Open Networking Foundation

TAPI v2.4.0 Reference Implementation Agreement

TR-547

Version 2.0 (Dec 2022)

ONF Document Type: Technical Recommendation

Open Networking Foundation
1000 El Camino Real, Suite 100, Menlo Park, CA 94025
www.opennetworking.org

Table of Contents

List of Figures

List of Tables

Document History

Version	Date	Description of Change
1.0	July 28, 2020	TR Official version.
1.1a	December 15, 2020	New complete draft for next version of TR-547 v1.1 Includes new use cases: 0d, 1g, 1h, 2a, 2b, 2c, 3d, 3e, 3f, 5d, 11a, 11b, 13b, 13c, 16a, 16b
1.1g	July 2021	Reviewed draft with selected UC for 1.1
1.1	December 2021	Final v1.1
2.0	December 2022	Updated to cover TAPI v2.4.0

1 Introduction

1.1 General introduction to the model

This ONF Technical Recommendation (TR) is the Reference Implementation Agreement (RIA) for a Transport API (TAPI) based RESTCONF implementation focused on the v2.4.0 version of the TAPI information models (pruned/refactored from the ONF Core Information Model 1.4 [ONF TR-512]) and available in the public ONF GitHub repository at:

https://github.com/OpenNetworkingFoundation/TAPI/releases/tag/v2.4.0

1.1.1 Disclaimer

This RIA is an evolving document that considers use cases as defined by network operators and end users. Such use cases often present changing or partially defined requirements. The TAPI models change based on such requirements and this is reflected in the maturity of the different use cases presented in this document.

Therefore, use cases may be listed in a draft state. Feedback from the implementations as well as the consumers of the interfaces is welcome.

1.2 Introduction to this document

This document provides a set of guidelines and recommendations for a standard use of the TAPI models in combination with the RESTCONF protocol for the implementation of the interface between network systems in charge of the control/management of networks based on WDM/OTN technologies. This document can be used in conjunction with [TR-548] which is the Reference Implementation Agreement for TAPI Streaming. [TR-548] provides an additional mechanism to some of the capabilities in this document. These are highlighted throughout this document.

The target architectures, for which this reference implementation is proposed, are conceptually described in Figure 1-1. This reference NBI will be the single interface instance[1] between Operations Support System (OSS), Orchestrator, (super or parent) Controller, etc.[2] The scope of the architecture covers multiple domains within the same network, and it might consist of one or more layers of controllers, where each layer controller will export a certain level of abstraction through its TAPI context (e.g., a hierarchical controller may consume several domain SDN-C TAPI contexts to conform a multi-domain network and exposed it as an aggregated TAPI context).

In this document we will refer to the controllers in the lower layer as **SDN domain controller or SDN-C**, and, to any hierarchical controller performing the same management/control capabilities or use cases over multiple network domains as **Software-Defined Transport Network (SDTN) controller**.

This specification is intended for the interface between an SDN-C and its client, be an Orchestrator, (super or parent) Controller or client layer systems (such OSS), where the SDN-C provides its network management through a TAPI context[3] and maintains a synchronized view in a database. The client layer which will consume the TAPI context systems may have distinct roles (e.g., physical inventory) and they may be composed of different components or applications. E.g., an OSS system composed by different pieces dedicated to different applications (such inventory, assurance, or planning).

[1] This RIA considers a single interface instance. It does not exclude operation with multiple clients that share responsibilities (such as a resilient solution or a solution where a migration from one control system to another is underway) but does not cover these cases.
[2] Any system with a repository that maintains alignment with a view of the underlying system as presented by the controller.
[3] The use cases defined in this RIA assume that the client of the NBI of the SDN-C is exclusively in charge of service/intent creation etc. such that no changes to service/intent are performed at the SDN-C or directly in the controlled network. It is recognised that in a practical environment there may be intent derived from the network (control plane) and via the UI of the SDN-C. Whilst not covered by this RIA, this behaviour is not excluded and is supported by the broader TAPI definition.

This document aims to define the base requirements for any TAPI Server entity (e.g., an SDN-C) which is intended to expose the management/control[4] capabilities of any use case such activation/configuration, service provisioning, path-computation, and monitoring over a WDM/OTN network, through the interface defined in this document.

The term management/control shall express that the scope is much wider than just configuration. The proposed common interface shall account for:

- **Configuration**, e.g., for automating and optimizing the network services creation and processes.
- **Status**, e.g., for automated configuration depending on current network status.
- **Events** (Alarms), e.g., for automated initiation of countermeasures.
- Current and Historical **Performance Values**, e.g., for perpetual network analysis.

This specification is supported by standards, protocol specifications, IETF RFCs, ITU-T recommendations and the ONF TAPI documentation. The appropriate references to this supplementary material are included where appropriate along the document to support the statements which conforms this specification. However, this document does not intend to re-define the protocols or information models composing the specification but to complement, clarify or extends in those cases where a corner case or different interpretations have been found along the mentioned standards.

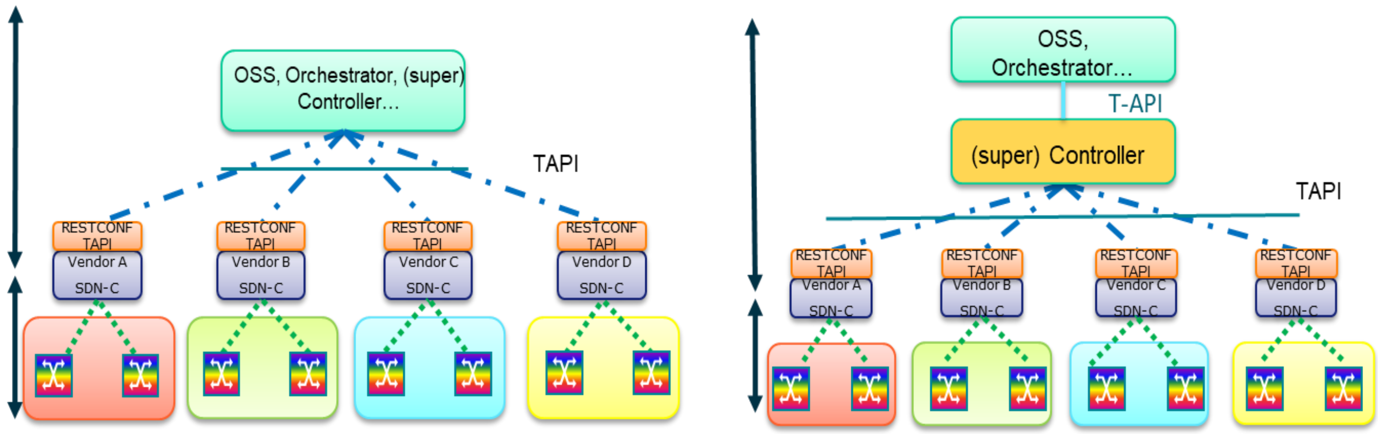


Figure 1-1 Example SDN architecture for WDM/OTN network

[4] At the time management is automated it simply becomes control as explained by [ONF TR-512].

2 RESTCONF/YANG Protocol considerations

RESTCONF [RFC 8040] is proposed as the transport protocol for all the defined operations in the SDN architecture NBIs. It is a HTTP-based protocol that provides a programmatic interface for accessing data defined in YANG [RFC 6020] using the data store concepts defined in the Network Configuration Protocol (NETCONF) [RFC 6241].

The RESTCONF specification consists of the following resources:

- **{+restconf}/data (Data API):** Create/Retrieve/Update/Delete (CRUD) based API for the entire data tree defined in the TAPI information model YANG files (see Section 3.3).
- **{+restconf}/operations (Operations API):** RPC based API consisting of a small set of operations defined as RPCs in the TAPI information model YANG files.
- **{+restconf}/data/ietf-restconf-monitoring:restconf-state/streams (Notifications API):** Implementation of the RESTCONF protocol Notifications, as defined in https://tools.ietf.org/html/rfc8040#section-6.3.
- **{+restconf}/yang-library-version:** This mandatory leaf identifies the revision date of the "ietf-yang-library" YANG module that is implemented by this server.
- **{+restconf}/data/ietf-restconf-monitoring:restconf-state/capabilities:** leaf to report the server capability of supporting query parameters defined in https://tools.ietf.org/html/rfc8040#section-9.1.

2.1 Root tree discovery

The RESTCONF API **{+restconf}** root resource can be discovered by getting either the *"/.well-known/host-meta" or the "/.well-known/host-meta.json"* resource as per [RFC6415] as described next and checking the "Link" element containing the *"restconf"* attribute. A compliant TAPI server MUST implement at least one of the following root tree discovery methods (using XRD or JRD as specified in https://datatracker.ietf.org/doc/html/rfc6415#appendix-A).

2.1.1 Extensible Resource Discovery (XRD) method

If the server supports the XRD+XML method, it MUST reply to a client sending a root tree discovery request (getting the *"/.well-known/host-meta" resource)* and using the *Accept: application/xrd+xml*

For example, the client MAY send the following query:

```
GET /.well-known/host-meta HTTP/1.1
Host: example.com
Accept: application/xrd+xml
```

In this case, the server MUST respond as follows:

```
HTTP/1.1 200 OK
Content-Type: application/xrd+xml
Content-Length: nnn
<XRD xmlns='http://docs.oasis-open.org/ns/xri/xrd-1.0'>
  <Link rel='restconf' href='/restconf'/>
</XRD>
```

2.1.2 JSON Resource Discovery (JRD) method

If the server supports the JRD method, it MUST reply to a client that is requesting the *"/.well-known/host-meta" or the "/.well-known/host-meta.json"* resource with *Accept: application/json*. The JRD document format is a general-purpose XRD 1.0 representation -- uses the JavaScript Object Notation (JSON) format defined in [RFC4627].

In this case, the client MAY use either query:

```
GET /.well-known/host-meta HTTP/1.1
Host: example.com
Accept: application/json
```

Or

```
GET /.well-known/host-meta.json HTTP/1.1
Host: example.com
Accept: application/json
```

The server MUST reply with Content-type: "application/json". Any other "Content-Type" value (or lack thereof) indicates that the server does not support the JRD format. The reply MUST be as follows:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnn

  { ...

      "links":[
        {
          "rel":"restconf",
          "href": "/restconf/",
        }, ...
```

2.2 YANG model's discovery

RESTCONF utilizes the YANG library [RFC 8525] to allow a client to discover the YANG module conformance information. The server MUST implement the "*ietf-yang-library*" module, which MUST identify all the YANG modules used by the server. This is located at **{+restconf}/data/ietf-yang-library**:**yang-library**

As per RFC 7950, the module is the base unit of definition in YANG. A module can augment an existing data model with additional nodes. Submodules are partial modules that contribute definitions to a module. A module may include any number of submodules, but each submodule may belong to only one module.

A module uses the "include" statement to list all its submodules. A module, or submodule belonging to that module, can reference definitions in the module and all submodules included by the module.

A module or submodule uses the "import" statement to reference external modules. Statements in the module or submodule can reference definitions in the external module using the prefix specified in the "import" statement.

The following yang tree shows the main entries from the yang-library. Note that TAPI currently does not use submodules.

