODUflex for each resource can be terminated). The traffic coming from each of the 10GE access links on one access card can be mapped only into any of the ODU terminations which reside on the same access card.

In the last two cases, only the ODUs terminated on the same access card where the access links resides can carry the traffic coming from that 10GE access link. Terminated ODUs can instead be sent to any of the OTU4 interfaces

In all these cases, terminated ODUs can be sent to any of the OTU4 interfaces assuming the implementation is based on a non-blocking ODU cross-connect.

If the access links are reported via MPI in some, still to be defined, client topology, it is possible to report each set of ODU termination resources as an ODU TTP within the ODU Topology of Figure 3 and to use either the inter-layer lock-id or the transitional link, as described in sections [3.4](#) and [3.10](#) of [TE-TOP0], to correlate the access links, in the client topology, with the ODU TTPs, in the ODU topology, to which access link are connected to.

[5.2.](#) YANG Models for Service Configuration

The service configuration procedure is assumed to be initiated (step 1 in Figure 4) at the CMI from CNC to MDSC. Analysis of the CMI models is (e.g., L1SM, L2SM, Transport-Service, VN, et al.) is outside the scope of this document.

As described in [section 4.3](#), it is assumed that the CMI YANG models provides all the information that allows the MDSC to understand that it needs to coordinate the setup of a multi-domain ODU connection (or connection segment) and, when needed, also the configuration of the adaptation functions in the edge nodes belonging to different domains.

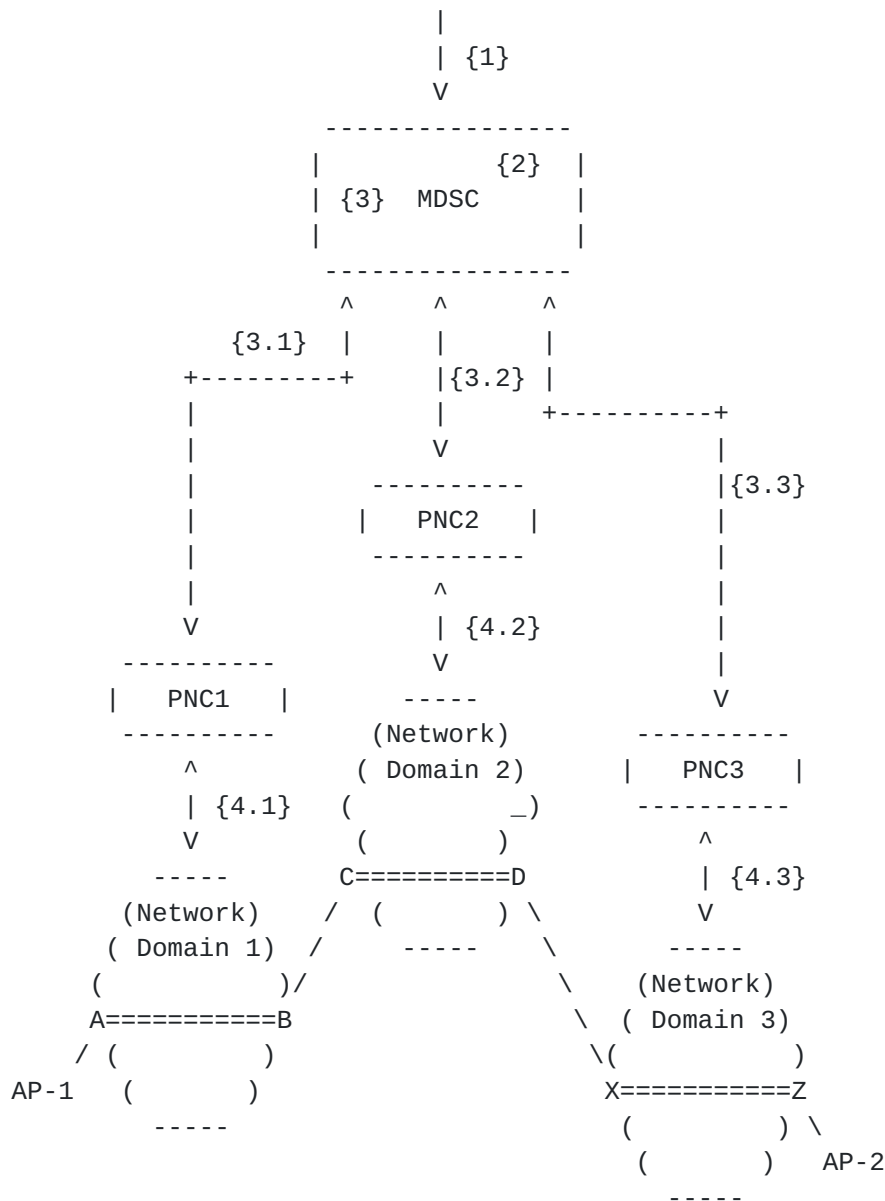


Figure 4 Multi-domain Service Setup

As an example, the objective in this section is to configure a transport service between R1 and R5. The cross-domain routing is assumed to be R1 <-> S3 <-> S2 <-> S31 <-> S33 <-> S34 <-> S15 <-> S18 <-> R5.

According to the different client signal type, there is different adaptation required.

After receiving such request, MDSC determines the domain sequence, i.e., domain 1 <-> domain 2 <-> domain 3, with corresponding PNCs and inter-domain links (step 2 in Figure 4).

As described in [[PATH-COMPUTE](#)], the domain sequence can be determined by running the MDSC own path computation on the MDSC internal topology, defined in [section 5.1.4](#), if and only if the MDSC has enough topology information. Otherwise the MDSC can send path computation requests to the different PNCs (steps 2.1, 2.2 and 2.3 in Figure 4) and use this information to determine the optimal path on its internal topology and therefore the domain sequence.

The MDSC will then decompose the tunnel request into a few tunnel segments via tunnel model (including both TE tunnel model and OTN tunnel model), and request different PNCs to setup each intra-domain tunnel segment (steps 3, 3.1, 3.2 and 3.3 in Figure 4).

Assume that each intra-domain tunnel segment can be set up successfully, and each PNC response to the MDSC respectively. Based on each segment, MDSC will take care of the configuration of both the intra-domain tunnel segment and inter-domain tunnel via corresponding MPI (via TE tunnel model and OTN tunnel model). More specifically, for the inter-domain configuration, the ts-bitmap and tpn attributes need to be configured using the OTN Tunnel model [xxx]. Then the end-to-end OTN tunnel will be ready.

In any case, the access link configuration is done only on the PNCs that control the access links (e.g., PNC-1 and PNC-3 in our example) and not on the PNCs of transit domain (e.g., PNC-2 in our example). Access link will be configured by MDSC after the OTN tunnel is set up. Access configuration is different and dependent on the different type of service. More details can be found in the following sections.

[5.2.1](#). ODU Transit Service

In this scenario, described in [section 4.3.1](#), the access links are configured as ODU Links.

Since it is assumed that the physical access links are pre-configured, each PNC exposes, at its MPI, one TE Link (called "ODU Link") for each of these physical access link. These links are reported, together with any other ODU internal or inter-domain link, within the OTN abstract topology exposed by each PNC, at its own MPI.

To setup this IP link, between R1 and R5, the CNC requests, at the CMI, the MDSC to setup an ODU transit service.

From the topology information described in [section 5.1](#) above, the MDSC understands that R1 is attached to the access link terminating on S3-1 LTP in the ODU Topology exposed by PNC1 and that R5 is attached to the access link terminating on AN2-1 LTP in the ODU Topology exposed by PNC2.

MDSC would then request, at MPI1, the PNC1 to setup an ODU2 (Transit Segment) Tunnel with one primary path between S3-1 and S2-1 LTPs:

- o Source and Destination TTPs are not specified (since it is a Transit Tunnel)
- o Ingress and egress points are indicated in the route-object-include-exclude list of the explicit-route-objects of the primary path:
 - o The first element references the access link terminating on S3-1 LTP
 - o The last two element references respectively the inter-domain link terminating on S2-1 LTP and the data plane resources (i.e., the timeslots and the TPN, called "OTN Label") used by the ODU2 connection over that link.

The configuration of the timeslots used by the ODU2 connection on the internal links within a PNC domain (i.e., on the internal links domain) is outside the scope of this document since it is a matter of the PNC domain internal implementation.

However, the configuration of the timeslots used by the ODU2 connection at the transport network domain boundaries (e.g., on the inter-domain links) needs to take into account the timeslots available on physical nodes belonging to different PNC domains (e.g., on node S2 within PNC1 domain and on node S31 within PNC3 domain).

The MDSC, when coordinating the setup of a multi-domain ODU connection, also configures the data plane resources (i.e., the timeslots and the TPN) to be used on the inter-domain links. The MDSC can know the timeslots which are available on the physical OTN nodes terminating the inter-domain links (e.g., S2 and S31) from the OTN Topology information exposed, at the MPIs, by the PNCs

controlling the OTN physical nodes (e.g., PNC1 and PNC3 controlling respectively the physical nodes S2 and S31).

[Appendix B.2.1](#) provides the detailed JSON code ("mpi1-odu2-service-config.json") describing how the setup of this ODU2 (Transit Segment) Tunnel can be requested by the MDSC, using the [\[TE-TUNNEL\]](#) and [\[OTN-TUNNEL\]](#) YANG models at MPI1.

The Transport PNC performs path computation and sets up the ODU2 cross-connections within the physical nodes S3, S5 and S6, as shown in [section 4.3.1](#).

[5.2.1.1](#). Single Domain Example

To setup an ODU2 end-to-end connection, supporting an IP link, between R1 and R3, the CNC requests, at the CMI, the MDSC to setup an ODU transit service.

