Transport Northbound Interface Applicability Statement   
draft-ietf-ccamp-transport-nbi-app-statement-05

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 August 22, 2019.

Copyright Notice

Copyright (c) 2019 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](#_Toc1723562)

[1.1. The scope of this document 4](#_Toc1723563)

[1.2. Assumptions 5](#_Toc1723564)

[2. Terminology 6](#_Toc1723565)

[3. Conventions used in this document 7](#_Toc1723566)

[3.1. Topology and traffic flow processing 7](#_Toc1723567)

[3.2. JSON code 8](#_Toc1723568)

[4. Scenarios Description 9](#_Toc1723569)

[4.1. Reference Network 9](#_Toc1723570)

[4.2. Topology Abstractions 14](#_Toc1723571)

[4.3. Service Configuration 15](#_Toc1723572)

[4.3.1. ODU Transit 16](#_Toc1723573)

[4.3.2. EPL over ODU 17](#_Toc1723574)

[4.3.3. Other OTN Clients Services 18](#_Toc1723575)

[4.3.4. EVPL over ODU 19](#_Toc1723576)

[4.4. Multi-function Access Links 20](#_Toc1723577)

[4.5. Protection and Restoration Configuration 21](#_Toc1723578)

[4.5.1. Linear Protection (end-to-end) 22](#_Toc1723579)

[4.5.2. Segmented Protection 23](#_Toc1723580)

[4.6. Notification 23](#_Toc1723581)

[4.7. Path Computation with Constraint 24](#_Toc1723582)

[5. YANG Model Analysis 25](#_Toc1723583)

[5.1. YANG Models for Topology Abstraction 25](#_Toc1723584)

[5.1.1. Domain 1 Black Topology Abstraction 26](#_Toc1723585)

[5.1.2. Domain 2 Black Topology Abstraction 28](#_Toc1723586)

[5.1.3. Domain 3 White Topology Abstraction 28](#_Toc1723587)

[5.1.4. Multi-domain Topology Stitching 29](#_Toc1723588)

[5.1.5. Access Links 31](#_Toc1723589)

[5.2. YANG Models for Service Configuration 33](#_Toc1723590)

[5.2.1. ODU Transit Service 35](#_Toc1723591)

[5.2.1.1. Single Domain Example 37](#_Toc1723592)

[5.2.2. EPL over ODU Service 38](#_Toc1723593)

[5.2.3. Other OTN Client Services 40](#_Toc1723594)

[5.2.4. EVPL over ODU Service 40](#_Toc1723595)

[5.3. YANG Models for Protection Configuration 41](#_Toc1723596)

[5.3.1. Linear Protection (end-to-end) 41](#_Toc1723597)

[5.4. Notifications 41](#_Toc1723598)

[5.5. Path Computation with Constraints 42](#_Toc1723599)

[6. Security Considerations 42](#_Toc1723600)

[7. IANA Considerations 42](#_Toc1723601)

[8. References 42](#_Toc1723602)

[8.1. Normative References 42](#_Toc1723603)

[8.2. Informative References 44](#_Toc1723604)

[9. Acknowledgments 44](#_Toc1723605)

[Appendix A. Validating a JSON fragment against a YANG Model 46](#_Toc1723606)

[A.1. Manipulation of JSON fragments 46](#_Toc1723607)

[A.2. Comments in JSON fragments 47](#_Toc1723608)

[A.3. Validation of JSON fragments: DSDL-based approach 47](#_Toc1723609)

[A.4. Validation of JSON fragments: why not using a XSD-based approach 48](#_Toc1723610)

[Appendix B. Detailed JSON Examples 49](#_Toc1723611)

[B.1. JSON Examples for Topology Abstractions 49](#_Toc1723612)

[B.1.1. JSON Code: mpi1-otn-topology.json 49](#_Toc1723613)

[B.2. JSON Examples for Service Configuration 49](#_Toc1723614)

[B.2.1. JSON Code: mpi1-odu2-service-config.json 49](#_Toc1723615)

[B.2.2. JSON Code: mpi1-odu2-tunnel-config.json 49](#_Toc1723616)

[B.2.3. JSON Code: mpi1-epl-service-config.json 49](#_Toc1723617)

[B.3. JSON Example for Protection Configuration 50](#_Toc1723618)

# 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 the 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 also enable service provisioning coordination/automation. This can be achieved by using standardized YANG models, used together with an appropriate protocol (e.g., [RFC8040]).