```
module: ietf-yang-library
     +--ro yang-library
        +--ro module-set* [name]
        |  +--ro name                  string
        |  +--ro module* [name]
        |  |  +--ro name         yang:yang-identifier
        |  |  +--ro revision?    revision-identifier
        |  |  +--ro namespace    inet:uri
        |  |  +--ro location*    inet:uri
        |  |  +--ro submodule* [name]
        |  |  |  +--ro name        yang:yang-identifier
        |  |  |  +--ro revision?   revision-identifier
        |  |  |  +--ro location*   inet:uri
        |  |  +--ro feature*     yang:yang-identifier
        |  |  +--ro deviation*   -> ../../module/name
```



```
        |  +--ro import-only-module* [name revision]
        |     +--ro name         yang:yang-identifier
        |     +--ro revision     union
        |     +--ro namespace    inet:uri
        |     +--ro location*    inet:uri
        |     +--ro submodule* [name]
        |        +--ro name        yang:yang-identifier
        |        +--ro revision?   revision-identifier
        |        +--ro location*   inet:uri
        +--ro schema* [name]
        |  +--ro name          string
        |  +--ro module-set*   -> ../../module-set/name
        +--ro datastore* [name]
        |  +--ro name      ds:datastore-ref
        |  +--ro schema    -> ../../schema/name
        +--ro content-id    string
```

This version of the RIA only mandates the usage of the **yang-library/module-set**. Implementations MUST provide the list of supported TAPI modules with name, revision (mandatory) and namespace as shown in the following example for illustrative purposes.

```
{
   "ietf-yang-library:yang-library" : {

        "module-set" : [{
              "name" : "tapi-2.4-modules",
              "module" : [{
                   "name" : "tapi-common",
                   "revision" : "2022-10-30",    /* as example */
                   "namespace" : "urn:onf:otcc:yang:tapi-common"
                    ...
               }]
         }],
          ...
   }
}
```

2.3 Operations API (RPC) vs Data API

There are two allowed APIs resources defined in RESTCONF: direct data and RPC based. Given the low penetration in the industry of the RPC-based API implementation, this specification does not currently consider it. **In this specification, the support of the RESTCONF 'data' API is mandatory and the support of the 'operations' API, based on the TAPI defined RPCs, is optional.**

2.4 JSON encoding

2.4.1 Numbers

As per [RFC7951], a value of the "int8", "int16", "int32", "uint8", "uint16", or "uint32" type is represented as a JSON number. A value of the "int64", "uint64", or "decimal64" type is represented as a JSON string whose content is the lexical representation of the corresponding YANG type as specified in Sections 9.2.1 and 9.3.1 of [RFC7950]. The special handling of 64-bit numbers follows from the I-JSON recommendation to encode numbers exceeding the IEEE 754-2008 double-precision range [IEEE754-2008] as strings; see Section 2.2 in [RFC7493].

2.4.2 Empty Lists

Note the following considerations:

- Unless explicitly stated, a list without elements is NOT listed as an empty list (i.e., using ": []" in JSON encoding) and MUST NOT appear in the encoded object.
- Therefore, a container data node (which is not a presence container) that has empty lists as only children will not appear in the encoded object.
- In all specifications where a Yang list or leaf-list appears as Mandatory (M), this applies to non-empty lists.

Examples:

- If a given TAPI context has neither connectivity services nor connections instantiated upon a GET operation, the connectivity-context TAPI context augmentation will not appear even if the server supports the model (the connectivity context is not a presence container).

- I there are no CEPs instantiated over a given NEP, the NEP attribute cep-list will not appear.

2.5 Query filtering

According to the RESTCONF specification, each operation allows zero or more query parameters to be present in the request URI. Specifically, query operations' parameters are described in Section 4.8 of [RFC 8040]. Thus, the following query parameters MUST be supported by any interface compliant with this specification:

Table 1: RESTCONF Query filters

Name	Methods	Description
content	GET, HEAD	Select config and/or non-config data resources
depth	GET, HEAD	Request limited subtree depth in the reply content (Note: this parameter is deprecated and will be removed in a future version of this specification).
fields	GET, HEAD	Request a subset of the target resource contents
filter	GET, HEAD	Boolean notification filter for event stream resources. The filter contains an expression that needs to be evaluated so when the expression is "true", the event notification is delivered.
with-defaults	GET, HEAD	Control the retrieval of default values
start-time	GET, HEAD	Replay buffer start time for event stream resources
stop-time	GET, HEAD	Replay buffer stop time for event stream resources

The specific use of these query parameters will be detailed in the different Use Cases. The *"depth", "fields", "filter", "replay" (which applies to "start-time" and "stop-time" query parameters) and "with-defaults"* query parameter URIs SHALL be listed in the *"capability"* leaf-list as part of the container definition in the *"ietf-restconf-monitoring"* module, defined in Section 9.3 of [RFC 8040], to advertise the server capability of supporting these query parameters. This resource shall be located at:

- **{+restconf}/data/ietf-restconf-monitoring:restconf-state/capabilities**

2.6 JSON Data encoding

The JSON encoding MUST be supported by implementations, according to Section 3.2 of [RFC 8040]. Thus, solutions adhering to this specification MUST support media type "*application/yang-data+json*" as defined in [RFC 7951]. This MUST be advertised in the HTTP Header fields "Accept" or "Content-Type" of the corresponding HTTP Request/Response messages.

2.6.1 Namespace Qualification

According to Section 1.1.5 of [RFC 8040], "*The JSON representation is defined in "JSON Encoding of Data Modeled with YANG" [RFC7951] and supported with the "application/yang-data+json" media type"*. Any implementation according to this specification MUST be compliant with the rules and definitions included in [RFC 7951], specifically those related to namespaces qualification included in Section 4 of [RFC 7951]. For example, for an HTTP GET operation aiming at retrieving the context (note the context object is qualified)

```
GET /restconf/data/tapi-common:context HTTP/1.1
Host: example.com
Accept: application/yang-data+json
```

the response would be as follows (snippet):

```
{

      "tapi-common:context": {
      # Root tree object is qualified by the module name.

            "tapi-connectivity:connectivity-context": {

                  # Any augmentation introduces a new qualification
                  # of the module name
                  # where the augmentation was defined.


                  "connectivity-service": [{
                      "uuid": "0b530f9f-0fc3-4d27-b6c3-5c821214db1f"
                  ...
```

2.7 RESTCONF Notifications

The TAPI v2.4.0 **tapi-notification** data model defines:

- The TAPI notification context that allows to access notifications, notification channels, and to create/delete notification-subscription-services.

- Two YANG notification statements called *notification (deprecated)* and *event-notification* that wrap all notifications generated by the server.

See Section 3.2.8 for further details.

[**mandatory.restconf.notifications**] Although RESTCONF [RFC 8040] Sect 6.1 states "*A RESTCONF server MAY support RESTCONF notifications. Clients may determine if a server supports (...)*", support for RESTCONF notification is MANDATORY in this RIA, as covered in Section 2.7.1.

[**optional.streaming.notifications**] An implementation MAY support TAPI Streaming as defined in [ONF TR-548].

2.7.1 RESTCONF Notifications and Stream discovery

The support of RESTCONF notifications in this RIA is aligned with [RFC 8040], Section 6, where "*the solution preserves aspects of NETCONF event notifications [RFC5277] while utilizing the Server-Sent Events [W3C.REC-SSE]*". [RFC 8040] further explicitly states, in Sect 6.3.1, "*The server SHOULD support the NETCONF event stream defined in Section 3.2.3 of [RFC5277]. The notification messages for this stream are encoded in XML(...) the server MAY support additional streams that represent the semantic content of the NETCONF event stream but using a representation with a different media type*".

[**mandatory.json.stream**] this RIA mandates the support of event streams *with JSON encoding format*. *This RIA does not mandate the support of the NETCONF event stream. A conformant server MUST support a stream that represents the semantic content of the NETCONF event stream in JSON, the "TAPI Default RESTCONF stream(s)", as detailed below.*

2.7.1.1 SSE vs WebSocket

As stated above, the RESTCONF standard defines the *Server Sent Events (SSE*) [W3C.REC-SSE] as the standard protocol for RESTCONF stream notification service. However, some implementations (such as those demonstrated in OIF TAPI interoperability activities) rely on the use of *WebSockets (WS)* [RFC 6455] to support RESTCONF notifications. As a consequence, this RIA allows the use of either SSE or WS protocol.

2.7.1.2 RESTCONF Stream discovery

Conformant solutions MUST expose *supported notification streams* by populating the *"restconf-state/streams"* container in the *"ietf-restconf-monitoring"* module defined in Section 9.3 of [RFC 8040]. The streams resource can be found at: **{+restconf}/data/ietf-restconf-monitoring:restconf-state/streams.** The YANG tree diagram for the "ietf-restconf-monitoring" module is:

```
+--ro restconf-state
   +--ro capabilities
   |  +--ro capability*   inet:uri
   +--ro streams
      +--ro stream* [name]
         +--ro name                        string
         +--ro description?                string
         +--ro replay-support?             boolean
         +--ro replay-log-creation-time?   yang:date-and-time
         +--ro access* [encoding]
            +--ro encoding  string
            +--ro location  inet:uri
```

2.7.1.3 TAPI Default RESTCONF stream

Conformant solutions MUST expose *one stream called "tapi-notification"* **supporting the Yang notifications** defined in `tapi-notification.yang` with JSON encoding, as shown (Note that, unlike RFC5277, the use of a stream named "NETCONF" is not mandated in this specification). Solutions MAY expose additional streams. The client MUST be able to retrieve the *tapi-notification* stream location (https://example.com/streams/tapi-notification in the example) :

The streams/access/location specifies the stream source address.

```
GET /restconf/data/ietf-restconf-monitoring:restconf-state/streams HTTP/1.1
```



```
      Host: example.com
      Accept: application/json

      HTTP/1.1 200 OK
      Content-Type: application/json

      {
        "streams" : {
          "stream" [ {
               "name": "tapi-notification",
               "description" ...
               "access" : [
                    {
                      "encoding" : "json",
                      "location" : "https://example.com/streams/tapi-notification"
                    },
           ...
```

Note that the client MAY retrieve the location of the tapi-notification stream directly using:

```
GET /restconf/data/ietf-restconf-monitoring:restconf-state/streams/stream=tapi-
notification/access=json/location
```

2.7.1.4 Additional RESTCONF stream creation via TAPI (optional feature)

In addition to the existing "tapi-notification" event stream (see previous section) an implementation MAY support the dynamic creation of TAPI NotificationSubscriptionServices. This notification subscription mechanism implies the creation of RESTCONF streams and should not be confused with the RESTCONF *subscription* operation shown next. The dynamic creation of TAPI NotificationSubscriptionServices relies on sending a POST command to the notification context object with the data regarding the subscription-filter, as shown next.