The Transport PNC reports the status of the created ODU2 (Transit Segment) Tunnel and its path within the ODU Topology as shown in Figure 5 below:

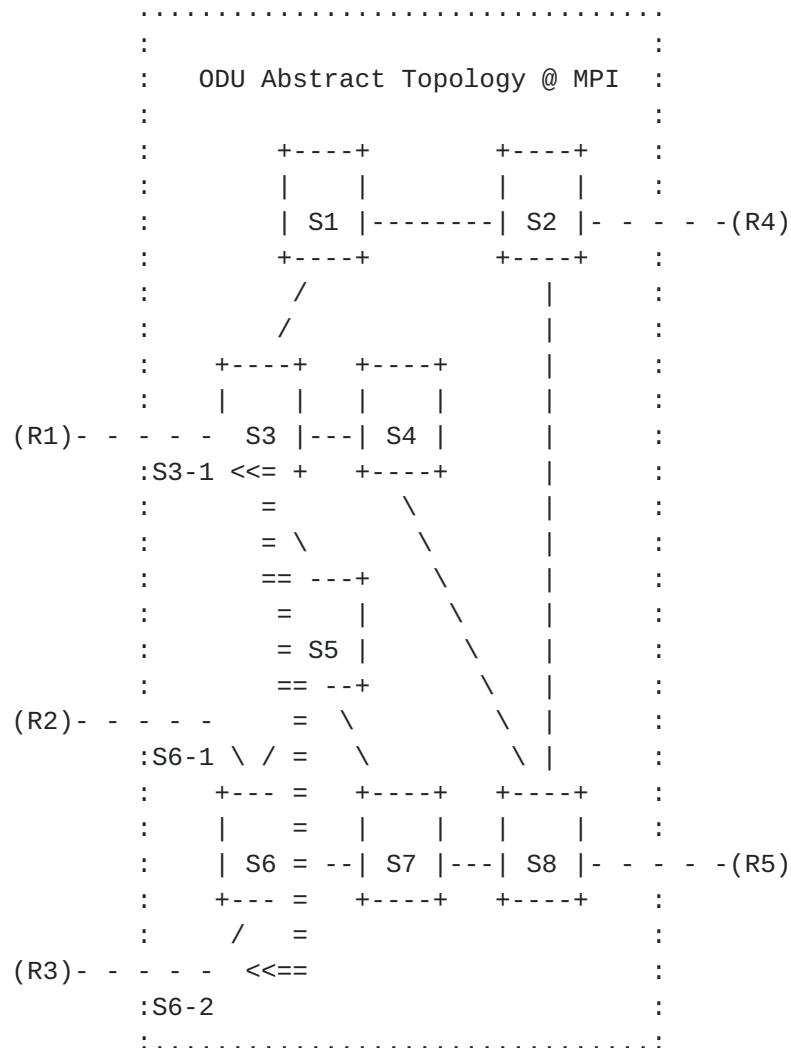


Figure 5 ODU2 Transit Tunnel

5.2.2. EPL over ODU Service

In this scenario, described in [section 4.3.2](#), the access links are configured as Ethernet Links.

To setup this IP link, between R1 and R5, the CNC requests, at the CMI, the MDSC to setup an EPL service.

As described in [section 5.1.5](#) above, it is not clear in this case how the Ethernet access links between the transport network and the IP router, are reported by the PNC to the MDSC.

If the 10GE physical links are not reported as ODU links within the ODU topology information, described in [section 5.1.1](#) above, then the MDSC will not have sufficient information to know that R1 and R5 are attached to the access links terminating on S3 and S6.

Assuming that the MDSC knows how R1 and R3 are attached to the transport network, the MDSC would request the Transport PNC to setup an ODU2 end-to-end Tunnel between S3 and S6.

This ODU Tunnel is setup between two TTPs of nodes S3 and S6. In case nodes S3 and S6 support more than one TTP, the MDSC should decide which TTP to use.

As discussed in 5.1.5, depending on the different hardware implementations of the physical nodes S3 and S6, not all the access links can be connected to all the TTPs. The MDSC should therefore not only select the optimal TTP but also a TTP that would allow the Tunnel to be used by the service.

It is assumed that in case node S3 or node S6 supports only one TTP, this TTP can be accessed by all the access links.

[Appendix B.2.2](#) provides the detailed JSON code ("mpi1-odu2-tunnel-config.json") describing how the setup of this ODU2 (Head Segment) Tunnel can be requested by the MDSC, using the [\[TE-TUNNEL\]](#) and [\[OTN-TUNNEL\]](#) YANG models at MPI1.

Once the ODU2 Tunnel setup has been requested, unless there is a one-to-one relationship between the S3 and S6 TTPs and the Ethernet access links toward R1 and R3 (as in the case, described in [section 5.1.5](#), where the Ethernet access links reside on different/dedicated access card such that the ODU2 tunnel can only carry the Ethernet traffic from the only Ethernet access link on the same access card where the ODU2 tunnel is terminated), the MDSC also needs to request the setup of an EPL service from the access links on S3 and S6, attached to R1 and R3, and this ODU2 Tunnel.

[Appendix B.2.3](#) provides the detailed JSON code ("mpi1-epl-service-config.json") describing how the setup of this EPL service using the ODU2 Tunnel can be requested by the MDSC, using the [\[CLIENT-SVC\]](#) YANG model at MPI1.

5.2.3. Other OTN Client Services

In this scenario, the access links are configured as one of the OTN clients (e.g., STM-64) links.

As described in [section 4.3.3](#), the CNC needs to setup an STM-64 Private Link service, supporting an IP link, between R1 and R3 and requests this service at the CMI to the MDSC.

MDSC needs to setup an STM-64 Private Link service between R1 and R3 supported by an ODU2 end-to-end connection between S3 and S6.

As described in [section 5.1.5](#) above, it is not clear in this case how the access links (e.g., the STM-N access links) between the transport network and the IP router, are reported by the PNC to the MDSC.

The same issues, as described in [section 5.2.2](#), apply here:

- o the MDSC needs to understand that R1 and R3 are connected, through STM-64 access links, with S3 and S6
- o the MDSC needs to understand which TTPs in S3 and S6 can be accessed by these access links
- o the MDSC needs to configure the private line service from these access links through the ODU2 tunnel

5.2.4. EVPL over ODU Service

In this scenario, the access links are configured as Ethernet links, as described in [section 5.2.2](#) above.

As described in [section 4.3.4](#), the CNC needs to setup EVPL services, supporting IP links, between R1 and R3, as well as between R1 and R4 and requests these services at the CMI to the MDSC.

MDSC needs to setup two EVPL services, between R1 and R3, as well as between R1 and R4, supported by ODU0 end-to-end connections between S3 and S6 and between S3 and S2 respectively.

As described in [section 5.1.5](#) above, it is not clear in this case how the Ethernet access links between the transport network and the IP router, are reported by the PNC to the MDSC.

The same issues, as described in [section 5.1.5](#) above, apply here:

- o the MDSC needs to understand that R1, R3 and R4 are connected, though the Ethernet access links, with S3, S6 and S2
- o the MDSC needs to understand which TTPs in S3, S6 and S2 can be accessed by these access links
- o the MDSC needs to configure the EVPL services from these access links through the ODU0 tunnels

In addition, the MDSC needs to get the information that the access links on S3, S6 and S2 are capable to support EVPL (rather than just EPL) as well as to coordinate the VLAN configuration, for each EVPL service, on these access links (this is a similar issue as the timeslot configuration on access links discussed in [section 4.3.1](#) above).

[5.3.](#) YANG Models for Protection Configuration

[5.3.1.](#) Linear Protection (end-to-end)

To be discussed in future versions of this document.

[5.3.2.](#) Segmented Protection

To be discussed in future versions of this document.

[6.](#) Security Considerations

This section is for further study

[7.](#) IANA Considerations

This document requires no IANA actions.

[8.](#) References

[8.1.](#) Normative References

[RFC7926] Farrel, A. et al., "Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks", [BCP 206](#), [RFC 7926](#), July 2016.

- [RFC4427] Mannie, E., Papadimitriou, D., "Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)", [RFC 4427](#), March 2006.
- [ACTN-Frame] Ceccarelli, D., Lee, Y. et al., "Framework for Abstraction and Control of Transport Networks", [draft-ietf-teas-actn-framework](#), work in progress.
- [ITU-T G.709] ITU-T Recommendation G.709 (06/16), "Interfaces for the optical transport network", June 2016.
- [ITU-T G.808.1] ITU-T Recommendation G.808.1 (05/14), "Generic protection switching - Linear trail and subnetwork protection", May 2014.
- [ITU-T G.873.1] ITU-T Recommendation G.873.1 (05/14), "Optical transport network (OTN): Linear protection", May 2014.
- [TE-TOP0] Liu, X. et al., "YANG Data Model for TE Topologies", [draft-ietf-teas-yang-te-topo](#), work in progress.
- [OTN-TOP0] Zheng, H. et al., "A YANG Data Model for Optical Transport Network Topology", [draft-ietf-ccamp-otn-topo-yang](#), work in progress.
- [CLIENT-TOP0] Zheng, H. et al., "A YANG Data Model for Client-layer Topology", [draft-zheng-ccamp-client-topo-yang](#), work in progress.
- [TE-TUNNEL] Saad, T. et al., "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", [draft-ietf-teas-yang-te](#), work in progress.
- [PATH-COMPUTE] Busi, I., Belotti, S. et al, "Yang model for requesting Path Computation", [draft-busibel-teas-yang-path-computation](#), work in progress.
- [OTN-TUNNEL] Zheng, H. et al., "OTN Tunnel YANG Model", [draft-ietf-ccamp-otn-tunnel-model](#), work in progress.
- [CLIENT-SVC] Zheng, H. et al., "A YANG Data Model for Optical Transport Network Client Signals", [draft-zheng-ccamp-otn-client-signal-yang](#), work in progress.