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.

## The 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 [RFC8453].

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 [RFC8453].

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]. It also considers the Ethernet Client Topology augmentation defined in [CLIENT-TOPO] as well as the Client Signal YANG models defined in [CLIENT-SIGNAL] for both Transparent and Ethernet clients.

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 input for defining the reference scenarios analyzed in this document.

## Assumptions

This document is making the following assumptions:

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.

See also section 1.7 of [TE-TUTORIAL] for more details.

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

1. The topology information for the Ethernet Client access links is modelled using the YANG model defined in [CLIENT-TOPO].
2. The topology information for the OTN and Transparent Client access links is modelled using the YANG model defined in [OTN-TOPO].
3. The mapping information for Ethernet and Transparent Client signals are modelled using the YANG model defined in [CLIENT-SIGNAL].

Finally, the Network Elements (NEs) described in the scenarios used in document are using ODU switching. It is assumed that the ODU links are pre-configured and using mechanisms such as WDM wavelength, which are outside the scope of this document.

# Terminology

Domain: A domain as defined by [RFC4655] is "any collection of network elements within a common sphere of address management or path computation responsibility". Specifically, within this document we mean a part of an operator's network that is under common management (i.e., under shared operational management using the same instances of a tool and the same policies). Network elements will often be grouped into domains based on technology types, vendor profiles, and geographic proximity

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

*[****Editors’ note****:] Add terminology for end-to-end data plane connection, data plane segment connection.*

# Conventions used in this document

## 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 just switching at a given layer "[(switching)]" or also having an adaptation of a client layer into a server layer "[(client) -> server]" or [client -> (server)], depending on whether the node is switching in the client or in the server layer.

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.

## 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 will insert 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 validated against the YANG model definitions.

# Scenarios Description

## 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 which provides transport connectivity services to an IP customer network through eight access links:

........................

.......... : :

: : Network domain 1 : .............

Customer: : : : :

domain : : S1 -------+ : : Network :

: : / \ : : domain 3 : ..........

R1 ------- S3 ----- S4 \ : : : :

: : \ \ S2 --------+ : :Customer

: : \ \ | : : \ : : domain

: : S5 \ | : : \ : :

R2 ------+ / \ \ | : : S31 --------- R7

: : \ / \ \ | : : / \ : :

: : S6 ---- S7 ---- S8 ------ S32 S33 ------ R8

: : / | | : : / \ / : :.......

R3 ------+ | | : :/ S34 : :

: :..........|.......|...: / / : :

........: | | /:.../.......: :

| | / / :

...........|.......|..../..../... :

: | | / / : ..............

: Network | | / / : :

: domain 2 | | / / : :Customer

: S11 ---- S12 / : : domain

: / | \ / : :

: S13 S14 | S15 ------------- R4

: | \ / \ | \ : :

: | S16 \ | \ : :

: | / S17 -- S18 --------- R5

: | / \ / : :

: S19 ---- S20 ---- S21 ------------ R6

: : :

:...............................: :.............

1. - Reference network

This document assumes that all the transport network switching nodes Si are capable of switching in the electrical domain (ODU switching) and that all the Si-Sj OTN links within the transport network (intra-domain or inter-domain) are 100G links while the access Ri-Sj links are 10G links.

It is also assumed that, within the transport network, the physical/optical interconnections supporting the Si-Sj 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.

Different technologies can be used on the access links (e.g., Ethernet, STM-N and OTU). Section 4.3 provides more details about the different assumptions on the access links for different services types and section 4.4 describes the control of access links which can support different technology configuration (e.g., STM-64, 10GE or OTU2) depending on the type of service being configured (multi-function access links).

The transport domain control architecture, shown in Figure 2, follows the ACTN architecture and framework document [RFC8453], and functional components:

--------------

| |

| CNC |

| |

--------------

|

....................|....................... CMI

|

----------------

| |

| MDSC |

| |

----------------

/ | \

/ | \

............../.....|......\................ MPIs

/ | \

/ ---------- \

/ | PNC2 | \

/ ---------- \

---------- | \

| PNC1 | ----- \

---------- ( ) ----------

| ( ) | PNC3 |

----- ( Network ) ----------

( ) ( Domain 2 ) |

( ) ( ) -----

( Network ) ( ) ( )

( Domain 1 ) ----- ( )

( ) ( Network )

( ) ( Domain 3 )

----- ( )

( )

-----

1. - Controlling Hierarchy

The ACTN framework facilitates the detachment of the network and service control from the underlying technology and helps the customer define the network as desired by business needs. Therefore, care must be taken to keep a minimal level of 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.