```
module: tapi-notification
  augment /tapi-common:context:
    +--rw notification-context
       +--rw notif-subscription* [uuid]
        ...
       |  +--rw subscription-filter
       |  |  +--rw requested-notification-types*   notification-type
       |  |  +--rw requested-object-types*         object-type
       |  |  +--rw requested-layer-protocols*      tapi-common:layer-protocol-name
       |  |  +--rw requested-object-identifier*    tapi-common:uuid
       |  |  +--rw include-content?                boolean
       |  |  +--rw local-id?                       string
       |  |  +--rw name* [value-name]
       |  |     +--rw value-name    string
       |  |     +--rw value?        string
```

NOTE: ***include-content*** indicates whether the published Notification includes content or just the Notification Id (which would enable retrieval of the notification at the later stage). The default tapi-notification stream and the additional created streams MUST behave AS IF include-content was true.

After the NotificationSubscriptionService has been created, the object includes a *notification-channel* subtree which, notably, includes the stream-address:

```
module: tapi-notification
  augment /tapi-common:context:
    +--rw notification-context
       +--rw notif-subscription* [uuid]
       |  +--ro notification* [uuid]
```



```
        |  |  +--ro notification-type?          notification-type
        |  |  +--ro target-object-type?         object-type
        |  |  +--ro target-object-identifier?   tapi-common:uuid
        |  |  +--ro target-object-name* [value-name]
        |  |  |  +--ro value-name    string
        |  |  |  +--ro value?        string
        |  |  +--ro event-time-stamp?           tapi-common:date-and-time
        |  |  +--ro sequence-number?            uint64
        |
        |  ...
        |  +--ro notification-channel
        |  |  +--ro stream-address?     string
        |  |  +--ro next-sequence-no?   uint64
        |  |  +--ro local-id?           string
        |  |  +--ro name* [value-name]
        |  |     +--ro value-name    string
        |  |     +--ro value?        string
```

Moreover, its uuid appears both at the notification subscription service object AND in the restconf-state/streams container as shown below. Note that the access/location attribute of the new RESTCONF stream and the notification-channel/stream-address MUST be equal.

The server MUST support a client that queries the list of streams, as in:

```
GET /restconf/data/ietf-restconf-monitoring:restconf-state/streams HTTP/1.1
Host: example.com
Accept: application/json
```

With an example reply:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "streams" : {
    "stream" : [ {
         "name": "tapi-notification",
         "description" ...
         "access" : [
           {
            "encoding" : "json",
            "location" : "https://example.com/streams/tapi-notification"
           },
           ...
       }, {
         "name": "{{uuid-of-tapi-notif-subscription-service}}",
         "description" ...
         "access" : [
           {
            "encoding" : "json",
            "location" : "https://example.com/streams/{{uuid-of-tapi-notif-subs-service}}"
           },
 ...
```

2.7.1.5 RESTCONF stream subscription

For the default RESTCONF stream (and for the optionally created additional RESTCONF streams if such capability is supported), the RESTCONF server MUST support the RESTCONF Notifications subscription mechanism as defined in Section 6.3 of [RFC 8040]. For example, to subscribe to the default RESTCONF tapi-notification stream the client sends:

```
GET /streams/tapi-notification HTTP/1.1
Host: example.com
Accept: text/event-stream
```

Additionally, the server MUST support the *"filter"* Query Parameter, as defined in Section 4.8.4 of [RFC 8040], to indicate the target subset of the possible events being advertised by a RESTCONF server stream.

```
GET /streams/tapi-notification?filter={filter expression} HTTP/1.1
Host: example.com
Accept: text/event-stream
```

For additional created streams, the RESTCONF subscription is as follows (assuming the location starts at /streams)

```
GET /streams/{{uuid-of-tapi-notif-subscription-service}}?filter={filter expression}
HTTP/1.1
Host: example.com
Accept: text/event-stream
```

Note that this RIA does not specify which {filter expressions} are mandatory. Implementations should document applicable restrictions. For examples regarding the usage of RESTCONF notifications see use cases defined in Section 6.

3 ONF Transport – API (TAPI) considerations

3.1 TAPI SDK version and documentation

The ONF Transport API (TAPI) project is constantly evolving, and new releases of the information models are periodically updated. All TAPI release notes can be found at:

https://github.com/OpenNetworkingFoundation/TAPI/releases

Current document focuses on the TAPI v2.4.0 release.

3.2 TAPI Information model

The Transport API abstracts a common set of control plane functions such as Network Topology, Connectivity Requests, Path Computation, OAM, and Network Virtualization to a set of Service interfaces. It also includes support for the following technology-specific interface profiles for Carrier Ethernet (L2), Optical Transport Network (OTN) framework (L1-ODU) and Photonic Media (L0-WDM).

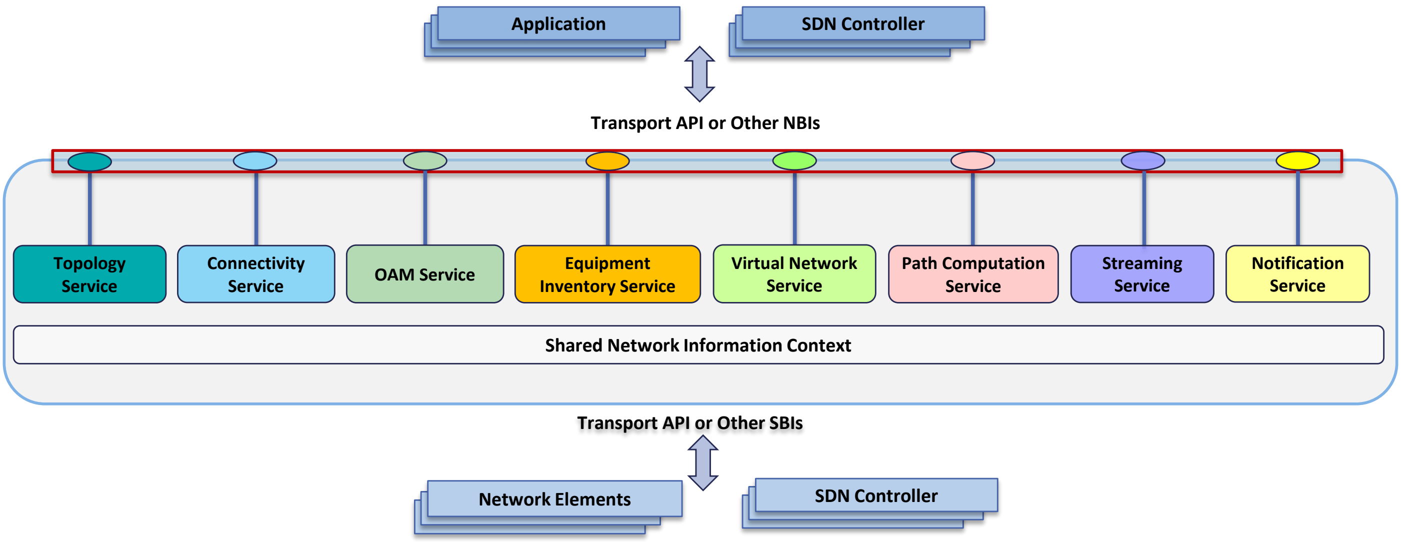


Figure 3-1 Transport API Functional Architecture

The relevant list of YANG models composing the TAPI information model of relevance for this RIA can be found in Table 2.

Table 2: TAPI YANG models summary.

Model	Version	Revision (dd/mm/yyyy)
tapi-common.yang	2.4.0	21/11/2022
tapi-connectivity.yang	2.4.0	21/11/2022
tapi-digital-otn.yang	2.4.0	21/11/2022
tapi-dsr.yang	2.4.0	21/11/2022

tapi-equipment.yang	2.4.0	21/11/2022
tapi-eth.yang	2.4.0	21/11/2022 (not used in this RIA)
tapi-fm.yang	2.4.0	21/11/2022
tapi-notification.yang	2.4.0	21/11/2022
tapi-oam.yang	2.4.0	21/11/2022
tapi-path-computation.yang	2.4.0	21/11/2022
tapi-photonic-media.yang	2.4.0	21/11/2022
tapi-streaming.yang	2.4.0	21/11/2022
tapi-topology.yang	2.4.0	21/11/2022

These models can be found at: https://github.com/OpenNetworkingFoundation/TAPI/blob/v2.4.0/YANG

TAPI models are pruned/refactored from the ONF Core Information Model (Core IM) 1.5 [ONF TR-512], thus some of the Core IM model concepts are key to understand the TAPI semantics and meanings. In this section, we introduce some associations to ONF Core IM concepts, for more a full explanation of these concepts please refer to [ONF TR-512] document.

3.2.1 Context

TAPI is based on a context relationship between a server and client. A *Context* is an abstraction that allows for logical isolation and grouping of network resource abstractions for specific purposes/applications and/or information exchange with its users/clients over an interface. It is understood that the APIs are executed within a shared Context between the API provider and its client application. A shared Context models everything that exists in an API provider to support a given API client. The TAPI server ***tapi-common:context*** includes the following information:

- The set of **Service Interface Points (SIPs)** exposed to the TAPI client applications representing the available customer-facing access points for requesting network services. This set may allow connectivity-service creation at the following layers (depending on actual deployments and hardware capabilities):
  - **DSR Layer:** Models a Digital Signal of a given rate and structure where the intent is to transparently forward the signal with minimum signal processing. It could be any type of DSR signal such xGigE, FC-x, STM-x or out-k which are included as DSR **tapi-dsr:DIGITAL_SIGNAL_TYPE** valid identities in ***tapi-dsr***. The DSR layer can be used when the intent is to represent *a basic digital layer signal processing* akin to sub-interface/circuit switching (dealing with timing, justification, buffering, etc.) . Most tapi-dsr valid identities imply a given data rate. For example, for Ethernet-based DSR types (such as DIGITAL_SIGNAL_TYPE_X_GigE), switching is based on forwarding the entire signal (all frames) as a single flow, regardless of Ethernet headers. The particular case with LAYER_PROTOCOL_QUALIFIER_UNSPECIFIED means that no information on the signal type/rate specified and could be used for variable capacity generic signals assuming the underlying devices are aware of the required signal-specific processing. *Note this RIA does not currently consider Ethernet Switching (Ethernet as a layer with its own protocol layer qualifiers e.g., terminating MAC frames, processing of C-VIDs, etc...).*
  - **DIGITAL_OTN Layer:** Models the ODU/OTU layer as per [ITU-T G.709].
  - **PHOTONIC_MEDIA Layer:** Models the OTSi/OTSiA/OTSiG, Media Channels (NMC/MC/MCA) and OMS, OTS layers as per [ITU-T G.872] using a unified set of protocol layer qualifiers: OTSiMC, MC, OMS and OTS_MEDIA.