8.2. Informative References

- [RFC5151] Farrel, A. et al., "Inter-Domain MPLS and GMPLS Traffic Engineering --Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions", [RFC 5151](#), February 2008.
- [RFC6898] Li, D. et al., "Link Management Protocol Behavior Negotiation and Configuration Modifications", [RFC 6898](#), March 2013.
- [RFC8309] Wu, Q. et al., "Service Models Explained", [RFC 8309](#), January 2018.
- [ACTN-YANG] Zhang, X. et al., "Applicability of YANG models for Abstraction and Control of Traffic Engineered Networks", [draft-zhang-teas-actn-yang](#), work in progress.
- [I2RS-TOP0] Clemm, A. et al., "A Data Model for Network Topologies", [draft-ietf-i2rs-yang-network-topo](#), work in progress.
- [ONF TR-527] ONF Technical Recommendation TR-527, "Functional Requirements for Transport API", June 2016.
- [ONF GitHub] ONF Open Transport (SNOWMASS)
<https://github.com/OpenNetworkingFoundation/Snowmass-ONFOpenTransport>

9. Acknowledgments

The authors would like to thank all members of the Transport NBI Design Team involved in the definition of use cases, gap analysis and guidelines for using the IETF YANG models at the Northbound Interface (NBI) of a Transport SDN Controller.

The authors would like to thank Xian Zhang, Anurag Sharma, Sergio Belotti, Tara Cummings, Michael Scharf, Karthik Sethuraman, Oscar Gonzalez de Dios, Hans Bjursrom and Italo Busi for having initiated the work on gap analysis for transport NBI and having provided foundations work for the development of this document.

The authors would like to thank the authors of the TE Topology and Tunnel YANG models [[TE-TOP0](#)] and [[TE-TUNNEL](#)], in particular Igor Bryskin, Vishnu Pavan Beeram, Tarek Saad and Xufeng Liu, for their support in addressing any gap identified during the analysis work.

This document was prepared using 2-Word-v2.0.template.dot.

Appendix A Validating a JSON fragment against a YANG Model

The objective is to have a tool that allows validating whether a piece of JSON code embedded in an Internet-Draft is compliant with a YANG model without using a client/server.

[A.1. Manipulation of JSON fragments](#)

This section describes the various ways JSON fragments are used in the I-D processing and how to manage them.

Let's call "folded-JSON" the JSON embedded in the I-D: it fits the 72 chars width and it is acceptable for it to be invalid JSON.

We then define "unfolded-JSON" a valid JSON fragment having the same contents of the "folded-JSON" without folding, i.e. limits on the text width. The folding/unfolding operation may be done according to [draft-kwatsen-netmod-artwork-folding](#). The "unfolded-JSON" can be edited by the authors using JSON editors with the advantages of syntax validation and pretty-printing.

Both the "folded" and the "unfolded" JSON fragments can include comments having descriptive fields and directives we'll describe later to facilitate the reader and enable some automatic processing.

The presence of comments in the "unfolded-JSON" fragment makes it an invalid JSON encoding of YANG data. Therefore we call "naked JSON" the JSON where the comments have been stripped out: not only it is valid JSON but it is a valid JSON encoding of YANG data.

The following schema resumes these definitions:

	unfold_it -->		stripper -->	
	Folded-JSON	Unfolded-JSON	Naked JSON	
	<-- fold_it		<-- author edits	
<=72-chars?	MUST	MAY	MAY	
valid JSON?	MAY	MUST	MUST	
JSON-encoding	MAY	MAY	MUST	
of YANG data				

Our validation toolchain has been designed to take a JSON in any of the three formats and validate it automatically against a set of relevant YANG modules using available open-source tools. It can be found at: <https://github.com/GianmarcoBruno/json-yang/>

A.2. Comments in JSON fragments

We found useful to introduce two kinds of comments, both defined as key-value pairs where the key starts with "//":

- free-form descriptive comments, e.g. "// COMMENT" : "refine this" to describe properties of JSON fragments.

- machine-usable directives e.g. "// __REFERENCES__DRAFTS__" : { "ietf-routing-types@2017-12-04": "[rfc8294](#)", } which can be used to automatically download from the network the relevant I-Ds or RFCs and extract from them the YANG models of interest. This is particularly useful to keep consistency when the drafting work is rapidly evolving.

A.3. Validation of JSON fragments: DSDL-based approach

The idea is to generate a JSON driver file (JTOX) from YANG, then use it to translate JSON to XML and validate it against the DSDL schemas, as shown in Figure 6.

Useful link: <https://github.com/mbj4668/pyang/wiki/XmlToJson>

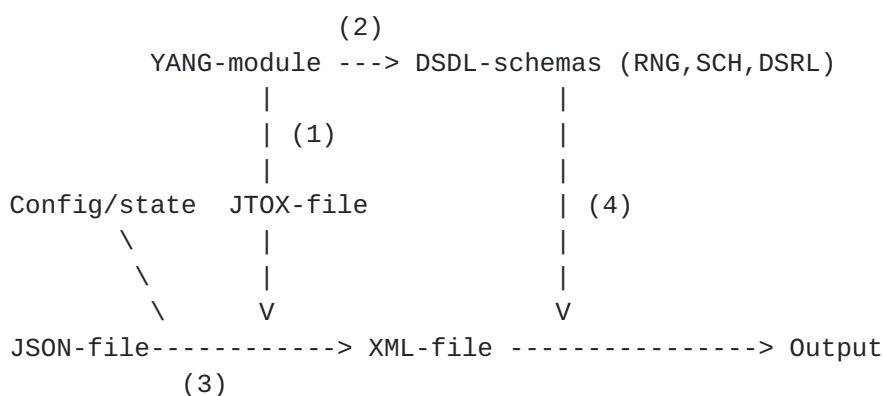


Figure 6 - DSDL-based approach for JSON code validation

In order to allow the use of comments following the convention defined in [section 3](#) without impacting the validation process, these comments will be automatically removed from the JSON-file that will be validate.

A.4. Validation of JSON fragments: why not using a XSD-based approach

This approach has been analyzed and discarded because no longer supported by pyang.

The idea is to convert YANG to XSD, JSON to XML and validate it against the XSD, as shown in Figure 7:

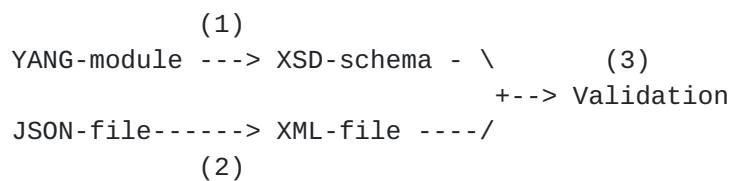


Figure 7 - XSD-based approach for JSON code validation

The pyang support for the XSD output format was deprecated in 1.5 and removed in 1.7.1. However pyang 1.7.1 is necessary to work with YANG 1.1 so the process shown in Figure 7 will stop just at step (1).

Appendix B Detailed JSON Examples

B.1. JSON Examples for Topology Abstractions**B.1.1. JSON Code: mpi1-otn-topology.json**

```
{
  "// __TITLE__": "ODU Abstract Topology @ MPI1",
  "// __LAST_UPDATE__": "July 2, 2018",
  "// __RESTCONF_OPERATION__": {
    "operation": "GET",
    "url":
      "http://{{PNC1-ADDR}}/restconf/data/ietf-network:networks"
  },
  "// __REFERENCE_DRAFTS__": {
    "ietf-network@2017-12-18":
      "draft-ietf-i2rs-yang-network-topo-20",
    "ietf-network-topology@2017-12-18":
      "draft-ietf-i2rs-yang-network-topo-20",
    "ietf-te-topology@2018-02-21":
      "draft-ietf-teas-yang-te-topo-15",
    "ietf-te-types@2018-02-19": "draft-ietf-teas-yang-te-12",
    "ietf-routing-types@2017-12-04": "rfc8294",
    "ietf-otn-topology@2017-10-30":
      "draft-ietf-ccamp-otn-topo-yang-02",
    "ietf-otn-types@2017-10-30":
      "draft-ietf-ccamp-otn-tunnel-model-01"
  },
  "// __MISSING_ATTRIBUTES__": true,
  "ietf-network:networks": {
    "network": [
      {
        "network-id": "tnbi",
        "network-types": {
          "ietf-te-topology:te-topology": {
            "ietf-otn-topology:otn-topology": {}
          }
        },
        "// __DISCUSS__ ietf-te-topology:provider-id": null,
        "ietf-te-topology:provider-id": 0,
        "// __DISCUSS__ ietf-te-topology:client-id": 0,
        "ietf-te-topology:client-id": 0,
        "// __DISCUSS__ ietf-te-topology:te-topology-id": null,
        "ietf-te-topology:te-topology-id": "tnbi",
        "// comment": [

```

```

    "te container presence requires:          ",
    " provider-id, client-id and te-topology-id"
  ],
  "ietf-te-topology:te": {
    "name": "gotham-city:metro-transport-network"
  },
  "ietf-network:node": [
    {
      "/// __NODE__:__DESCRIPTION__": [
        "S3",
        "10.0.0.3",
        "ADM"
      ],
      "node-id": "10.0.0.3",
      "ietf-te-topology:te-node-id": "10.0.0.3",
      "ietf-te-topology:te": {
        "oper-status": "up",
        "te-node-attributes": {
          "/// comment-on-name": [
            "Often transport domains contain subdomain      ",
            "topological partitioning that may be reported   ",
            "in node te name                                   "
          ],
          "name": "S3-metro_main_ring-gateway",
          "admin-status": "up",
          "/// __DISCUSS__ domain-id": 65001
        }
      },
      "/// __DISCUSS__ tunnel-termination-point": []
    },
    "ietf-network-topology:termination-point": [
      {
        "/// __DESCRIPTION__:__LTP__": [
          "S3-1 LTP",
          "connected to (C-R1)",
          "unnumberd/ifIndex: 1",
          "OTU-2",
          "tributary port"
        ],
        "tp-id": "1",
        "ietf-te-topology:te-tp-id": 1,
        "ietf-te-topology:te": {
          "/// _OBSOLETE_ ietf-otn-topology:client-facing": [
            null
          ],
          "admin-status": "up",