The control interfaces within the scope of this document are the three MPIs shown in Figure 2.

It is worth noting that the split of functionality at the MPI in the ACTN architecture between the MDSC and the PNCs is equivalent/analogous to the split of functionality which is assumed for the ONF T-API interface when used between a multi‑domain controller and domain controllers, as described in the ONF T-API multi‑domain use cases [ONF TR-527], as well as at the MEF PRESTO interface between the Service Orchestration Functionality (SOF) and the Infrastructure Control and Management (ICM) in the MEF LSO Architecture [MEF 55].

This document does not make any assumption about the control architecture of the customer IP network: in line with [RFC8453], the CNC is just a functional component within the customer control architecture which is capable to request, at the CMI, transport connectivity between IP routers, when needed.

The CNC can request transport connectivity services between IP routers which can be attached to different transport domains (e.g., between R1 and R5 in Figure 1) or to the same transport domain (e.g., between R1 and R3 in Figure 1). Since the CNC is not aware of the transport network controlling hierarchy, the mechanisms used by the CNC to request, at the CMI, transport connectivity services are independent on whether the service request is single-domain or multi-domain.

It is assumed that the CMI allows the CNC to provide all the information that is required by the MDSC to understand the connectivity service request and to decide the network configurations to be requested, at the MPIs, to its underlying PNCs to support the requested connectivity service.

When a single-domain service is requested by the CNC at the CMI (e.g., between R1 and R3 in Figure 1), the MDSC can follow the same procedures, described above for the multi-domain service, and decide the network configuration to request only at the MPI of the PNC controlling that domain (e.g., MPI1 of PNC1 in Figure 2).

Alternatively, the MDSC can pass the service configuration to that PNC and let the PNC take all the decisions about the network configuration required to support the connectivity requested by the CNC within its domain.

## 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 the 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.”

[RFC8453] 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 [RFC8453], there are three types of topology:

* 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;
* 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;
* Grey topology: This abstraction level is between black topology and white topology from a granularity point of view. This is an 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 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 network topology abstractions hiding the internal details of the physical domain network topology controlled by the PNC and this abstraction is independent from the abstractions provided by other PNCs. Therefore it is possible that different PNCs provide different types of topology abstractions and each MPI operates on the abstract topology regardless of, and independently from, the type of abstraction provided by the PNC.

To analyze how the MDSC can operate on abstract topologies independently from the topology abstraction provided by each PNC and, therefore, that different PNCs can provide different topology abstractions, that the following examples are assumed:

* PNC1 and PNC2 provide black topology abstractions which expose at MPI1, and MPI2 respectively, a single virtual node (representing the whole network domain 1, and domain 2 respectively).
* PNC3 provides a white topology abstraction which exposes at MPI3 all the physical nodes and links within network domain 3.

The MDSC should be capable of stitching together the abstract topologies provided by each PNC 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. Analyzing the topology abstractions provided by the MDSC to its CMIs is outside the scope of this document.

## Service Configuration

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

The type of services could depend on 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, Ri (PKT -> foo) and Rj (foo -> 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.

### ODU Transit

The physical links interconnecting the IP routers and the transport network can be 10G 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 between the PNCs and the MDSC.

For simplicity of the description, it is also assumed that these interfaces are not channelized (i.e., they can only support one ODU2).

To setup a 10Gb IP link between R1 and R5, an ODU2 end-to-end connection needs to be created, passing through transport nodes S3, S1, S2, S31, S33, S34, S15 and S18 which belong to different PNC domains (multi-domain service request):

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

The MDSC understands that it needs to setup of an ODU2 transit service between the access links on S3 and S18, which belong to different PNC domains (multi‑domain service request). It also decides the network configurations to request, at the MPIs, to its underlying PNCs, to coordinate the setup of a multi‑domain ODU2 segment connection between the access links on S3 and S18.

To setup of a 10Gb IP link between R1 and R3, an ODU2 end-to-end connection needs to be created, passing through transport nodes S3, S5 and S6 which belong to the same PNC domain (single-domain service request):

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

As described in section 4.1, the mechanisms used by the CNC at the CMI are independent on whether the service request is single-domain service or multi-domain.