Note that OCH is deprecated, implementations that, for example, instantiate OCH over OMS/UNSPECIFIED should migrate to OTSiMC qualifiers over OMS (with optional MC and addressing fixed grid constraints as needed). See, for example, Figure 5-32 Scenario 2 : Integrated Management, MC and OTSiMC+ODU CSs, MC not represented

- A **topology-context** which includes one or more top-level **Topology** objects. This RIA describes the use of one flat topology.
- A **connectivity-context** which includes the list of **Connectivity-Service** and **Connection** objects created within the TAPI Context.
- A **physical-context** which includes the list of **Devices, Equipment** and **Physical-spans** objects representing the physical inventory provided by the TAPI server.
- A **path-computation-context** which includes the list of **Path Computation Services** (*tapi-path-computation:path-comp-service)* requested to the TAPI server and the set of **Path** objects computed by the server.
- A **notification-context** which includes the list of **notification subscriptions** and, optionally, the list **of notifications** emitted through each notification subscription stream.
- An **oam-context** which includes the list of **OAM Services**, **OAM Profiles, OAM Jobs, and OAM MEGs.**
- A **streaming-context** with the list of available streams, and supported stream types (for further details, see companion document [TR-548]).

3.2.2 TAPI representations of the ONF Core IM Forwarding Domain

The Forwarding-Domain described in the ONF Core IM [ONF TR-512], represents the opportunity to enable forwarding between its FdPorts. The Forwarding-Domain can hold zero or more instances of Forwarding Constructs (or Connections) and provides the context for requesting and instructing the formation, adjustment, and removal of Connections. The Forwarding-Domain supports a recursive aggregation relationship such that the internal construction of a Forwarding-Domain can be exposed as multiple lower-level Forwarding-Domains and associated Links (partitioning).

For the purposes of API requirements, the Forwarding-Domain has been refactored as two separate entities: Topology and Node.

3.2.2.1 Topology

The TAPI Topology is an abstract representation of the topological-aspects of a particular set of Network Resources. It is described in terms of the underlying topological network of Nodes and Links that enable the forwarding capabilities of that set of Network Resources.

3.2.2.2 Node

The TAPI Node is an abstract representation of the forwarding-capabilities of a particular set of Network Resources. It is described in terms of the aggregation of set of ports (Node-Edge-Point, or NEP) belonging to those Network Resources and the potential to enable forwarding of information between those edge ports.

3.2.2.3 Link

A TAPI Link is a topological entity which is an abstract representation of the effective adjacency between two or more Node instances (specifically NodeEdgePoint instances) in a Topology.

3.2.3 TAPI representations of the ONF Core IM Logical Termination Point

The LogicalTerminationPoint (LTP) of the ONF Core IM [ONF TR-512] is realized by four different TAPI constructs:

- Service-Interface-Point (SIP)
- Connectivity Service-End-Point (CSEP)
- Node-Edge-Point (NEP)
- Connection-End-Point (CEP).

As the LTP is a generalized representation of termination and adaptation, each construct can model:

- Different technology layers
- Different network configurations
- Different vendor equipment capabilities

The LTP is an abstraction of the underlying network capability. Via LTP abstraction a consistent function representation can be achieved for a variety of underlying implementations as the focus of the abstraction is the functional effect of the underlying implementation, not the intricate specific implementation structure. As a consequence, the four TAPI constructs can be used to form patterns for consistent representation of solution of very different implementations.

The LTP is an encapsulation of an assembly of LayerProtocol (LP) units where the relationship between each is 1:1 fixed and immutable. The LP is an encapsulation of the addressing, mapping, termination, adaptation, and OAM functions for one transport layer. The LP can model any transport layers including analogue, circuit, and packet forms. Hence, the LTP is an encapsulation of an assembly of functions from one or more transport layers where the LPs of that assembly cane be joined client-server, client-client and/or server-server.

LTPs may be related in assemblies where there is a n:1 relationship between client and server such that the layers are split over separate instances of LTP.

An LTP instance may represent either a unidirectional function, a bidirectional function or some combination of unidirectional and bidirectional functions.

The following figure shows a mapping between ITU-T G.800/805, ONF Core and TAPI constructs. As can be seen from the figure the ONF Core LP may be split across a TAPI NEP-CEP pair.

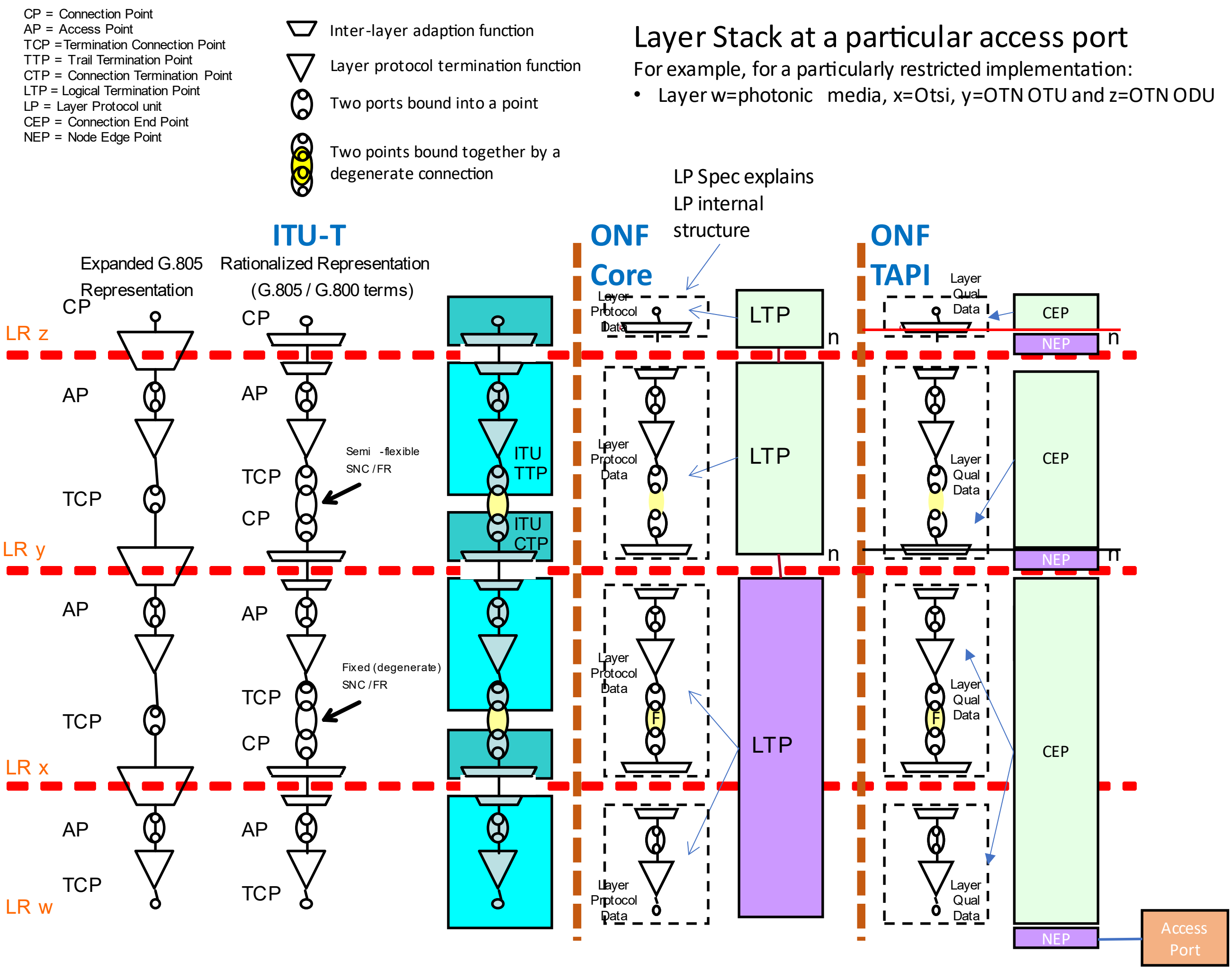


Figure 3-2 TAPI Mapping from ITU-T.

The TAPI model can be considered from several perspectives

- Potential capacity: Expressing the capacity at points and across the network as provided by existing infrastructure.
- Usage in connectivity: Expressing capacity used in a connection in the network
- Service potential: Expressing the points available for creation of services
- Service intent: Expressing the intention to use points and network capacity to achieve connectivity services.

Each of the above requires some aspect if the ONF Core LTP to be represented as discussed below.

3.2.3.1 Connection-End-Point (CEP)

The CEP (*tapi-connectivity:connection-end-point*) represents capacity and functionality used, at a particular point in the network to directly support a connection (usage in connectivity). As shown above, the CEP may cover degrees of termination, adaptation and connection flexibility at a layer. The CEP represents a binding of a portion of a ONF Core LTP and the corresponding ONF Core FcPort. The Connection-End-Point represents the ingress/egress port aspects that access the forwarding function provided by the Connection. The Connection-End-Points have a client-server relationship with the Node-Edge-Points. The Connection-End-Points have a specific role and directionality with respect to a specific Connection.

3.2.3.2 Node Edge Point (NEP)

A NEP (*tapi-topology:node-edge-point*) represents specific capacity offered by functional infrastructure at a point in the network (potential capacity). The use of this capacity will be exposed via the creation of CEPs within the NEPs and, as a consequence, the NEP can also be seen as a pool of CEPs. A NEP exposes access to the forwarding capabilities provided by a Node. It encapsulates aspects of the ONF Core LTP including mapping and adaptation with limited address processing. It may incorporate some very limited OAM functions. The NEP usually relates to a single transport layer but it may represent the mapping to several layers. It does not represent any termination or connectivity capability.

A NEP may be at the end of a link (all links end on NEPs). When a NEP is involved in a link, it represents a binding of a portion of a ONF Core LTP and the corresponding ONF Core LinkPort.

3.2.3.3 Service Interface Point (SIP)

The SIP (*tapi-common:service-interface-point*) represents the capacity at a point in the network available for creation of connectivity-services (service potential). A connectivity-service can only be created between referenced SIPs. A SIP may exist at:

- The boundary of the network where there is an inter-network interconnect (for example, where the signal passes to another operator)
- The boundary of a network protocol where there is a Termination Function as discussed earlier
- At some relevant demarcation in the network where an infrastructure service is to be started/ended.

A SIP may be referenced by zero or more NEPs where the NEP expresses actual network capacity and where that capacity is then available to the SIP and hence available for connectivity-service creation. Not all NEPs will reference a SIP as not all NEPs are available for connectivity-service creation.

A SIP may also be referenced by zero or more access-ports. The SIP then represents opportunity for connectivity-service creation from one or more of the NEPs that are present in the stack of layers associated with the access-port via the NEP that references it. Not all NEPs in the stack will be available and the expression in the SIP will clarify which are available. Not all NEPs will be associated with an access-port either directly or via a NEP-CEP hierarchy.

Hence, a SIP is an abstraction of a NEP representing specific capacity and identifying opportunity for connectivity-service creation. The SIP represents the potential/available capacity aspects of the ONF Core LTP.

3.2.3.4 Connectivity Service End Point (CSEP)

The CSEP (*tapi-connectivity:connectivity-service-end-point*) represents a port of a connectivity-service, and as such is a composed part of that connectivity-service. From an ONF Core perspective it is the port aspect of the ForwardingConstruct intention (service intent).

The CSEP moves through a lifecycle as the service is created initially only referencing the SIP as initially requested and eventually also referencing the CEP.