```

```
        "oper-status": "up",
        "// ietf-otn-topology:supported-payload-types":
            "__DISCUSS__",
        "// _OBSOLETE_ ietf-otn-topology:protocol-type":
            "ietf-transport-types:prot-OTU2",
        "// _OBSOLETE_ ietf-otn-topology:adaptation-type":
            "ODU"
    }
},
{
    "// __DESCRIPTION__:__LTP__": [
        "S3-2 LTP",
        "connected to S1-2",
        "unnumberd/ifIndex: 2",
        "OTU-4",
        "line port"
    ],
    "tp-id": "2",
    "ietf-te-topology:te-tp-id": 2,
    "ietf-te-topology:te": {
        "admin-status": "up",
        "oper-status": "up",
        "// ietf-otn-topology:supported-payload-types":
            "__DISCUSS__",
        "// _OBSOLETE_ ietf-otn-topology:protocol-type":
            "ietf-transport-types:prot-OTU4",
        "// _OBSOLETE_ ietf-otn-topology:adaptation-type":
            "ODU"
    }
},
{
    "// __DESCRIPTION__:__LTP__": [
        "S3-3 LTP",
        "connected to S4-1",
        "unnumberd/ifIndex: 3",
        "OTU-4",
        "line port"
    ],
    "tp-id": "3",
    "ietf-te-topology:te-tp-id": 3,
    "ietf-te-topology:te": {
        "admin-status": "up",
        "oper-status": "up",
        "// ietf-otn-topology:supported-payload-types":
            "__DISCUSS__",
```



```

        "// _OBSOLETE_ ietf-otn-topology:protocol-type":
        "ietf-transport-types:prot-OTU4",
        "// _OBSOLETE_ ietf-otn-topology:adaptation-type":
        "ODU"
    }
},
{
    "// __DESCRIPTION__:__LTP__": [
        "S3-4 LTP",
        "connected to S5-1",
        "unnumberd/ifIndex: 4",
        "OTU-4",
        "line port"
    ],
    "tp-id": "4",
    "ietf-te-topology:te-tp-id": 4,
    "ietf-te-topology:te": {
        "admin-status": "up",
        "oper-status": "up",
        "// ietf-otn-topology:supported-payload-types":
        "__DISCUSS__",
        "// _OBSOLETE_ ietf-otn-topology:protocol-type":
        "ietf-transport-types:prot-OTU4",
        "// _OBSOLETE_ ietf-otn-topology:adaptation-type":
        "ODU"
    }
}
]
},
{
    "// __NODE__:__DESCRIPTION__": [
        "S4",
        "10.0.0.4",
        "pizza-box"
    ],
    "node-id": "10.0.0.4",
    "ietf-te-topology:te-node-id": "10.0.0.4",
    "ietf-te-topology:te": {
        "// comment": "TO BE COMPLETED",
        "oper-status": "up",
        "te-node-attributes": {
            "name": "S4-line-metro_main_ring",
            "admin-status": "up",
            "// __DISCUSS__ domain-id": 65001
        }
    },

```

```
    "// __DISCUSS__ tunnel-termination-point": []
  },
  "ietf-network-topology:termination-point": [
    {
      "// __DESCRIPTION__:__LTP__": [
        "S4-1 LTP",
        "connected to S3-3",
        "unnumberd/ifIndex: 1",
        "OTU-4",
        "line port"
      ],
      "// comment": "S4-1 LTP",
      "tp-id": "1",
      "ietf-te-topology:te-tp-id": 1,
      "ietf-te-topology:te": {
        "admin-status": "up",
        "oper-status": "up",
        "// ietf-otn-topology:supported-payload-types":
          "__DISCUSS__",
        "// _OBSOLETE_ ietf-otn-topology:protocol-type":
          "ietf-transport-types:prot-OTU4",
        "// _OBSOLETE_ ietf-otn-topology:adaptation-type":
          "ODU"
      }
    },
    {
      "// __DESCRIPTION__:__LTP__": [
        "S4-2 LTP",
        "connected to S8-3",
        "unnumberd/ifIndex: 2",
        "OTU-4",
        "line port"
      ],
      "// comment": "S4-2 LTP",
      "tp-id": "2",
      "ietf-te-topology:te-tp-id": 2,
      "ietf-te-topology:te": {
        "admin-status": "up",
        "oper-status": "up",
        "// ietf-otn-topology:supported-payload-types":
          "__DISCUSS__",
        "// _OBSOLETE_ ietf-otn-topology:protocol-type":
          "ietf-transport-types:prot-OTU4",
        "// _OBSOLETE_ ietf-otn-topology:adaptation-type":
          "ODU"
      }
    }
  ]
}
```

```

    }
  }
]
},
{
  "// __NODE__:__DESCRIPTION__": [
    "S1",
    "10.0.0.1",
    "pizza-box"
  ],
  "node-id": "10.0.0.1",
  "ietf-te-topology:te-node-id": "10.0.0.1",
  "ietf-te-topology:te": {
    "// comment": "TO BE COMPLETED",
    "oper-status": "up",
    "te-node-attributes": {
      "name": "S1-robinson_park_ring-line",
      "admin-status": "up",
      "// __DISCUSS__ domain-id": 65001
    },
    "// __DISCUSS__ tunnel-termination-point": []
  },
  "ietf-network-topology:termination-point": [
    {
      "// __DESCRIPTION__:__LTP__": [
        "S1-1 LTP",
        "connected to S2-2",
        "unnumberd/ifIndex: 1",
        "OTU-4",
        "line port"
      ],
      "// comment": "S1-1 LTP",
      "tp-id": "1",
      "ietf-te-topology:te-tp-id": 1,
      "ietf-te-topology:te": {
        "admin-status": "up",
        "oper-status": "up",
        "// ietf-otn-topology:supported-payload-types":
          "__DISCUSS__",
        "// _OBSOLETE_ ietf-otn-topology:protocol-type":
          "ietf-transport-types:prot-OTU4",
        "// _OBSOLETE_ ietf-otn-topology:adaptation-type":
          "ODU"
      }
    }
  ],
},

```

```

    {
      "// __DESCRIPTION__:__LTP__": [
        "S1-2 LTP",
        "connected to S3-2",
        "unnumberd/ifIndex: 2",
        "OTU-4",
        "line port"
      ],
      "// comment": "S1-2 LTP",
      "tp-id": "2",
      "ietf-te-topology:te-tp-id": 2,
      "ietf-te-topology:te": {
        "admin-status": "up",
        "oper-status": "up",
        "// ietf-otn-topology:supported-payload-types":
          "__DISCUSS__",
        "// _OBSOLETE_ ietf-otn-topology:protocol-type":
          "ietf-transport-types:prot-OTU4",
        "// _OBSOLETE_ ietf-otn-topology:adaptation-type":
          "ODU"
      }
    }
  ],
},
{
  "// __NODE__:__DESCRIPTION__": [
    "S2",
    "10.0.0.2",
    "ADM"
  ],
  "node-id": "10.0.0.2",
  "ietf-te-topology:te-node-id": "10.0.0.2",
  "ietf-te-topology:te": {
    "// comment": "TO BE COMPLETED",
    "oper-status": "up",
    "te-node-attributes": {
      "name": "S2-robinson_park_ring-access",
      "admin-status": "up",
      "// __DISCUSS__ domain-id": 65001
    },
    "// __DISCUSS__ tunnel-termination-point": []
  },
  "ietf-network-topology:termination-point": [
    {
      "// __DESCRIPTION__:__LTP__": [

```

```

        "S2-1 LTP",
        "connected to (C-R4)",
        "unnumberd/ifIndex: 1",
        "OTU-2",
        "tributary port"
    ],
    "// comment": "S2-1 LTP",
    "tp-id": "1",
    "ietf-te-topology:te-tp-id": 1,
    "ietf-te-topology:te": {
        "// _OBSOLETE_ ietf-otn-topology:client-facing": [
            null
        ],
        "admin-status": "up",
        "oper-status": "up",
        "// ietf-otn-topology:supported-payload-types":
            "__DISCUSS__",
        "// _OBSOLETE_ ietf-otn-topology:protocol-type":
            "ietf-transport-types:prot-OTU2",
        "// _OBSOLETE_ ietf-otn-topology:adaptation-type":
            "ODU"
    }
},
{
    "// __DESCRIPTION__:__LTP__": [
        "S2-2 LTP",
        "connected to S1-1",
        "unnumberd/ifIndex: 2",
        "OTU-4",
        "line port"
    ],
    "// comment": "S2-2 LTP",
    "tp-id": "2",
    "ietf-te-topology:te-tp-id": 2,
    "ietf-te-topology:te": {
        "admin-status": "up",
        "oper-status": "up",
        "// ietf-otn-topology:supported-payload-types":
            "__DISCUSS__",
        "// _OBSOLETE_ ietf-otn-topology:protocol-type":
            "ietf-transport-types:prot-OTU4",
        "// _OBSOLETE_ ietf-otn-topology:adaptation-type":
            "ODU"
    }
},