The MDSC can understand that it needs to setup an ODU2 transit service between the access links on S3 and S6, which belong to the same PNC domain (single‑domain service request).

As described in section 4.1, the MDSC may decide to pass this service request to the underlying PNC (e.g., PNC1). In this case, the PNC (e.g., PNC1) can autonomously decide to setup the single-domain ODU2 segment connection between its access links on S3 and S6, passing through node S5.

### EPL over ODU

The physical links interconnecting the IP routers and the transport network can be 10G Ethernet physical links (10GE).

In this case, it is assumed that the Ethernet physical interfaces (up to the MAC layer) are pre‑configured using mechanisms which are outside the scope of this document and not exposed at the MPIs between the PNCs and the MDSC.

To setup a 10Gb IP link between R1 and R5, an EPL service needs to be created, supported by an ODU2 end-to-end connection, between transport nodes S3 and S18, passing through transport nodes S1, S2, S31, S33, S34 and S15, which belong to different PNC domains (multi-domain service request):

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

The MDSC understands that it needs to setup an EPL service between the access links on S3 and S18, which belong to different PNC domains (multi‑domain service request). It also decides the network configurations to request, at the MPIs, to its underlying PNCs, to coordinate the setup of an end‑to‑end ODU2 connection between the nodes S3 and S8, including the configuration of the adaptation functions inside these edge nodes, such as S3 (ETH -> [ODU2]) and S18 ([ODU2] -> ETH).

To setup a 10Gb IP link between R1 and R3, an EPL service needs to be created, supported by an ODU2 end-to-end connection between transport nodes S3 and S6, passing through the transport node S5, which belong to the same PNC domain (single-domain service request):

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

As described in section 4.1, the mechanisms used by the CNC at the CMI are independent on whether the service request is single-domain service or multi-domain.

Based on the assumption above, in this case, the MDSC can understand that it needs to setup an EPL service between the access links on S3 and S6, which belong to the same PNC domain (single-domain service request).

As described in section 4.1, the MDSC may decide to pass this service request to the underlying PNC (e.g., PNC1). In this case, the PNC (e.g., PNC1) can autonomously decide to setup the single-domain ODU2 end-to-end connection between nodes S3 and S6, passing through node S5, as well as the configuration of the adaptation functions on these edge nodes.

### Other OTN Clients Services

[ITU-T G.709] defines mappings of different Transparent 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.) interconnecting the IP routers and the transport network.

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

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 already described (section 4.1) CNC provides the essential information to permit the MDSC to understand which type of service is needed, in this case, an STM-64 Private Line service between the access links on S3 and S8, and it also decides the network configurations, including the configuration of the adaptation functions inside these edge nodes, such as S3 (STM-64 -> [ODU2]) and S18 ([ODU2] -> STM-64).

To setup a 10Gb IP link between R1 and R3), an STM-64 Private Line service needs to be created between R1 and R3 (single-domain service request):

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

As described in section 4.3.2 (single-domain service request), the mechanisms used by the CNC at the CMI are independent on whether the service request is single-domain or multi-domain and the MDSC may decide to pass this service request to the underlying PNC (e.g., PNC1), which would autonomously decide the configuration of the network domain under its control.

### EVPL over ODU

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

As described in section 4.3.2, it is assumed that the Ethernet physical interfaces (up to the MAC layer) are pre‑configured.

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:

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

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

It is worth noting that the fist EVPL service is required between access links which belong to the same PNC domain (single-domain service request) while the second EVPL service is required between access links which belong to different PNC domains (multi-domain service request).

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.

Based on the assumptions described in section 4.3.2, the CNC requests at the CMI the MDSC to setup these EVPL services and the MDSC understands what to do as described in section 4.3.2.

## 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 which are outside the scope of this document. In this case, these links will appear at the MPI as links supporting only one mode (depending on the a-priori configuration) and will be controlled at the MPI as discussed in section 4.3: for example, a 10G multi‑function access link can be configured to operate as a 10G OTN link (section 4.3.1), a 10GE link (section 4.3.2) or an STM-64 link (section 4.3.3).

It is also possible not to configure these links a-priori and let the MDSC (or, in case of a single-domain service request, the PNC) to decide how to configure these links, based on the service configuration.

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])

The CNC is capable to request at the CMI the setup either an STM-64 Private Line service, between R1 and R7, or an EPL service, between R1 and R5.