3.2.3.5 NEP / CEP stack modeling

The NEP / CEP stack is modeled by using the following considerations:

- Every CEP directly instantiated on top of a given NEP is listed in the cep-list parameter of the NEP.
- A single NEP reference within a CEP (***tapi-connectivity:connection-end-point/parent-node-edge-point)*** points to the NEP supporting the CEP (and which is also implicit by the position of the CEP in the Yang tree)
- A list of NEP references within a CEP (***tapi-connectivity:connection-end-point/client-node-edge-point)*** points to the NEPs instantiated over the CEP.

as shown in the Yang tree snippet below:


```
augment /tapi-common:context/tapi-topology:topology-context/tapi-topology:topology/tapi-
topology:node/tapi-topology:owned-node-edge-point:
    +--ro cep-list
       +--ro connection-end-point* [uuid]
          +--ro parent-node-edge-point
          |  +--ro topology-uuid?
          |  +--ro node-uuid?
          |  +--ro node-edge-point-uuid?
          +--ro client-node-edge-point* [topology-uuid node-uuid node-edge-point-uuid]
          |  +--ro topology-uuid
          |  +--ro node-uuid
          |  +--ro node-edge-point-uuid
```

3.2.4 TAPI Global and Local objects

TAPI models define *Global objects* and *Local objects*:

- A global object (an object that belongs to the GlobalClass) includes an *uuid* that is unique (not only in the scope of its containing parent/ancestors but also at least for the applicable TAPI context).

- A local object (an object that belongs to the LocalClass) includes a local-id which is an identifier that is unique in the context of the GlobalClass from which it is inseparable.

It is important to note that both global and local objects have a corresponding identity which is based on (inherits from) the “OBJECT_TYPE” identity in **tapi-common.yang**. Examples of global objects are SIPs, Connectivity Services, Connections, Nodes, NEPs, CEPs. Examples of local classes are CSEPs, MEPs, MIPs, Routes, ...

TAPI models assume that only a single level of containment relationship is possible between a global object and local objects (local objects cannot contain local objects).

Both local and global objects contain *a list of name-value pairs*. The list is indexed by the value name and each entry contains the value name and the actual value. This can be used e.g., in GET operations as in .../path-to-object/name["value-name"]/value.

3.2.5 Equipment model

When a TAPI server implements the equipment model, the TAPI context is augmented with additional tapi-equipment related information. The tapi-equipment/physical-context encompasses a list of devices and a list of physical-spans.

device: A logical grouping of Equipment and Access Ports that are closely located and support a coherent system of related functions. A device may be formed from one or more equipments. Examples of devices are a ROADM or an amplifier.

equipment: A (solid) physical entity[5] that is field replaceable[6]. An equipment may also include expressed non-field replaceable parts. An equipment may have holders within it.

holder: A physical space that can be fitted with an equipment.

access-port: A logical grouping of one or more pins/connectors from one or more equipments within the device that contains the access-port, that together support an indivisible flow of signal (where any one pin/connector removed from the group will prevent the signal from flowing successfully). Note that an access-port may be facing out from the device or may be internal to the device.

[5] A physical entity is something that can be measured with a ruler.

[6] An equipment is a solid physical entity that does not directly express any functionality.

physical-span: A logical grouping abstract-strands which joins two (or more) access-ports where the abstract-strands may be in series and in parallel in the physical-span. Note that not all access-ports will have associated physical-spans.

abstract-strand: A logical grouping of one or more strands[7] where the strands may be in parallel or in series, where the series of strands may be joined with a splice or a connector and where that join may be represented by one or more strand-joints.

strand-joint: An abstract representation of some of the effects of a joint between two fibers where the joint may be a simple splice, a connector or back-to-back connectors joined by fiber. A joint between two fibers may be represented by multiple strand-joints where each strand-joint carries some of the properties of the joint. A strand-joint may represent characteristics (impairments etc.) of normal flow, contra flow, reflections etc.

Note that connectors, pins, and strands are intentionally not modelled directly. The abstract access-port could be used to model an individual pin of an individual connector, the abstract-strand could be used to model a single strand and the physical-span could be used to model a cable. However, the intention is that the entities provide a significant degree of abstraction in a usual deployment.

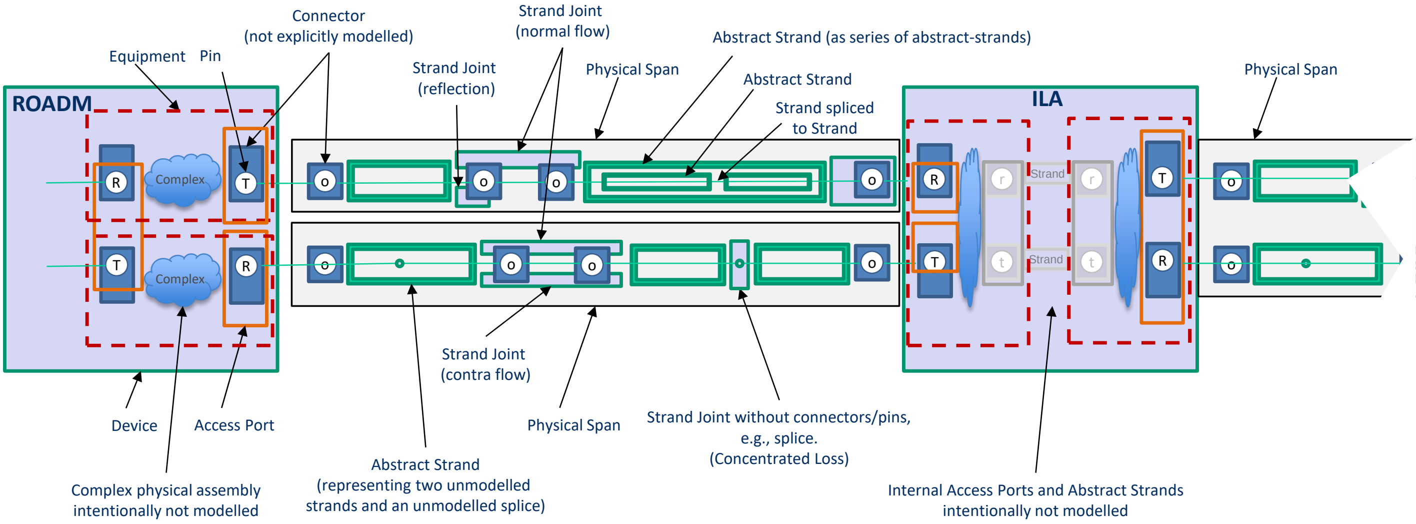


Figure 3-3 View of the Physical Span model

A connection may identify the equipments through which it passes using one or more ***physical-routes***. A physical-route is an ordered list of physical-route-elements each of which describes the connector-pin on an equipment through which the signal of the connection passes where the description is either directly in terms of connector-pin details or in terms of an access-port which then provides the connector-pin details. Any combination of direct connector-pin statements and access-port statements is allowed. This is described in the Figure 3-4.

[7] A long, thin piece of a medium such as glass fiber or copper wire with 2 ends.

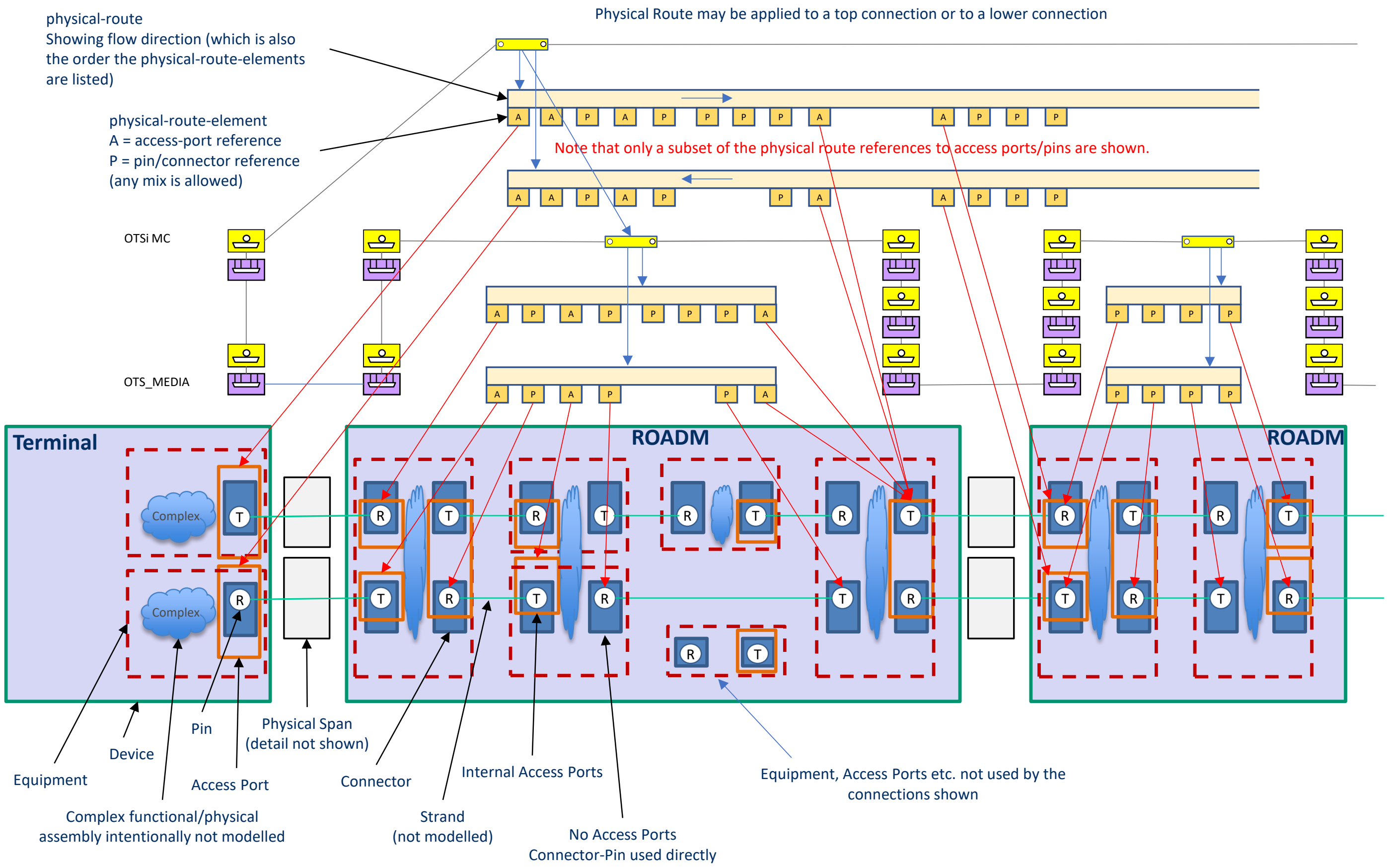


Figure 3-4 View of the Physical Route model

3.2.6 Media Channel Optical Power Considerations

TAPI SIPs and NEPs expose power capabilities (**power-management-capability-pac**), CSEPs encompass intent (**power-management-config-pac**) and CEPs expose actual configuration (**power-measurement-pac**).