```

```

    {
      "/* __DESCRIPTION__:__LTP__": [
        "S2-3 LTP",
        "connected to S8-2",
        "unnumberd/ifIndex: 3",
        "OTU-4",
        "line port"
      ],
      "/* comment": "S2-3 LTP",
      "tp-id": "3",
      "ietf-te-topology:te-tp-id": 3,
      "ietf-te-topology:te": {
        "admin-status": "up",
        "oper-status": "up",
        "/* ietf-otn-topology:supported-payload-types":
          "__DISCUSS__",
        "/* __OBSOLETE__ ietf-otn-topology:protocol-type":
          "ietf-transport-types:prot-OTU4",
        "/* __OBSOLETE__ ietf-otn-topology:adaptation-type":
          "ODU"
      }
    }
  ],
},
{
  "/* __NODE__:__DESCRIPTION__": [
    "S7",
    "10.0.0.7",
    "ADM"
  ],
  "node-id": "10.0.0.7",
  "ietf-te-topology:te-node-id": "10.0.0.7",
  "ietf-te-topology:te": {
    "/* comment": "TO BE COMPLETED",
    "oper-status": "up",
    "te-node-attributes": {
      "name": "S7-east_end_ring-gateway",
      "admin-status": "up",
      "/* __DISCUSS__ domain-id": 65001
    },
    "/* __DISCUSS__ tunnel-termination-point": []
  },
  "ietf-network-topology:termination-point": [
    {
      "/* __DESCRIPTION__:__LTP__": [

```

```

        "S7-1 LTP",
        "connected to S5-3",
        "unnumberd/ifIndex: 1",
        "OTU-4",
        "line port"
    ],
    "// comment": "S7-1 LTP",
    "tp-id": "1",
    "ietf-te-topology:te-tp-id": 1,
    "ietf-te-topology:te": {
        "admin-status": "up",
        "oper-status": "up",
        "// ietf-otn-topology:supported-payload-types":
            "__DISCUSS__",
        "// _OBSOLETE_ ietf-otn-topology:protocol-type":
            "ietf-transport-types:prot-OTU4",
        "// _OBSOLETE_ ietf-otn-topology:adaptation-type":
            "ODU"
    }
},
{
    "// __DESCRIPTION__:__LTP__": [
        "S7-2 LTP",
        "connected to S6-4",
        "unnumberd/ifIndex: 2",
        "OTU-4",
        "line port"
    ],
    "// comment": "S7-2 LTP",
    "tp-id": "2",
    "ietf-te-topology:te-tp-id": 2,
    "ietf-te-topology:te": {
        "admin-status": "up",
        "oper-status": "up",
        "// ietf-otn-topology:supported-payload-types":
            "__DISCUSS__",
        "// _OBSOLETE_ ietf-otn-topology:protocol-type":
            "ietf-transport-types:prot-OTU4",
        "// _OBSOLETE_ ietf-otn-topology:adaptation-type":
            "ODU"
    }
},
{
    "// __DESCRIPTION__:__LTP__": [
        "S7-3 LTP",

```

```

        "connected to S8-4",
        "unnumberd/ifIndex: 3",
        "OTU-4",
        "line port"
    ],
    "// comment": "S7-3 LTP",
    "tp-id": "3",
    "ietf-te-topology:te-tp-id": 3,
    "ietf-te-topology:te": {
        "admin-status": "up",
        "oper-status": "up",
        "// ietf-otn-topology:supported-payload-types":
            "__DISCUSS__",
        "// _OBSOLETE_ ietf-otn-topology:protocol-type":
            "ietf-transport-types:prot-OTU4",
        "// _OBSOLETE_ ietf-otn-topology:adaptation-type":
            "ODU"
    }
}
]
},
{
    "// __NODE__:__DESCRIPTION__": [
        "S8",
        "10.0.0.8",
        "ADM"
    ],
    "node-id": "10.0.0.8",
    "ietf-te-topology:te-node-id": "10.0.0.8",
    "ietf-te-topology:te": {
        "// comment": "TO BE COMPLETED",
        "oper-status": "up",
        "te-node-attributes": {
            "name": "S8-metro_main_ring-access",
            "admin-status": "up",
            "// __DISCUSS__ domain-id": 65001
        },
        "// __DISCUSS__ tunnel-termination-point": []
    },
    "ietf-network-topology:termination-point": [
        {
            "// __DESCRIPTION__:__LTP__": [
                "S8-1 LTP",
                "connected to (C-R5)",
                "unnumberd/ifIndex: 1",

```



```
        "OTU-2",
        "tributary port"
    ],
    "// comment": "S8-1 LTP",
    "tp-id": "1",
    "ietf-te-topology:te-tp-id": 1,
    "ietf-te-topology:te": {
        "// _OBSOLETE_ ietf-otn-topology:client-facing": [
            null
        ],
        "admin-status": "up",
        "oper-status": "up",
        "// ietf-otn-topology:supported-payload-types":
            "__DISCUSS__",
        "// _OBSOLETE_ ietf-otn-topology:protocol-type":
            "ietf-transport-types:prot-OTU2",
        "// _OBSOLETE_ ietf-otn-topology:adaptation-type":
            "ODU"
    }
},
{
    "// __DESCRIPTION__:__LTP__": [
        "S8-2 LTP",
        "connected to S2-3",
        "unnumberd/ifIndex: 2",
        "OTU-4",
        "line port"
    ],
    "// comment": "S8-2 LTP",
    "tp-id": "2",
    "ietf-te-topology:te-tp-id": 2,
    "ietf-te-topology:te": {
        "admin-status": "up",
        "oper-status": "up",
        "// ietf-otn-topology:supported-payload-types":
            "__DISCUSS__",
        "// _OBSOLETE_ ietf-otn-topology:protocol-type":
            "ietf-transport-types:prot-OTU4",
        "// _OBSOLETE_ ietf-otn-topology:adaptation-type":
            "ODU"
    }
},
{
    "// __DESCRIPTION__:__LTP__": [
        "S8-3 LTP",
```

```

        "connected to S4-2",
        "unnumberd/ifIndex: 3",
        "OTU-4",
        "line port"
    ],
    "// comment": "S8-3 LTP",
    "tp-id": "3",
    "ietf-te-topology:te-tp-id": 3,
    "ietf-te-topology:te": {
        "admin-status": "up",
        "oper-status": "up",
        "// ietf-otn-topology:supported-payload-types":
            "__DISCUSS__",
        "// _OBSOLETE_ ietf-otn-topology:protocol-type":
            "ietf-transport-types:prot-OTU4",
        "// _OBSOLETE_ ietf-otn-topology:adaptation-type":
            "ODU"
    }
},
{
    "// __DESCRIPTION__:__LTP__": [
        "S8-4 LTP",
        "connected to S7-3",
        "unnumberd/ifIndex: 4",
        "OTU-4",
        "line port"
    ],
    "// comment": "S8-4 LTP",
    "tp-id": "4",
    "ietf-te-topology:te-tp-id": 4,
    "ietf-te-topology:te": {
        "admin-status": "up",
        "oper-status": "up",
        "// ietf-otn-topology:supported-payload-types":
            "__DISCUSS__",
        "// _OBSOLETE_ ietf-otn-topology:protocol-type":
            "ietf-transport-types:prot-OTU4",
        "// _OBSOLETE_ ietf-otn-topology:adaptation-type":
            "ODU"
    }
}
]
},
{
    "// __NODE__:__DESCRIPTION__": [

```

```

    "S5",
    "10.0.0.5",
    "ADM"
  ],
  "node-id": "10.0.0.5",
  "ietf-te-topology:te-node-id": "10.0.0.5",
  "ietf-te-topology:te": {
    "// comment": "TO BE COMPLETED",
    "oper-status": "up",
    "te-node-attributes": {
      "name": "S5-east_end_ring-gateway",
      "admin-status": "up",
      "// __DISCUSS__ domain-id": 65001
    },
    "// __DISCUSS__ tunnel-termination-point": []
  },
  "ietf-network-topology:termination-point": [
    {
      "// __DESCRIPTION__:__LTP__": [
        "S5-1 LTP",
        "connected to S3-4",
        "unnumberd/ifIndex: 1",
        "OTU-4",
        "line port"
      ],
      "// comment": "S5-1 LTP",
      "tp-id": "1",
      "ietf-te-topology:te-tp-id": 1,
      "ietf-te-topology:te": {
        "admin-status": "up",
        "oper-status": "up",
        "// ietf-otn-topology:supported-payload-types":
          "__DISCUSS__",
        "// _OBSOLETE_ ietf-otn-topology:protocol-type":
          "ietf-transport-types:prot-OTU4",
        "// _OBSOLETE_ ietf-otn-topology:adaptation-type":
          "ODU"
      }
    },
    {
      "// __DESCRIPTION__:__LTP__": [
        "S5-2 LTP",
        "connected to S6-3",
        "unnumberd/ifIndex: 2",
        "OTU-4",

```

```

        "line port"
      ],
      "// comment": "S5-2 LTP",
      "tp-id": "2",
      "ietf-te-topology:te-tp-id": 2,
      "ietf-te-topology:te": {
        "admin-status": "up",
        "oper-status": "up",
        "// ietf-otn-topology:supported-payload-types":
          "__DISCUSS__",
        "// _OBSOLETE_ ietf-otn-topology:protocol-type":
          "ietf-transport-types:prot-OTU4",
        "// _OBSOLETE_ ietf-otn-topology:adaptation-type":
          "ODU"
      }
    },
    {
      "// __DESCRIPTION__:__LTP__": [
        "S5-3 LTP",
        "connected to S7-1",
        "unnumberd/ifIndex: 3",
        "OTU-4",
        "line port"
      ],
      "// comment": "S5-3 LTP",
      "tp-id": "3",
      "ietf-te-topology:te-tp-id": 3,
      "ietf-te-topology:te": {
        "admin-status": "up",
        "oper-status": "up",
        "// ietf-otn-topology:supported-payload-types":
          "__DISCUSS__",
        "// _OBSOLETE_ ietf-otn-topology:protocol-type":
          "ietf-transport-types:prot-OTU4",
        "// _OBSOLETE_ ietf-otn-topology:adaptation-type":
          "ODU"
      }
    }
  ]
},
{
  "// __NODE__:__DESCRIPTION__": [
    "S6",
    "10.0.0.6",
    "ADM"
  ]
}