The MDSC, based on the service being request, decides the network configurations to request, at the MPIs, to its underlying PNCs, to coordinate the setup of an end‑to‑end ODU2 connection, either between nodes S3 and S31, or between nodes S3 and S18, including the configuration of the adaptation functions on these edge nodes, and in particular whether the multi-function access link between R1 and S3 should operate as an STM-64 or as a 10GE link.

## 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 the 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. The description of services using dynamic restoration is outside the scope of this document.

The MDSC needs to be capable of decide the network configuration to request to different PNCs to coordinate the protection switching configuration to support 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 is assumed to be provided only at the OTN ODU layer.

### Linear Protection (end-to-end)

In order to protect the connectivity services described in section 4.3 from failures within the OTN multi-domain transport network, the MDSC can decide to request its underlying PNCs to configure ODU2 linear protection between the access nodes (e.g., nodes S3 and S18 for the services setup between R1 and R5).

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:

* 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

* 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.

### Segmented Protection

To protect the connectivity services defined in section 4.3 from failures within the OTN multi-domain transport network, the MDSC can decide to request its underlying PNCs to configure ODU2 linear protection between the edge nodes of each domain.

For example, MDSC can request PNC1 to configure linear protection between its edge nodes S3 and S2:

Working transport entity: S3, S1, S2

Protection transport entity: S3, S4, S8, S2

MDSC can also request PNC2 to configure linear protection between its edge nodes S15 and S18:

Working transport entity: S15, S18

Protection transport entity: S15, S12, S17, S18

MDSC can also request PNC3 to configure linear protection between its edge nodes S31 and S34:

Working transport entity: S31, S33, S34

Protection transport entity: S31, S32, S34

## 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.

## Path Computation with Constraint

It is possible to define constraints to be taken into account during path computation procedures (e.g., IRO/XRO).

For example, the CNC can request, at the CMI, an ODU transit service, as described in section 4.3.1, between R1 and R5 with the constraint to pass through the link from S2 to S31 (IRO), such that a qualified path could be:

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

If the CNC instead requested to pass through the link from S8 to S12, then the above path would not be qualified, while the following would be:

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

The mechanisms used by the CNC to provide path constraints at the CMI are outside the scope of this document. It is assumed that the MDSC can understand these constraints and take them into account in its path computation procedures (which would decide at least which domains and inter‑domain links) and in the path constraints to provide to its underlying PNCs, to be taken into account in the path computation procedures implemented by the PNCs (with a more detailed view of topology).

# 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.

## YANG Models for Topology Abstraction

Each PNC reports its respective abstract topology to the MDSC, as described in section 4.2.

### Domain 1 Black Topology Abstraction

PNC1 provides the required black topology abstraction, as described in section 4.2, to expose to the MDSC, at MPI1, one TE Topology instance for the ODU layer (MPI1 OTN Topology) containing only one abstract TE node (i.e., AN1) and only inter-domain and access abstract TE links (which represent the inter-domain and access physical links), as shown in Figure 3 below.

...................................

: :

: +-----------------+ :

: | | :

(R1)- - --------| |-------- - -(S31)

: AN1-1 | | AN1-2 :

: | | :

(R2)- - --------| | :

: AN1-3 | AN1 | :

: | | :

(R3)- - --------| |-------- - -(S32)

: AN1-7 | | AN1-4 :

: | | :

: +-----------------+ :

: | | :

: AN1-6 | | AN1-5 :

:..........|..........|...........:

| |

(S11) (S12)

1. - Abstract Topology exposed at MPI1 (MPI1 OTN Topology)

***[Editors’ note:]*** *Update figure 3 to match with the new topology abstraction*

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 abstract TE Link, within the MPI1 OTN topology, for each OTU2 or OTU4 trail which support an abstract TE link in the MPI1 ODU Topology.

*[****Editors’ note:****] Add some description about the relationship between the abstract and the physical topology within the PNC1 “brain.”*

..................................