3.2.6.1 power-management-capability-pac

The **power-management-capability-pac** is a list of entries, each one specifies:

- spectrum with upper-frequency and lower-frequency defining the applicable frequency range.
- 4 power data nodes each including total-power (in dBm) and/or power-spectral-density (how power of a signal is distributed over frequency specified in nW/MHz)

Note that if the capabilities are homogeneous across the whole supported frequency ranges, this list shall contain only one entry.

The 4 power nodes are:

- supportable-max-output-power
- supportable-min-output-power
- tolerable-max-input-power
- tolerable-min-input-power

For a *transceiver line port*, they refer to the range of i) output power that can be delivered towards the media channel and ii) input power that can tolerated (*expected*) from the media channel.

For a *ROADM add/drop port*, they refer to the range of i) output power that the (line) system can deliver to the next system (e.g., transponder Rx function) and ii) input power that can be tolerated (expected) from the previous system (e.g., transponder Tx function).

3.2.6.2 power-management-config-pac

The **power-management-config-pac** is a single object specifying:

- 4 power data nodes each including total-power (in dBm) and/or power-spectral-density (how power of a signal is distributed over frequency specified in nW/MHz)

The 4 power-related data nodes are:

- max-output-power
- min-output-power
- max-input-power
- min-input-power

power-management-config-pac is *optional* [the usage of this object needs clarification, and it is for further study]. It can be used for terminated (e.g., transceivers to transceiver) or unterminated (e.g., add/drop to add-drop) connectivity services

Terminated (i.e., OTSiMC)

• output-power defines a range of power that should be delivered e.g., by the local transceiver towards the MC.
• input-power defines a range of power that should be delivered e.g., by the OLS towards the local transceiver from the MC.

Unterminated

• output-power defines a range of power that should be delivered e.g., by the OLS from the MC to the local transceiver.
• input-power defines a range of power that should be delivered e.g., by the local transceiver towards the MC.

3.2.6.3 power-measurement-pac

The **power-measurement-pac** is a single object specifying:

- 2 power data nodes each including total-power (in dBm) and/or power-spectral-density (how power of a signal is distributed over frequency specified in nW/MHz)

The 2 power nodes are:

- measured-output-power measured power at the CEP
- measured-input-power measured power at the CEP

3.2.7 OTSi Optical Power Considerations

3.2.7.1 power-management-config-pac

For the provisioning of Connectivity Services (e.g., DSR or ODU) the client MAY specify layer protocol constraints that apply at the OTSi(MC), included in *tapi-connectivity:connectivity-service/end-point/layer-protocol-constraint/tapi-photonic-media:otsia-connectivity-service-end-point-spec/otsi-config/power-management-config-pac.*

In such case the min and max output power provide a valid range of launch optical power (Tx) for the transceiver. The usage of min-input-power and max-input-power is left for further specification.

3.2.8 Connectivity Model

3.2.8.1 Connectivity-Service (CS)

The TAPI Connectivity-Service represents a request for connectivity between two or more Service-Interface-Points exposed by the Context. As such, a Connectivity-Service is a container for connectivity constraints and is distinct from the Connection(s) that realize the request.[8]

3.2.8.2 Connection

The TAPI Connection represents an enabled (provisioned) forwarding capability (including all circuit and packet forms) between two or more CEPs. As such, the Connection is a container for allocated connectivity that tracks the state of the allocated resources. In this specification we distinguish two different types of connections:

- **Cross-Connections (XC)** – defined as a connection between Connection-End-Points of the same layer within a node that cannot be further decomposed into topology (represented as a *tapi-topology:node* object). Note that this RIA only considers a flat topology, so all nodes are not decomposable.
- **Top Connections**–is a connection object that represents connectivity at the highest level of partition (it is not a lower connection of another connection) and abstraction for a given layer protocol name and qualifier supporting a given connectivity service. See Section 5 for further details.

3.2.8.3 Route

The TAPI Route is an ordered list of Connection End-Points (CEPs) that reflect resources allocated to a top connection for a specific signal flow. A top connection must have at least one Route and may have more (for example, due to resilience). The CEPs in a given route include those referred to by the top connection itself as well as those referred to by a subset of the supporting cross-connections (that is, the underlying Lower-Connections referenced in the lower-connection list of the Top Connection)[9].

For a given Route instance, the following route states are foreseen:

- Current route, i.e., the route where the signal is flowing according to Controller’s best knowledge.
- Not Current route, applicable in case of resiliency schemes.

Note that *lower-connections* are used to reflect partitioning and *route* to reflect signal flow.

3.2.8.4 Path

The TAPI Path is an ordered list of TAPI Links. It is currently used to model the output of a path computation service and it is possible to refer to an existing path instance (by its uuid) during a provisioning process.

Note: A Connection is realized by concatenating link resources (resources associated with a Link) and the lower-level Connections (e.g., cross-connections) in the different Nodes.

[8] In related terminology, a connectivity service may be considered as an *intent.*

[9] The TAPI Connection Route is described in terms of Cross-Connections rather than Link-Connections (Top Connections). Conceptually a Connection Route is concatenation of Link Connections (resources associated with a Link) and Cross-Connections (resources within the Nodes in the underlying Topology).

3.2.9 Notification Model

The current TAPI information model includes two mechanisms (RESTCONF notifications and Streaming) for reporting changes using several related yang models:

- the **tapi-notification.yang**, which defines the TAPI notifications format along with a custom TAPI notification subscription procedure to enable a TAPI clients to subscribe to receive these notifications in the form of asynchronous events.
- The **tapi-fm.yang**, which contains TAPI fault management model definitions.
- the **tapi-streaming.yang**, which defines a specific TAPI streaming mechanism (as described in [ONF TR-548]).

The TAPI server MUST support tapi-notification / tapi-fm and MAY support tapi-streaming. The TAPI Notification mechanism MUST be compatible with the standard RESTCONF notification subscription mechanism described in Section 2.7.

3.2.9.1 Notification relevant parameters

For TAPI 2.4.0 there are two defined notifications, as described next. The TAPI "*notification*" notification was in use in RIA 1.1. and TAPI 2.1.3 and is currently deprecated. The new TAPI "*event-notification*" unifies the tapi-streaming and tapi-notification representations.

3.2.9.1.1 TAPI notification (until 2.4)

The TAPI *notification* notification is used to report events such as object creation, deletion or change as well as alarms (using *the tapi-fm:alarm-info* augment) and threshold crossing alerts (using *tapi-fm:tca-info* augment).

```
notifications:
    +---n notification
    |  +--ro notification-type?          notification-type
    |  +--ro target-object-type?         tapi-common:object-type
    |  +--ro target-object-identifier?   tapi-common:uuid
    |  +--ro target-object-name* [value-name]
    |  |  +--ro value-name    string
    |  |  +--ro value?        string
    |  +--ro event-time-stamp?           tapi-common:date-and-time
    |  +--ro sequence-number?            uint64
    |  +--ro source-indicator?           source-indicator
    |  +--ro layer-protocol-name?        tapi-common:layer-protocol-name
    |  +--ro layer-protocol-qualifier?   tapi-common:layer-protocol-qualifier
    |  +--ro changed-attributes* [value-name]
    |  |  +--ro value-name    string
    |  |  +--ro old-value?    string
    |  |  +--ro new-value?    string
    |  +--ro additional-info* [value-name]
    |  |  +--ro value-name    string
    |  |  +--ro value?        string
    |  +--ro additional-text?            string
    |  +--ro uuid?                       uuid
    |  +--ro name* [value-name]
    |  |  +--ro value-name    string
    |  |  +--ro value?        String

    |  +--ro tapi-fm:alarm-info
    |  |  +--ro tapi-fm:alarm-name?           tapi-common:alr
    |  |  +--ro tapi-fm:native-alarm-info?    string
    |  |  +--ro tapi-fm:is-transient?         boolean
    |  |  +--ro tapi-fm:perceived-severity?   perceived-severity-type
    |  |  +--ro tapi-fm:service-affecting?    service-affecting
    |  |  +--ro tapi-fm:alarm-category?       alarm-category
    |  |  +--ro tapi-fm:alarm-qualifier* [value-name]
    |  |     +--ro tapi-fm:value-name    string
    |  |     +--ro tapi-fm:value?        string
```



```
     |  +--ro tapi-fm:tca-info
     |     +--ro tapi-fm:threshold-indicator-name?     tapi-common:pm-parameter-name
     |     +--ro tapi-fm:is-transient?                 boolean
     |     +--ro tapi-fm:perceived-tca-severity?       perceived-tca-severity
     |     +--ro tapi-fm:threshold-observed-value
     |     |  +--ro tapi-fm:pm-parameter-int-value?    uint64
     |     |  +--ro tapi-fm:pm-parameter-real-value?   decimal64
     |     +--ro tapi-fm:threshold-configured-value
     |     |  +--ro tapi-fm:pm-parameter-int-value?    uint64
     |     |  +--ro tapi-fm:pm-parameter-real-value?   decimal64
     |     +--ro tapi-fm:oam-job?                      tapi-common:uuid
```

This section clarifies which parameters are mandatory in the use cases.

Table 3: notification object definition

Notification	**/tapi-notification:notification**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
notification-type	One of { NOTIFICATION_TYPE_OBJECT_CREATION, NOTIFICATION_TYPE_OBJECT_DELETION, NOTIFICATION_TYPE_ATTRIBUTE_VALUE_CHANGE, NOTIFICATION_TYPE_FM_ALARM_EVENT, NOTIFICATION_TYPE_FM_THRESHOLD_CROSSING_ALERT }	RO	M	• Provided by *tapi-server* • Depends on Use Case
target-object-type	See object-type list	RO	M	• Provided by *tapi-server* • Depends on Use Case • Can refer to global or local object types.
target-object-identifier	Uuid of the object to which the notification relates (see <uuid> in the examples below).	RO	M	• Provided by *tapi-server* • In case the notification refers to a TAPI local object, the target-object-identifier MUST refer to the containing parent TAPI global object. The target-object-name will specify the local-object itself.
target-object-name	List of name value pairs. 1) Includes the names of the object to which the notification relates, if any. 2) Additional name value pairs MUST be included: - "value-name": "**DRI**" - "value": Data Resource Identifier of the target object (path expression or api-path) as a string e.g., For a global object: `"/restconf/data/tapi-common:context/tapi-topology:topology-context/topology=<uuid>/node=<uuid>"` For a local object: `"/restconf/data/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service=<uuid>/end-point=<local-id>"`	RO	M	• Provided by *tapi-server* • Note the target-object-name has a min-element = 1 and the list has key "value-name" The mandatory "DRI" name value pair is as per RFC8040 section 3.5.3. *Encoding Data Resource Identifiers in the Request URI*