```

```

],
"node-id": "10.0.0.6",
"ietf-te-topology:te-node-id": "10.0.0.6",
"ietf-te-topology:te": {
  "// comment": "TO BE COMPLETED",
  "oper-status": "up",
  "te-node-attributes": {
    "name": "S6-east_end_ring-access",
    "admin-status": "up",
    "// __DISCUSS__ domain-id": 65001
  },
  "// __DISCUSS__ tunnel-termination-point": []
},
"ietf-network-topology:termination-point": [
  {
    "// __DESCRIPTION__:__LTP__": [
      "S6-1 LTP",
      "connected to (C-R2)",
      "unnumberd/ifIndex: 1",
      "OTU-2",
      "tributary port"
    ],
    "// comment": "S6-1 LTP",
    "tp-id": "1",
    "ietf-te-topology:te-tp-id": 1,
    "ietf-te-topology:te": {
      "admin-status": "up",
      "oper-status": "up",
      "// _OBSOLETE_ ietf-otn-topology:client-facing": [
        null
      ],
      "// ietf-otn-topology:supported-payload-types":
        "__DISCUSS__",
      "// _OBSOLETE_ ietf-otn-topology:protocol-type":
        "ietf-transport-types:prot-OTU2",
      "// _OBSOLETE_ ietf-otn-topology:adaptation-type":
        "ODU"
    }
  },
  {
    "// __DESCRIPTION__:__LTP__": [
      "S6-2 LTP",
      "connected to (C-R3)",
      "unnumberd/ifIndex: 2",
      "OTU-2",

```

```

        "tributary port"
    ],
    "// comment": "S6-2 LTP",
    "tp-id": "2",
    "ietf-te-topology:te-tp-id": 2,
    "ietf-te-topology:te": {
        "admin-status": "up",
        "oper-status": "up",
        "// _OBSOLETE_ ietf-otn-topology:client-facing": [
            null
        ],
    },
    "// ietf-otn-topology:supported-payload-types":
        "__DISCUSS__",
    "// _OBSOLETE_ ietf-otn-topology:protocol-type":
        "ietf-transport-types:prot-OTU2",
    "// _OBSOLETE_ ietf-otn-topology:adaptation-type":
        "ODU"
    }
},
{
    "// __DESCRIPTION__:__LTP__": [
        "S6-3 LTP",
        "connected to S5-2",
        "unnumberd/ifIndex: 3",
        "OTU-4",
        "line port"
    ],
    "// comment": "S6-3 LTP",
    "tp-id": "3",
    "ietf-te-topology:te-tp-id": 3,
    "ietf-te-topology:te": {
        "admin-status": "up",
        "oper-status": "up",
        "// ietf-otn-topology:supported-payload-types":
            "__DISCUSS__",
        "// _OBSOLETE_ ietf-otn-topology:protocol-type":
            "ietf-transport-types:prot-OTU4",
        "// _OBSOLETE_ ietf-otn-topology:adaptation-type":
            "ODU"
    }
},
{
    "// __DESCRIPTION__:__LTP__": [
        "S6-4 LTP",
        "connected to S7-2",

```

```

        "unnumberd/ifIndex: 4",
        "OTU-4",
        "line port"
    ],
    "// comment": "S6-4 LTP",
    "tp-id": "4",
    "ietf-te-topology:te-tp-id": 4,
    "ietf-te-topology:te": {
        "admin-status": "up",
        "oper-status": "up",
        "// ietf-otn-topology:supported-payload-types":
            "__DISCUSS__",
        "// _OBSOLETE_ ietf-otn-topology:protocol-type":
            "ietf-transport-types:prot-OTU4",
        "// _OBSOLETE_ ietf-otn-topology:adaptation-type":
            "ODU"
    }
}
]
}
],
"// ietf-network-topology:link":
    "Access links to be added in a future update.",
"ietf-network-topology:link": [
    {
        "// __DESCRIPTION__:__LINK__": [
            "Link from S1-2 to S3-2",
            "internal link"
        ],
    },
    "// link-id": "S1, S1-2, S3, S3-2",
    "link-id": "10.0.0.1,2,10.0.0.3,2",
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "// comment": [
                "external-domain container",
                "not present for internal links"
            ],
        },
        "name": "Link between S1 and S3",
        "admin-status": "up",
        "interface-switching-capability": [
            {
                "switching-capability":
                    "ietf-te-types:switching-otn",
            }
        ]
    }
]

```

```
    "encoding": "ietf-te-types:lsp-encoding-oduk",
    "max-lsp-bandwidth": [
      {
        "priority": 0,
        "te-bandwidth": {
          "// _OBSOLETE_ otn": [
            "_WARNING_ : technology specific bandwidth definition",
            {
              "rate-type": "ietf-te-types:odu2",
              "counter": 1
            }
          ]
        }
      },
      {
        "priority": 1,
        "te-bandwidth": {
          "// _OBSOLETE_ otn": [
            "_WARNING_ : technology specific bandwidth definition",
            {
              "rate-type": "ietf-te-types:odu2",
              "counter": 1
            }
          ]
        }
      },
      {
        "priority": 2,
        "te-bandwidth": {
          "// _OBSOLETE_ otn": [
            "_WARNING_ : technology specific bandwidth definition",
            {
              "rate-type": "ietf-te-types:odu2",
              "counter": 1
            }
          ]
        }
      },
      {
        "priority": 3,
        "te-bandwidth": {
          "// _OBSOLETE_ otn": [
            "_WARNING_ : technology specific bandwidth definition",
            {
              "rate-type": "ietf-te-types:odu2",
```



```
        "counter": 1
      }
    ]
  }
},
{
  "priority": 4,
  "te-bandwidth": {
    "// _OBSOLETE_ otn": [
      "_WARNING_ : technology specific bandwidth definition",
      {
        "rate-type": "ietf-te-types:odu2",
        "counter": 1
      }
    ]
  }
},
{
  "priority": 5,
  "te-bandwidth": {
    "// _OBSOLETE_ otn": [
      "_WARNING_ : technology specific bandwidth definition",
      {
        "rate-type": "ietf-te-types:odu2",
        "counter": 1
      }
    ]
  }
},
{
  "priority": 6,
  "te-bandwidth": {
    "// _OBSOLETE_ otn": [
      "_WARNING_ : technology specific bandwidth definition",
      {
        "rate-type": "ietf-te-types:odu2",
        "counter": 1
      }
    ]
  }
},
{
  "priority": 7,
  "te-bandwidth": {
    "// _OBSOLETE_ otn": [
```

```

"_WARNING_ : technology specific bandwidth definition",
    {
        "rate-type": "ietf-te-types:odu2",
        "counter": 1
    }
]
}
}
}
}
}
}
},
{
    "// __DESCRIPTION__:__LINK__": [
        "Link from S5-2 to S6-3",
        "internal link"
    ],
    "link-id": "S5, S5-2, S6, S6-3",
    "link-id": "10.0.0.5,2,10.0.0.6,3",
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "name": "Link between S5 and S6",
            "admin-status": "up"
        }
    }
},
{
    "// __DESCRIPTION__:__LINK__": [
        "Link from S3-3 to S4-1",
        "internal link"
    ],
    "link-id": "S3, S3-3, S4, S4-1",
    "link-id": "10.0.0.3,3,10.0.0.4,1",
    "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
            "access-type": "point-to-point",
            "name": "Link between S3 and S4",
            "admin-status": "up"
        }
    }
}
}

```

```
},
{
  "// __DESCRIPTION__:__LINK__": [
    "Link from S5-3 to S7-1",
    "internal link"
  ],
  "// link-id": "S5, S5-3, S7, S7-1",
  "link-id": "10.0.0.5,3,10.0.0.7,1",
  "ietf-te-topology:te": {
    "oper-status": "up",
    "te-link-attributes": {
      "access-type": "point-to-point",
      "name": "Link between S5 and S7",
      "admin-status": "up"
    }
  }
},
{
  "// __DESCRIPTION__:__LINK__": [
    "Link from S3-4 to S5-1",
    "internal link"
  ],
  "// link-id": "S3, S3-4, S5, S5-1",
  "link-id": "10.0.0.3,4,10.0.0.5,1",
  "ietf-te-topology:te": {
    "oper-status": "up",
    "te-link-attributes": {
      "access-type": "point-to-point",
      "name": "Link between S3 and S5",
      "admin-status": "up"
    }
  }
},
{
  "// __DESCRIPTION__:__LINK__": [
    "Link from S4-2 to S8-3",
    "internal link"
  ],
  "// link-id": "S4, S4-2, S8, S8-3",
  "link-id": "10.0.0.4,2,10.0.0.8,3",
  "ietf-te-topology:te": {
    "oper-status": "up",
    "te-link-attributes": {
      "access-type": "point-to-point",
      "name": "Link between S4 and S8",
```

```
        "admin-status": "up"
      }
    }
  },
  {
    "// __DESCRIPTION__:__LINK__": [
      "Link from S2-3 to S8-2",
      "internal link"
    ],
    "// link-id": "S2, S2-3, S8, S8-2",
    "link-id": "10.0.0.2,3,10.0.0.8,2",
    "ietf-te-topology:te": {
      "oper-status": "up",
      "te-link-attributes": {
        "access-type": "point-to-point",
        "name": "Link between S2 and S8",
        "admin-status": "up"
      }
    }
  }
},
{
  "// __DESCRIPTION__:__LINK__": [
    "Link from S8-2 to S2-3",
    "internal link"
  ],
  "// link-id": "S8, S8-2, S2, S2-3",
  "link-id": "10.0.0.8,2,10.0.0.2,3",
  "ietf-te-topology:te": {
    "oper-status": "up",
    "te-link-attributes": {
      "access-type": "point-to-point",
      "name": "Link between S8 and S2",
      "admin-status": "up"
    }
  }
},
{
  "// __DESCRIPTION__:__LINK__": [
    "Link from S3-2 to S1-2",
    "internal link"
  ],
  "// link-id": "S3, S3-2, S1, S1-2",
  "link-id": "10.0.0.3,2,10.0.0.1,2",
  "ietf-te-topology:te": {
    "oper-status": "up",
```

```

        "te-link-attributes": {
            "access-type": "point-to-point",
            "name": "Link between S3 and S1",
            "admin-status": "up"
        }
    },
    {
        "/* __DESCRIPTION__:__LINK__": [
            "Link from S7-1 to S5-3",
            "internal link"
        ],
        "/* link-id": "S7, S7-1, S5, S5-3",
        "link-id": "10.0.0.7,1,10.0.0.5,3",
        "ietf-te-topology:te": {
            "oper-status": "up",
            "te-link-attributes": {
                "access-type": "point-to-point",
                "name": "Link between S7 and S5",
                "admin-status": "up"
            }
        }
    },
    {
        "/* __DESCRIPTION__:__LINK__": [
            "Link from S8-3 to S4-2",
            "internal link"
        ],
        "/* link-id": "S8, S8-3, S4, S4-2",
        "link-id": "10.0.0.8,3,10.0.0.4,2",
        "ietf-te-topology:te": {
            "oper-status": "up",
            "te-link-attributes": {
                "access-type": "point-to-point",
                "name": "Link between S8 and S4",
                "admin-status": "up"
            }
        }
    },
    {
        "/* __DESCRIPTION__:__LINK__": [
            "Link from S5-1 to S3-4",
            "internal link"
        ],
        "/* link-id": "S5, S5-1, S3, S3-4",