: :

: ODU Abstract Topology @ MPI :

: Gotham City Area :

: Metro Transport Network :

: :

: +----+ +----+ :

: | |S1-1 | |S2-1:

: | S1 |--------| S2 |----- - -(S31)

: +----+ S2-2+----+ :

: S1-2/ |S2-3 :

: S3-2/ Robinson Park | :

: +----+ +----+ | :

: | |3 1| | | :

(R1)- - -----| S3 |---| S4 | | :

:S3-1+----+ +----+ | :

: S3-4 \ \S4-2 | :

: \S5-1 \ | :

: +----+ \ | :

: | | \S8-3| :

: | S5 | \ | :

: +----+ Metro \ |S8-2 :

(R2)- - ------ 2/ E \3 Main \ | :

:S6-1 \ /3 a E \1 Ring \| :

: +----+s-n+----+ +----+ :

: | |t d| | | |S8-1:

: | S6 |---| S7 |---| S8 |----- - -(S32)

: +----+4 2+----+3 4+----+ :

: / | | :

(R3)- - ------ S7-4 | | S8-5 :

:S6-2 | | :

:...............|........|.......:

| |

(S11) (S12)

1. - Physical Topology discovered by PNC1

LTP mapping table:

AN1-1 -> S3-1

AN1-2 -> S2-1

AN1-3 -> S6-1

AN1-4 -> S8-1

AN1-5 -> S8-5

AN1-6 -> S7-4

AN1-7 -> S6-2

Appendix B.1.1 provides the detailed JSON code example ("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.

It is worth noting that this JSON code example does not provide all the attributes defined in the relevant YANG models:

* YANG attributes which are outside the scope of this document are not shown
* The attributes describing the label restrictions are also not shown to simplify the JSON code example
* The comments describing the rationale for not including some attributes in this JSON code example even if in the scope of this document are identified with the prefix “// \_\_COMMENT\_\_” and included only in the first object instance (e.g., in the Access Link from the AN1-1 description or in the AN1-1 LTP description)

### Domain 2 Black Topology Abstraction

PNC2 provides the required black topology abstraction, as described in section 4.2, to expose to the MDSC, at MPI2, one TE Topology instance for the ODU layer (MPI2 OTN Topology) containing only one abstract node (i.e., AN2) and only inter-domain and access abstract TE links (which represent the inter-domain and access physical links).

### Domain 3 White Topology Abstraction

PNC3 provides the required white topology abstraction, as described in section 4.2, to expose to the MDSC, at MPI3, one TE Topology instance for the ODU layer (MPI3 OTN Topology) containing one abstract TE node for each physical node and one abstract TE link for each physical link (internal links, inter-domain links or access links).

### Multi-domain Topology Stitching

As assumed at 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 may be in an abstracted shape (refer to section 5.2 of [RFC8453] for a 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 needed 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.

*[****Editors’ note:****] Add some description of the abstract multi-domain topology within the MDSC “brain.”*

........................

: :

: Network domain 1 : .............

: Grey Topology : : :

: Abstraction : : Network :

: : : domain 3 :

(R1)- - -------+ : : (White) :

: \ +--------------+ :

: \ / : : \ :

: \ / : : \ :

(R2)- - --------- AN1 --+ : : S31 ---- - (R7)

: /|\ \ : : / \ : :

: / | \ +--------- S32 S33 - - (R8)

: / | \ : :/ \ / :

(R3)- - -------+ | +---+ : / S34 :

:..........|.......|...: /: / :

| | / :../........:

| | / /

...........|.......|.../..../....

: | | / / :

: Network | + / / :

: domain 2 | / / / :

: | / / / :

: | + / +--+ :

: | |/ / +--- - -(R4)

: Black +--- AN2 ---------+ :

: Topology | | :

: Abstraction | +-------------- - -(R5)

: | :

: +---------------- - -(R6)

: :

:...............................:

1. – Multi-domain Abstract Topology discovered by MDSC

### 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 a 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 the 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 reside 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-TOPO], to correlate the access links, in the client topology, with the ODU TTPs, in the OTN topology, to which access link are connected to.

## YANG Models for Service Configuration

The service configuration procedure is assumed to be initiated (step 1 in Figure 6) 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 provide 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.

|

| {1}

V

----------------

| {2} |

| {3} MDSC |

| |

----------------

^ ^ ^

{3.1} | | |

+---------+ |{3.2} |

| | +----------+

| V |

| ---------- |{3.3}

| | PNC2 | |

| ---------- |

| ^ |

V | {4.2} |

---------- V |

| PNC1 | ----- V

---------- (Network) ----------

^ ( Domain 2) | PNC3 |

| {4.1} ( \_) ----------

V ( ) ^

----- C==========D | {4.3}

(Network) / ( ) \ V

( Domain 1) / ----- \ -----

( )/ \ (Network)

A===========B \ ( Domain 3)

/ ( ) \( )

AP-1 ( ) X===========Z

----- ( ) \

( ) AP-2

-----

1. - 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 6).