event-time-stamp	TAPI date-and-time	RO	M	• Provided by *tapi-server*
sequence-number	uint64 A monotonous increasing sequence number associated with the notification	RO	M	• Provided by *tapi-server* • NOTE: the sequence number MUST be monotonically increasing on a PER-CHANNEL basis. Two clients subscribing to the same stream with different filter query parameters will have notifications with different sequence numbers. • Clients MUST NOT rely on any expectation related to the actual sequence number values other than they are monotonically increasing.
source-indicator	One of { RESOURCE_OPERATION, MANAGEMENT_OPERATION, UNKNOWN }	RO	O	• Provided by *tapi-server*
layer-protocol-name	One of { DSR, DIGITAL_OTN, PHOTONIC_MEDIA }	RO	C	• Provided by *tapi-server* • This attribute is mandatory when it is not possible to infer the LPN from the target-object-type and identifier.
layer-protocol-qualifier	Identity based on LAYER_PROTOCOL_QUALIFIER	RO	C	• Provided by *tapi-server* • This attribute is mandatory when it is not possible to infer the LPQ from the target-object-type and identifier. • It is a leaf-list in event-notification
changed-attributes	In this RIA, the list of changed attributes contains ONLY one item with: - value-name: currently unused. - old-value : currently unused. - new-value : JSON object reflecting the changes of the target object as per JSON-PATCH RFC6902. Example: `[` ` {` `   "op": "add",` `   "path": "/path-to-data-node",` `   "value": [ "v1", "v2" ]` ` },` `]`	RO	C	• Provided by *tapi-server* • This field MUST appear ONLY with notification-type ATTRIBUTE_VALUE_CHANGE • NOTE: the JSON object must be included as a string. This means that the double quotes MUST be escaped, as described at ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf (Par. 9 - Strings) *"All characters may be placed within the quotation marks except for the characters that must be escaped and then it specifies: \" represents the quotation mark character (U+0022)"*
additional-info	List of name value pairs. MUST include the following: -"value-name": "**JSON**" - "value" : JSON encoded target object as a string.	RO	C	• Provided by *tapi-server* • This field MUST appear ONLY with notification-type OBJECT_CREATION

	Note that this includes ONLY the object and not the RESTCONF reply for a similar GET operation. That is, if the target object is a node, the value contains: `{` `  "uuid" : <node-uuid>,` `  "owned-node-edge-point"...` `}` And NOT `{` ` "tapi-topology:node" :` ` {` `  "uuid" : <node-uuid>,` `  "owned-node-edge-point"...` ` }` `}`			• NOTE: the JSON object must be included as a string. This means that the double quotes MUST be escaped, as described at ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf (Par. 9 - Strings) "*All characters may be placed within the quotation marks except for the characters that must be escaped and then it specifies: \" represents the quotation mark character (U+0022)*" • NOTE: event-notification is augmented with the target object for this purpose. This option is kept for backwards compatibility.
additional-text	String	RO	O	• Provided by *tapi-server*
tapi-fm:tca-info	See Section 3.2.8.4	RO	C	• Provided by *tapi-server* • This field MUST appear for TCA NOTIFICATION_TYPE_FM_THRESHOLD_CROSSING_ALERT
tapi-fm:alarm-info	See Section 3.2.8.3	RO	C	• Provided by *tapi-server* • This field MUST appear for Alarms NOTIFICATION_TYPE_FM_ALARM_EVENT
name	List of {value-name, value}	RO	O	• Provided by *tapi-server*
uuid	Notification UUID	RO	M	• Provided by *tapi-server*

3.2.9.1.2 TAPI event notification (from 2.4)

The TAPI *event-notification* notification is the new mechanism to report events such as object creation, deletion or change as well as *alarms and threshold crossing alerts* (known as detected conditions). It unifies RESTCONF notifications with TAPI streaming [TR-548] and, where applicable, it is augmented by the corresponding object.

```
+---n event-notification
       +--ro target-object-type?                                      tapi-common:object-type
       +--ro target-object-identifier?                                tapi-common:uuid
       +--ro target-local-object-type?                                tapi-common:object-type
       +--ro target-local-object-identifier?                          string
       +--ro target-object-dri?                                       string
       +--ro target-object-name* [value-name]
       |  +--ro value-name    string
       |  +--ro value?        String

       +--ro event-notification-type?                      notification-type
       +--ro event-time-stamp?                             tapi-common:date-and-time
       +--ro sequence-number?                              uint64
       +--ro source-indicator?                             source-indicator
       +--ro layer-protocol-name?                          tapi-common:layer-protocol-name
       +--ro layer-protocol-qualifier*                     tapi-common:layer-protocol-qualifier
       +--ro additional-info* [value-name]
       |  +--ro value-name    string
       |  +--ro value?        string
       +--ro uuid?                                                    uuid
       +--ro name* [value-name]
       |  +--ro value-name    string
```



```
       |  +--ro value?        string
       +--ro attribute-value-change
       |  +--ro changed-attributes?   string
       +--ro profile
       |  +--ro uuid?   uuid
       |  +--ro name* [value-name]
       |     +--ro value-name    string
       |     +--ro value?        String
...
...
       +--ro tapi-fm:detected-condition
           +--ro tapi-fm:detected-condition-name?          tapi-common:dc
           +--ro tapi-fm:detected-condition-native-name?   string
           +--ro tapi-fm:detected-condition-native-info?   string
           +--ro tapi-fm:detected-condition-qualifier?     string
           +--ro tapi-fm:oam-job?                          tapi-common:uuid
           +--ro tapi-fm:pm-metric-info
           |  +--ro tapi-fm:threshold-observed-value
           |  |  +--ro tapi-fm:pm-parameter-value?   decimal64
           |  |  +--ro tapi-fm:pm-parameter-unit?    string
           |  +--ro tapi-fm:threshold-configured-value
           |  |  +--ro tapi-fm:pm-parameter-value?   decimal64
           |  |  +--ro tapi-fm:pm-parameter-unit?    string
           |  +--ro tapi-fm:granularity-period
           |     +--ro tapi-fm:value?   uint64
           |     +--ro tapi-fm:unit?    time-unit
           +--ro tapi-fm:detector-info
           |  +--ro tapi-fm:perceived-severity?   perceived-severity-type
           |  +--ro tapi-fm:service-affecting?    service-affecting
           |  +--ro tapi-fm:is-acknowledge?       boolean
           |  +--ro tapi-fm:detector-category?    detector-category
           +--ro tapi-fm:simple-detector
              +--ro tapi-fm:simple-detector-state?   simple-detector-state
```

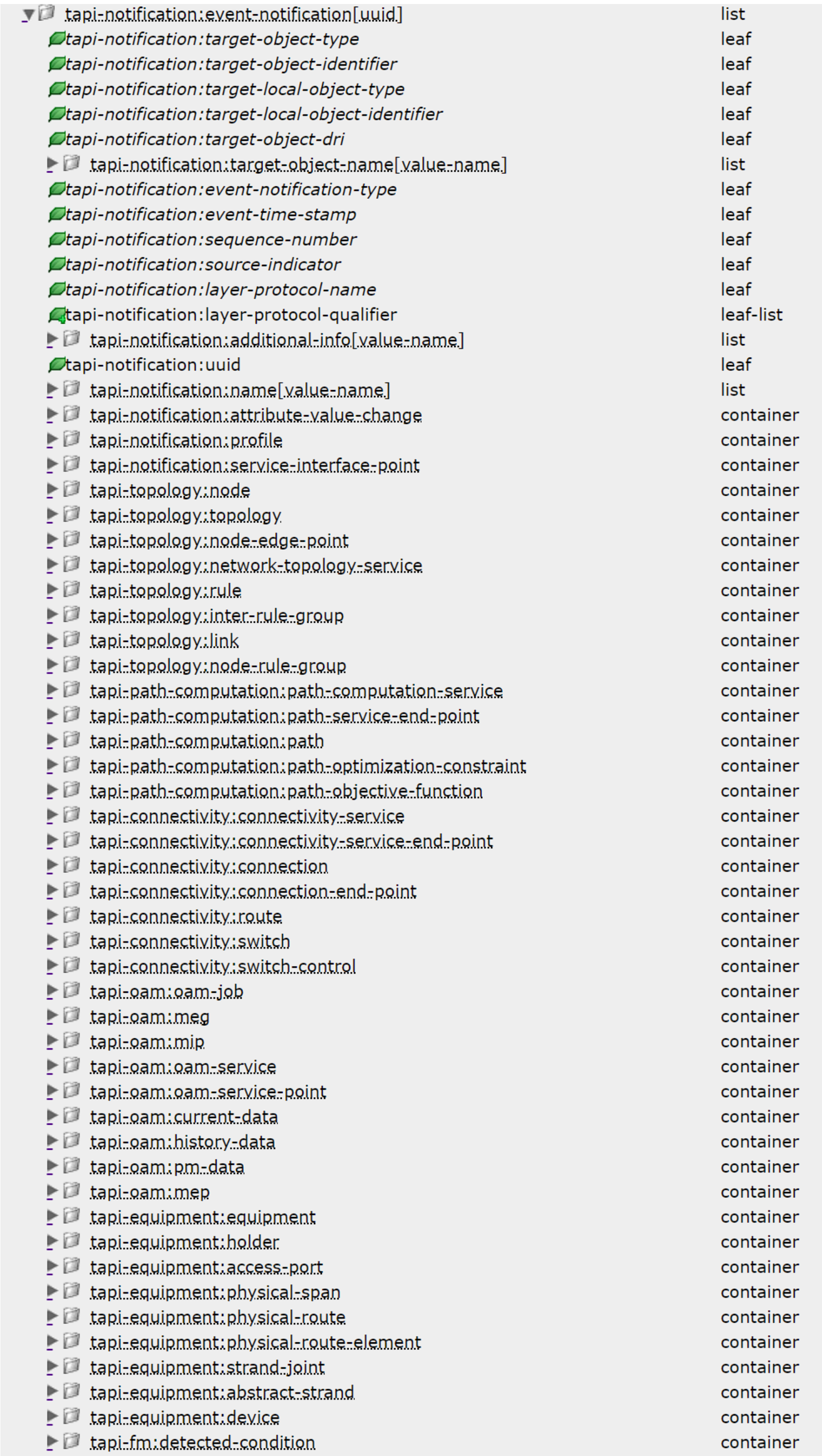
Table 4: event-notification object definition