```

```
"link-id": "10.0.0.5,1,10.0.0.3,4",
"ietf-te-topology:te": {
  "oper-status": "up",
  "te-link-attributes": {
    "access-type": "point-to-point",
    "name": "Link between S5 and S3",
    "admin-status": "up"
  }
}
},
{
  "// __DESCRIPTION__:__LINK__": [
    "Link from S2-2 to S1-1",
    "internal link"
  ],
  "// link-id": "S2, S2-2, S1, S1-1",
  "link-id": "10.0.0.2,2,10.0.0.1,1",
  "ietf-te-topology:te": {
    "oper-status": "up",
    "te-link-attributes": {
      "access-type": "point-to-point",
      "name": "Link between S2 and S1",
      "admin-status": "up"
    }
  }
}
},
{
  "// __DESCRIPTION__:__LINK__": [
    "Link from S7-2 to S6-4",
    "internal link"
  ],
  "// link-id": "S7, S7-2, S6, S6-4",
  "link-id": "10.0.0.7,2,10.0.0.6,4",
  "ietf-te-topology:te": {
    "oper-status": "up",
    "te-link-attributes": {
      "access-type": "point-to-point",
      "name": "Link between S7 and S6",
      "admin-status": "up"
    }
  }
}
},
{
  "// __DESCRIPTION__:__LINK__": [
    "Link from S8-4 to S7-3",
```

```
        "internal link"
      ],
      "// link-id": "S8, S8-4, S7, S7-3",
      "link-id": "10.0.0.8,4,10.0.0.7,3",
      "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
          "access-type": "point-to-point",
          "name": "Link between S8 and S7",
          "admin-status": "up"
        }
      }
    },
    {
      "// __DESCRIPTION__:__LINK__": [
        "Link from S6-3 to S5-2",
        "internal link"
      ],
      "// link-id": "S6, S6-3, S5, S5-2",
      "link-id": "10.0.0.6,3,10.0.0.5,2",
      "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
          "access-type": "point-to-point",
          "name": "Link between S6 and S5",
          "admin-status": "up"
        }
      }
    },
    {
      "// __DESCRIPTION__:__LINK__": [
        "Link from S4-1 to S3-3",
        "internal link"
      ],
      "// link-id": "S4, S4-1, S3, S3-3",
      "link-id": "10.0.0.4,1,10.0.0.3,3",
      "ietf-te-topology:te": {
        "oper-status": "up",
        "te-link-attributes": {
          "access-type": "point-to-point",
          "name": "Link between S4 and S3",
          "admin-status": "up"
        }
      }
    }
  ],
}
```

```

{
  "// __DESCRIPTION__:__LINK__": [
    "Link from S6-4 to S7-2",
    "internal link"
  ],
  "// link-id": "S6, S6-4, S7, S7-2",
  "link-id": "10.0.0.6,4,10.0.0.7,2",
  "ietf-te-topology:te": {
    "oper-status": "up",
    "te-link-attributes": {
      "access-type": "point-to-point",
      "name": "Link between S6 and S7",
      "admin-status": "up"
    }
  }
},
{
  "// __DESCRIPTION__:__LINK__": [
    "Link from S1-1 to S2-2",
    "internal link"
  ],
  "// link-id": "S1, S1-1, S2, S2-2",
  "link-id": "10.0.0.1,1,10.0.0.2,2",
  "ietf-te-topology:te": {
    "oper-status": "up",
    "te-link-attributes": {
      "access-type": "point-to-point",
      "name": "Link between S1 and S2",
      "admin-status": "up"
    }
  }
},
{
  "// __DESCRIPTION__:__LINK__": [
    "Link from S7-3 to S8-4",
    "internal link"
  ],
  "// link-id": "S7, S7-3, S8, S8-4",
  "link-id": "10.0.0.7,3,10.0.0.8,4",
  "ietf-te-topology:te": {
    "oper-status": "up",
    "te-link-attributes": {
      "access-type": "point-to-point",
      "name": "Link between S7 and S8",
      "admin-status": "up"
    }
  }
}

```



```

    }
  }
]
}
]
}
}

```

B.2. JSON Examples for Service Configuration

B.2.1. JSON Code: mpi1-odu2-service-config.json

```

{
  "// __TITLE__": "ODU2 Service Configuration @ MPI1",
  "// __LAST_UPDATE__": "July 2, 2018",
  "// __RESTCONF_OPERATION__": {
    "operation": "PUT",
    "url": "http://{{PNC1-ADDR}}/restconf/data/ietf-te:te"
  },
  "// __REFERENCE_DRAFTS__": {
    "ietf-te-types@2018-06-12": "draft-ietf-teas-yang-te-15",
    "ietf-routing-types@2017-12-04": "rfc8294",
    "ietf-te@2018-06-12": "draft-ietf-teas-yang-te-15",
    "ietf-otn-types@2018-06-07":
      "draft-ietf-ccamp-otn-tunnel-model-02",
    "ietf-otn-tunnel@2018-06-07":
      "draft-ietf-ccamp-otn-tunnel-model-02"
  },
  "// __MISSING_ATTRIBUTES__": true,
  "ietf-te:te": {
    "tunnels": {
      "tunnel": [
        {
          "name": "mpi1-odu2-service",
          "// identifier": "ODU2-SERVICE-TUNNEL-ID @ MPI1",
          "identifier": 1,
          "description":
            "ODU2 Service implemented by ODU2 OTN Tunnel Segment @ MPI1",
          "// encoding and switching-type": "ODU",
          "encoding": "ietf-te-types:lsp-encoding-oduk ",
          "switching-type": "ietf-te-types:switching-otn",
          "// source": "None: transit tunnel segment",
          "// destination": "None: transit tunnel segment",
          "// src-tp-id": "None: transit tunnel segment",

```

```
"// dst-tp-id": "None: transit tunnel segment",
"// __ ACTION __ ietf-otn-tunnel:payload-treatment": [
  "This attribute should be removed in the next otn-tunnel",
  " model update"
],
"ietf-otn-tunnel:src-client-signal":
  "ietf-otn-types:client-signal-ODU2",
"// __ ACTION __ src-tpn": [
  "This attribute should be removed in the next otn-tunnel",
  " model update"
],
"// __ ACTION __ src-tsg": [
  "This attribute should be removed in the next otn-tunnel",
  " model update"
],
"// __ ACTION __ src-tributary-slot-count": [
  "This attribute should be removed in the next otn-tunnel",
  " model update"
],
"// __ ACTION __ src-tributary-slots": [
  "This attribute should be removed in the next otn-tunnel",
  " model update"
],
"ietf-otn-tunnel:dst-client-signal":
  "ietf-otn-types:client-signal-ODU2",
"// __ ACTION __ dst-tpn": [
  "This attribute should be removed in the next otn-tunnel",
  " model update"
],
"// __ ACTION __ dst-tsg": [
  "This attribute should be removed in the next otn-tunnel",
  " model update"
],
"// __ ACTION __ dst-tributary-slot-count": [
  "This attribute should be removed in the next otn-tunnel",
  " model update"
],
"// __ ACTION __ dst-tributary-slots": [
  "This attribute should be removed in the next otn-tunnel",
  " model update"
],
"bidirectional": true,
"// __ DEFAULT __ protection": {
  "// __ DEFAULT __ enable": false
},
```

```

    "// __ DEFAULT __ restoration": {
      "// __ DEFAULT __ enable": false
    },
    "// __ DISCUSS __ te-topology-identifier": [
      "Need to add the te-topology-identifier",
      "information. Waiting for updates to topology identifiers"
    ],
    "te-bandwidth": {
      "ietf-otn-tunnel:odu-type": "ietf-otn-types:prot-ODU2"
    },
    "// hierarchical-link":
      "Not to be specified: transit tunnel segment",
    "p2p-primary-paths": {
      "p2p-primary-path": [
        {
          "name": "mpi1-odu2-tunnel-primary-path",
          "// __ DISCUSS __ path-scope": [
            "Need to align the model based on the on-going",
            " disucssion of the related open issue"
          ],
          "path-scope": "ietf-te-types:path-scope-segment",
          "// te-bandwidth": [
            "None: only the tunnel bandwidth needs to be specified",
            " in transport applications"
          ],
          "explicit-route-objects": {
            "route-object-include-exclude": [
              {
                "// comment":
                  "Tunnel hand-off OTU2 ingress interface (S3-1)",
                "index": 1,
                "explicit-route-usage":
                  "ietf-te-types:route-include-ero",
                "num-unnum-hop": {
                  "// node-id": "S3-NODE-ID",
                  "node-id": "10.0.0.3",
                  "// link-tp-id": "S3-1-LTP-ID",
                  "link-tp-id": 1,
                  "hop-type": "STRICT",
                  "direction": "INCOMING"
                }
              }
            ],
            {
              "// comment": [
                "Tunnel hand-off ODU2 ingress label (ODU2 over",