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 6) 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 6).

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. 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). An 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.

*[****Editor’s Note:]*** *Add some notes for the single-domain case*

### 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.

*[****Editors’ note****:] Add some information about the path computation step.*

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:

* Source and Destination TTPs are not specified (since it is a Transit Tunnel)
* Ingress and egress points are indicated in the route-object-include-exclude list of the explicit-route-objects of the primary path:
  + The first element references the access link terminating on S3-1 LTP

*[****Editor’s note****:] The need for the second element is for further study.*

* + 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 the physical nodes S2 and S31 respectively).

*[****Editor’s note****:] These working assumptions seem generic and not specific for the YANG models defined by IETF: should we move it to section 4?*

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.

*[****Editor’s note****:] Complete the description to cover the other domains as well as the status reporting.*

#### 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.

*[****Editor’s note****:] Complete the description of the single-domain scenario.*

The Transport PNC reports the status of the created ODU2 (Transit Segment) Tunnel and its path within the ODU Topology as shown in Figure 7 below:

..................................

: :

: ODU Abstract Topology @ MPI :

: :

: +----+ +----+ :

: | | | | :

: | S1 |--------| S2 |- - - - -(R4)

: +----+ +----+ :

: / | :

: / | :

: +----+ +----+ | :

: | | | | | :

(R1)- - - - - S3 |---| S4 | | :

:S3-1 <<= + +----+ | :

: = \ | :

: = \ \ | :

: == ---+ \ | :

: = | \ | :

: = S5 | \ | :

: == --+ \ | :

(R2)- - - - - = \ \ | :

:S6-1 \ / = \ \ | :

: +--- = +----+ +----+ :

: | = | | | | :

: | S6 = --| S7 |---| S8 |- - - - -(R5)

: +--- = +----+ +----+ :

: / = :

(R3)- - - - - <<== :

:S6-2 :

:................................:

1. - ODU2 Transit Tunnel

### EPL over ODU Service

In this scenario, described in section 4.3.2, the access links are configured as Ethernet Links.

*[****Editors’ note****:] Need to add information about the use of the Ethernet client topology.*

*[****Editor’s Note:]*** *Add considerations for the case the access links are multi-function access 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 OTN topology information, described in section 5.1.1 above than 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 of 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 select not only the optimal TTP but also a TTP that would allow the Tunnel to be used by the service.

It is assumed that in case of 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-SIGNAL] YANG model at MPI1.

### Other OTN Client Services

*[****Editor’s Note:]*** *Update this section to describe the multi-domain scenario*

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

*[****Editor’s Note:]*** *Add considerations for the case the access links are multi-function access 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:

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

### EVPL over ODU Service

*[****Editor’s Note:]*** *Update this section to describe the multi-domain scenario*

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:

* the MDSC needs to understand that R1, R3 and R4 are connected, thought the Ethernet access links, with S3, S6 and S2
* the MDSC needs to understand which TTPs in S3, S6 and S2 can be accessed by these access links
* 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 of supporting 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).

## YANG Models for Protection Configuration

### Linear Protection (end-to-end)

To be discussed in future versions of this document.

## Notifications

***[Editors’ note:]*** *Just provide references to specific drafts*

Further detailed analysis is outside the scope of this document

## Path Computation with Constraints

***[Editors’ note:]*** *Just provide references to specific drafts*

The path computation constraints that can be supported at the MPI using the IETF YANG models defined in [TE-TUNNEL] and [PATH-COMPUTE].

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.

Further detailed analysis is outside the scope of this document

# Security Considerations

Inherently OTN networks ensure privacy and security via hard partitioning of traffic onto dedicated circuits. The separation of network traffic makes it difficult to intercept data transferred between nodes over OTN-channelized links.

This document analyses the applicability of the YANG models being defined by the IETF to support OTN single and multi-domain scenarios.

There are no specific new security considerations introduced by this document.

In OTN the (General Communication Channel) GCC is used for OAM functions such as performance monitoring, fault detection, and signaling. The GCC control channel should be secured using a suitable mechanism.

# IANA Considerations

This document requires no IANA actions.

# References

## 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.

[RFC8453] Ceccarelli, D., Lee, Y. et al., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC8453, August 2018.