```

```

    " OTU2) at S3-1"
  ],
  "index": 2,
  "explicit-route-usage":
    "ietf-te-types:route-include-ero",
  "label-hop": {
    "te-label": {
      "// __ DISCUSS __ odu-label": [
        "How are HO-ODU (ODUk voer OTUk) label",
        " represented?"
      ],
      "// __ ACTION __ direction":
        "Check with TE Tunnel authors",
      "direction": "FORWARD "
    }
  }
},
{
  "// comment":
    "Tunnel hand-off OTU4 egress interface (S2-1)",
  "index": 3,
  "explicit-route-usage":
    "ietf-te-types:route-include-ero",
  "num-unnum-hop": {
    "// node-id": "S2-NODE-ID",
    "node-id": "10.0.0.2",
    "// link-tp-id": "S2-1-LTP-ID",
    "link-tp-id": 1,
    "hop-type": "STRICT",
    "direction": "OUTGOING"
  }
},
{
  "// comment": [
    "Tunnel hand-off ODU2 egress label (ODU2 over",
    " OTU4) at S2-1"
  ],
  "index": 4,
  "explicit-route-usage":
    "ietf-te-types:route-include-ero",
  "label-hop": {
    "te-label": {
      "ietf-otn-tunnel:tpn": 1,
      "ietf-otn-tunnel:tsg":
        "ietf-otn-types:tsg-1.25G",

```

[illegible]

B.2.2. JSON Code: `mpi1-odu2-tunnel-config.json`

```
{
  "/// __TITLE__": "ODU2 Tunnel Configuration @ MPI1",
  "/// __LAST_UPDATE__": "July 2, 2018",
  "/// __RESTCONF_OPERATION__": {
    "operation": "PUT",
    "url": "http://{{PNC1-ADDR}}/restconf/data/ietf-te:te"
  },
  "/// __REFERENCE_DRAFTS__": {
    "ietf-te-types@2018-06-12": "draft-ietf-teas-yang-te-15",
    "ietf-routing-types@2017-12-04": "rfc8294",
    "ietf-te@2018-06-12": "draft-ietf-teas-yang-te-15",
    "ietf-otn-types@2018-06-07":
      "draft-ietf-ccamp-otn-tunnel-model-02",
    "ietf-otn-tunnel@2018-06-07":
      "draft-ietf-ccamp-otn-tunnel-model-02"
  },
  "/// __MISSING_ATTRIBUTES__": true,
  "ietf-te:te": {
    "tunnels": {
      "tunnel": [
        {
          "name": "mpi1-odu2-tunnel",
          "/// identifier": "ODU2-TUNNEL-ID @ MPI1",
          "identifier": 2,
          "description":

```

```

    "TNBI Example for an ODU2 Head Tunnel Segment @ MPI1",
    "// encoding and switching-type": "ODU",
    "encoding": "ietf-te-types:lsp-encoding-oduk ",
    "switching-type": "ietf-te-types:switching-otn",
    "source": "10.0.0.3",
    "// destination": "None: head tunnel segment",
    "src-tp-id": "AAAB",
    "// dst-tp-id": "None: head tunnel segment",
    "ietf-otn-tunnel:src-client-signal":
    "ietf-otn-types:client-signal-ODU2",
    "ietf-otn-tunnel:dst-client-signal":
    "ietf-otn-types:client-signal-ODU2",
    "bidirectional": true,
    "// __ DEFAULT __ protection": {
    "    // __ DEFAULT __ enable": false
    },
    "// __ DEFAULT __ restoration": {
    "    // __ DEFAULT __ enable": false
    },
    "// __ DISCUSS __ te-topology-identifier":
    "Need to add the te-topology-identifier information",
    "te-bandwidth": {
    "ietf-otn-tunnel:odu-type": "ietf-otn-types:prot-ODU2"
    },
    "// __ DISCUSS __ hierarchical-link":
    "Optional: tunnel supports service, not link in the client layer",
    "p2p-primary-paths": {
    "p2p-primary-path": [
    {
    "name": "mpi1-odu2-tunnel-primary-path",
    "// __ DISCUSS __ path-scope":
    "Need to align the model",
    "path-scope": "ietf-te-types:path-scope-segment",
    "// te-bandwidth":
    "None: only the tunnel bandwidth needed in transport",
    "explicit-route-objects": {
    "route-object-include-exclude": [
    {
    "    // comment": "Tunnel TTP in node S3",
    "index": 1,
    "explicit-route-usage":
    "ietf-te-types:route-include-ero",
    "num-unnum-hop": {
    "    // node-id": "S3-NODE-ID",
    "node-id": "10.0.0.3",

```

```

        "hop-type": "STRICT",
        "// __ ACTION __ direction":
            "Check with TE Tunnel authors",
        "direction": "OUTGOING"
    }
},
{
    "// comment":
    "Tunnel hand-off OTU4 egress interface (S2-1)",
    "index": 2,
    "explicit-route-usage":
        "ietf-te-types:route-include-ero",
    "num-unnum-hop": {
        "// node-id": "S2-NODE-ID",
        "node-id": "10.0.0.2",
        "// link-tp-id": "S2-1-LTP-ID",
        "link-tp-id": 1,
        "hop-type": "STRICT",
        "direction": "OUTGOING"
    }
},
{
    "// comment":
    "Tunnel hand-off ODU2 egress label (ODU2 over OTU4) at S2-1",
    "index": 3,
    "explicit-route-usage":
        "ietf-te-types:route-include-ero",
    "label-hop": {
        "te-label": {
            "ietf-otn-tunnel:tpn": 2,
            "ietf-otn-tunnel:tsg":
                "ietf-otn-types:tsg-1.25G",
            "ietf-otn-tunnel:ts-list": "9-16",
            "// __ ACTION __ direction":
                "Check with TE Tunnel authors",
            "direction": "FORWARD "
        }
    }
}
]
}
}
}
}
}
```

```

    ]
  }
}
}

```

B.2.3. JSON Code: mpi1-epl-service-config.json

```

{
  "// __TITLE__": "EPL Configuration @ MPI1",
  "// __LAST_UPDATE__": "July 2, 2018",
  "// __RESTCONF_OPERATION__": {
    "operation": "PUT",
    "url":
"http://{{PNC1}}/restconf/data/ietf-trans-client-service:eth-t-svc"
  },
  "// __REFERENCE_DRAFTS__": {
    "ietf-te-types@2018-03-05": "draft-ietf-teas-yang-te-14",
    "ietf-eth-tran-types@2018-03-01":
      "draft-zheng-ccamp-otn-client-signal-yang-02",
    "ietf-routing-types@2017-12-04": "rfc8294",
    "ietf-eth-tran-service@2018-03-01":
      "draft-zheng-ccamp-otn-client-signal-yang-02"
  },
  "// __MISSING_ATTRIBUTES__": true,
  "ietf-eth-tran-service:eth-t-svc": {
    "eth-t-svc-instances": [
      {
        "eth-t-svc-name": "mpi1-epl-service",
        "eth-t-svc-descr":
          "TNBI Example for an EPL over ODU2 Service @ MPI1",
        "// __ DEFAULT __ eth-t-svc-type": "p2p-svc",
        "// __ DISCUSS __ te-topology-identifier": [
          "Would it be possible to use this grouping instead of",
          " re-defining the three attributes below?"
        ],
        "// __ ACTION __ access-provider-id":
          "Need to add the te-topology-identifier information",
        "// __ ACTION __ access-client-id":
          "Need to add the te-topology-identifier information",
        "// __ ACTION __ topology-id":
          "Need to add the te-topology-identifier information",
        "eth-t-svc-access-ports": [
          {
            "// comment": "10GE access interface (S3-1)",
            "access-port-id": 1,

```



```
    "// access-node-id": "S3-NODE-ID",
    "access-node-id": "10.0.0.3",
    "// access-ltp-id": "S3-1-LTP-ID",
    "access-ltp-id": 1,
    "service-classification-type":
        "ietf-eth-tran-types:port-classification",
    "// __ DISCUSS __ ingress-egress-bandwidth-profile-name":
        "10G-EPL-BWP",
    "// vlan-operations":
        "None: transparent VLAN operations"
  }
],
"eth-t-svc-tunnels": [
  {
    "tunnel-name": "mpi1-odu2-tunnel"
  }
],
"admin-status": "ietf-te-types:tunnel-state-up"
}
]
}
```

Authors' Addresses

Italo Busi (Editor)
Huawei

Email: italo.busi@huawei.com

Daniel King (Editor)
Lancaster University

Email: d.king@lancaster.ac.uk

Haomian Zheng (Editor)
Huawei

Email: zhenghaomian@huawei.com

Yunbin Xu (Editor)
CAICT

Email: xuyunbin@rictt.cn

Yang Zhao
China Mobile

Email: zhaoyangyjjy@chinamobile.com

Sergio Belotti
Nokia

Email: sergio.belotti@nokia.com

Gianmarco Bruno
Ericsson

Email: gianmarco.bruno@ericsson.com

Young Lee
Huawei

Email: leeyoung@huawei.com

Victor Lopez
Telefonica

Email: victor.lopezalvarez@telefonica.com

Carlo Perocchio
Ericsson

Email: carlo.perocchio@ericsson.com

Ricard Vilalta
CTTC

Email: ricard.vilalta@cttc.es