[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-TOPO] Liu, X. et al., "YANG Data Model for TE Topologies", draft-ietf-teas-yang-te-topo, work in progress.

[OTN-TOPO] Zheng, H. et al., "A YANG Data Model for Optical Transport Network Topology", draft-ietf-ccamp-otn-topo-yang, work in progress.

[CLIENT-TOPO] 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-ietf-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-SIGNAL] Zheng, H. et al., "A YANG Data Model for Optical Transport Network Client Signals", draft-zheng-ccamp-otn-client-signal-yang, work in progress.

## 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.

[RFC8040] Bierman, A. et al., "RESTCONF Protocol", RFC 8040, January 2017.

[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.

[RFC-FOLD] Watsen, K. et al., "Handling Long Lines in Artwork in Internet-Drafts and RFCs", draft-ietf-netmod-artwork-folding, work in progress

[TE-TUTORIAL] Bryskin, I. et al., "TE Topology and Tunnel Modeling for Transport Networks", draft-ietf-teas-te-topo-and-tunnel-modeling, 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>

*[MEF 55] Add reference to MEF 55 specification*

# 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-TOPO] 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.

The authors would like to thank Henry Yu and Aihua Guo for their input and review of the URIs structures used within the JSON code examples.

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

1. 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.

* 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 [RFC-FOLD]. 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/>

* 1. 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.

* 1. 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 8.

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

(2)

YANG-module ---> DSDL-schemas (RNG,SCH,DSRL)

| |

| (1) |

| |

Config/state JTOX-file | (4)

\ | |

\ | |

\ V V

JSON-file------------> XML-file ----------------> Output

(3)

1. – 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.

* 1. 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 9:

(1)

YANG-module ---> XSD-schema - \ (3)

+--> Validation

JSON-file------> XML-file ----/

(2)

1. – 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 9 will stop just at step (1).

1. Detailed JSON Examples

The JSON code examples provided in this appendix have been validated using the tools in Appendix A and folded using the tool in [RFC-FOLD].

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

This is the JSON code reporting the OTN Topology @ MPI:

*<< ADD text from mpi1-otn-topology.json in*

*>>*

*<<END>>*

* 1. JSON Examples for Service Configuration
     1. JSON Code: mpi1-odu2-service-config.json

This is the JSON code reporting the ODU2 transit service configuration @ MPI:

*<< ADD text from mpi1-odu2-service-config.json in*

*>>*

*<<END>>*

* + 1. JSON Code: mpi1-odu2-tunnel-config.json

The JSON code for this use case will be added in a future version of this document

An incomplete version is located on GitHub at:

<https://github.com/danielkinguk/transport-nbi>

* + 1. JSON Code: mpi1-epl-service-config.json

The JSON code for this use case will be added in a future version of this document

An incomplete version is located on GitHub at:

<https://github.com/danielkinguk/transport-nbi>

* 1. JSON Example for Protection Configuration

To be added in a future version

Authors’ Addresses

Italo Busi (Editor)

Huawei

Email: [italo.busi@huawei.com](mailto:italo.busi@huawei.com)

Daniel King (Editor)

Lancaster University

Email: [d.king@lancaster.ac.uk](mailto:d.king@lancaster.ac.uk)

Haomian Zheng (Editor)

Huawei

Email: [zhenghaomian@huawei.com](mailto:zhenghaomian@huawei.com)

Yunbin Xu (Editor)

CAICT

Email: [xuyunbin@ritt.cn](mailto:xuyunbin@ritt.cn)

Yang Zhao

China Mobile

Email: [zhaoyangyjy@chinamobile.com](mailto:zhaoyangyjy@chinamobile.com)

Sergio Belotti

Nokia

Email: [sergio.belotti@nokia.com](mailto:sergio.belotti@nokia.com)

Gianmarco Bruno

Ericsson

Email: [gianmarco.bruno@ericsson.com](mailto:gianmarco.bruno@ericsson.com)

Young Lee

Huawei

Email: [leeyoung@huawei.com](mailto:leeyoung@huawei.com)

Victor Lopez

Telefonica

Email: [victor.lopezalvarez@telefonica.com](mailto:victor.lopezalvarez@telefonica.com)

Carlo Perocchio

Ericsson

Email: [carlo.perocchio@ericsson.com](mailto:carlo.perocchio@ericsson.com)

Ricard Vilalta

CTTC

Email: [ricard.vilalta@cttc.es](mailto:ricard.vilalta@cttc.es)