Notification	**/tapi-notification:event-notification**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
event-notification-type	One of { NOTIFICATION_TYPE_OBJECT_CREATION, NOTIFICATION_TYPE_OBJECT_DELETION, NOTIFICATION_TYPE_ATTRIBUTE_VALUE_CHANGE, NOTIFICATION_TYPE_FM_ALARM_EVENT, NOTIFICATION_TYPE_FM_THRESHOLD_CROSSING_ALERT }	RO	M	• Provided by *tapi-server* Depends on Use Case
target-object-type	See object-type list	RO	M	• Provided by *tapi-server* • Depends on Use Case Can refer to global or the parent of a local object types.
target-object-identifier	Uuid of the object to which the notification relates.	RO	M	• Provided by *tapi-server* • The Notification instance is related to the object instance (of a global class) with this UUID value. Alternatively, the Notification is related to the object instance of a local class, whose global object has this UUID value.
target-object-local-type	See object-type list	RO	C	• Provided by *tapi-server* • Depends on Use Case • If the target of the notification is a local object this attribute MUST be present

target-object-local-identifier	string. Corresponds to the local-id	RO	C	• If the target of the notification is a local object this attribute MUST be present.
target-object-dri	String. Contains the Data Resource Identifier (DRI) of the target object (path expression or api-path) as a string e.g., For a global object: `"/restconf/data/tapi-common:context/tapi-topology:topology-context/topology=<uuid>/node=<uuid>"` For a local object: `"/restconf/data/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service=<uuid>/end-point=<local-id>"`	RO	M	• Provided by *tapi-server* • The mandatory "DRI" name value pair is as per RFC8040 section 3.5.3. *Encoding Data Resource Identifiers in the Request URI*
target-object-name	List of name value pairs. Includes the names of the object to which the notification relates, if any.	RO	M	• Provided by *tapi-server* • If this RIA specifies that the target object has mandatory object names (name value pairs inherited from the TAPI global class), the target-object-name MUST include them.
event-time-stamp	TAPI date-and-time	RO	M	• Provided by *tapi-server*
sequence-number	uint64 A monotonous increasing sequence number associated with the notification	RO	M	• Provided by *tapi-server* • NOTE: the sequence number MUST be monotonically increasing on a PER-CHANNEL basis. Two clients subscribing to the same stream with different filter query parameters will have notifications with different sequence numbers. • Clients MUST NOT rely on any expectation related to the actual sequence number values other than they are monotonically increasing.
source-indicator	One of { RESOURCE_OPERATION, MANAGEMENT_OPERATION, UNKNOWN }	RO	O	• Provided by *tapi-server*
layer-protocol-name	One of { DSR, DIGITAL_OTN, PHOTONIC_MEDIA }	RO	C	• Provided by *tapi-server* • This attribute is mandatory when it is not possible to infer the LPN from the target-object-type and identifier.
layer-protocol-qualifier	Leaf list of Identities based on LAYER_PROTOCOL_QUALIFIER	RO	C	• Provided by *tapi-server* • This attribute is mandatory when it is not possible to infer the LPQ from the

				target-object-type and identifier.
name	List of {value-name, value}	RO	O	• Provided by *tapi-server*
uuid	Notification UUID	RO	M	• Provided by *tapi-server*
attribute-value-change/changed-attributes	JSON object reflecting the changes of the target object as per JSON-PATCH RFC6902. Example: `[` ` {` `   "op": "add",` `   "path": "/path-to-data-node",` `   "value": [ "v1", "v2" ]` ` },` `]`	RO	C	• Provided by *tapi-server* • This field MUST appear ONLY with notification-type ATTRIBUTE_VALUE_CHANGE • NOTE: the JSON object must be included as a string. This means that the double quotes MUST be escaped, as described at ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf (Par. 9 - Strings) *"All characters may be placed within the quotation marks except for the characters that must be escaped and then it specifies: \" represents the quotation mark character (U+0022)"*
additional-info	Additional information that applies to the notification	RO	O	• Provided by *tapi-server*
tapi-fm:detected-condition	See Table 7	RO	C	• Provided by *tapi-server*

NOTE: event-notification is augmented with the target object for object creation notification purposes, as shown in the following (sub-)tree.


tapi-notification:event-notification[uuid] list
tapi-notification:target-object-type leaf
tapi-notification:target-object-identifier leaf
tapi-notification:target-local-object-type leaf
tapi-notification:target-local-object-identifier leaf
tapi-notification:target-object-dri leaf
tapi-notification:target-object-name[value-name] list
tapi-notification:event-notification-type leaf
tapi-notification:event-time-stamp leaf
tapi-notification:sequence-number leaf
tapi-notification:source-indicator leaf
tapi-notification:layer-protocol-name leaf
tapi-notification:layer-protocol-qualifier leaf-list
tapi-notification:additional-info[value-name] list
tapi-notification:uuid leaf
tapi-notification:name[value-name] list
tapi-notification:attribute-value-change container
tapi-notification:profile container
tapi-notification:service-interface-point container
tapi-topology:node container
tapi-topology:topology container
tapi-topology:node-edge-point container
tapi-topology:network-topology-service container
tapi-topology:rule container
tapi-topology:inter-rule-group container
tapi-topology:link container
tapi-topology:node-rule-group container
tapi-path-computation:path-computation-service container
tapi-path-computation:path-service-end-point container
tapi-path-computation:path container
tapi-path-computation:path-optimization-constraint container
tapi-path-computation:path-objective-function container
tapi-connectivity:connectivity-service container
tapi-connectivity:connectivity-service-end-point container
tapi-connectivity:connection container
tapi-connectivity:connection-end-point container
tapi-connectivity:route container
tapi-connectivity:switch container
tapi-connectivity:switch-control container
tapi-oam:oam-job container
tapi-oam:meg container
tapi-oam:mip container
tapi-oam:oam-service container
tapi-oam:oam-service-point container
tapi-oam:current-data container
tapi-oam:history-data container
tapi-oam:pm-data container
tapi-oam:mep container
tapi-equipment:equipment container
tapi-equipment:holder container
tapi-equipment:access-port container
tapi-equipment:physical-span container
tapi-equipment:physical-route container
tapi-equipment:physical-route-element container
tapi-equipment:strand-joint container
tapi-equipment:abstract-strand container
tapi-equipment:device container
tapi-fm:detected-condition container

3.2.9.2 State Propagation and Notification considerations

The following considerations specify the rules for state/notification propagation and apply to all TAPI global objects (with a uuid) as well as TAPI local objects (with a local-id within a global object). Note that for all Creation/Change notifications the Notification instance contains a Data Resource Identifier (DRI). The DRI includes the path to involved global or local object. This means a Notification of a child object will include in its DRI the identifier (address) of its parent and ancestors.

Macroscopically, to avoid excessive state propagation and a high number of notifications, this RIA, for the purposes of the generation of events/notifications, considers containment relationships as-if they were by reference. The following guidelines apply:

- [**G1**] The creation of a global object (A) that includes additional global or local objects (B) MUST trigger a CREATION notification for (A) and another CREATION notification for (B), respectively. Note that the notification associated to object (A) includes the entire subtree (as augment of the tapi-notification:event-notification/tapi-notification:object-notification data node)[10].

  The guideline implies some redundancy (i.e., for objects that have composed-by relationships, yang-tree sub-objects are included in the notification instance). For example, if a Node has been added to the Topology, the TAPI server notifications will include, at least, a notification for the node, a notification for each of its NEPs and a notification for each of the NEP's CEPs (although the content of the NEPs and CEPs was already notified in the Node notification. Each CEP gets then notified 3 times). As mentioned in the final notes, implementations MAY reduce this redundancy.

- [**G2**] A *containment relationship* (container/contained) in which a contained local/global object changes MUST NOT, by itself, be a cause for state propagation/reflection and consequently a cause for a *attribute change notification* for the container object.

- [**G3**] The creation (or deletion) of an object which is included in one or more list(s) MUST trigger: 1) a CREATION (or DELETION) notification for such object followed by 2) an ATTRIBUTE_CHANGE notification for the referencing object(s).

- [**G4**] A change in an object which is included in one or more list(s) (by reference or by value) MUST NOT trigger an ATTRIBUTE_CHANGE notification for the referencing or including object(s) UNLESS such change caused changes in other (direct) attributes of the referencing object(s).

  Note that if a client subscribes only to Node Notifications, the client will be notified of a Node when it is created (and the notification will contain the entire subtree, with its NEPs and CEPs at the time of creation), changed or deleted. That said, if one of its NEPs changes the client will not get a change notification. It is the responsibility of the client to ensure consistency.

Examples:

- A change in a NEP MUST NOT trigger a notification in the owning Node UNLESS other attributes of the node changed as a consequence of the NEP change. Examples that would trigger a Node ATTRIBUTE_CHANGE Notification:
  - o the *capacity* of the node may be present and depend on the individual *capacities* of the node NEPs, in such case an ATTRIBUTE_CHANGE notification for the Node is generated since the capacity attribute changes.

[10] Note that in TAPI 2.1.3 a deep copy of the object subject of the notification is included in the *additional-info* attribute of the notification. This is kept for backwards compatibility

    - the list of node rule groups of the node may also change. If an element (a node rule group) is added or removed from the list of node rule groups, then an ATTRIBUTE_CHANGE notification of the Node MUST be generated. If no element is added or removed from the list of node rule groups, and only an existing node rule group is affected (e.g., the NEP is added to it) only an ATTRIBUTE_CHANGE notification for the node rule group change will be generated.

- A creation (or a deletion) of a NEP MUST trigger a notification of the NEP (CREATION / DELETION) as well as an ATTRIBUTE_CHANGE notification of the Node (the list of NEPs has changed in number of elements).

- A change in a CSEP (which is a *local* object) MUST NOT trigger an ATTRIBUTE_CHANGE notification in the parent CS (which is a *global* object) UNLESS other attributes of the CS changed as a consequence of the CSEP change. As in the previous example, the CS *capacity* attribute MAY be present and depend on the CSEPs' *capacities*.

- A change in a Connection state MUST NOT trigger a notification regarding the Connectivity Service(s) that refer to such Connection UNLESS that connection caused a change in the Connectivity Service (e.g., newly included in the connectivity service's connection list). For example, if a CEP-list of a top-level connection changes, a Change Notification for the connection is emitted and also a Change Notification for all the CSEPs that refer to such added/removed CEPs but it MUST not cause a Change Notification for the CS since the CSEPs were existing.

- A change in a connection referred to by a connectivity service (e.g,, a re-route, where the route list changes) MUST cause a Change Notification in the Connection object and MUST NOT cause a Change Notification in the Connectivity Service.

- A change in a CEP MUST NOT trigger a notification regarding the parent NEP/Node UNLESS any of the other attributes of the parent NEP/Node changes due to the reflection or state propagation of the CEP change (e.g., available bandwidth).

- A change in a CEP MUST NOT trigger a notification regarding the owning Connection -- *related by reference* -- unless the change in the CEP caused a change in another Connection attribute. Likewise, a change in the CEP MUST NOT trigger a notification regarding the parent NEP -- *related by containment* -- unless there is a change in another NEP attribute (e.g., the CEP is newly created and included in the NEP's cep-list).

- A change in a Link or Node MUST NOT trigger a notification regarding the owning Topology object. A change in a NEP MUST NOT trigger a change in the parent Node unless there are additional changes.

It is understood that the process is fundamentally asynchronous and no expectations in the order of the notification of events shall be made (for example, a NEP may be notified before its corresponding Node). For this, notifications include the objects DRI (which allows placement of the target object in the Yang tree with regards to its ancestors). Clients MUST expect such notifications to happen at any order (e.g., do not expect NEPs to be announced before CEPs).

Note on notifications and subscriptions

When considering the server generated notifications upon a given network operation, this RIA provides a guideline (set of examples) of the notifications that MUST be notified to clients (for a given set of initial hypothesis and conditions). These notifications are understood in the scope of the main (default) notification stream (the actual stream and active subscriptions are orthogonal and may filter such sequence). With this in mind, note that,

1) as per the aforementioned guidelines, a change in a local object MUST NOT trigger a notification in the parent global object and
2) when considering actual client subscriptions:

- Subscribing to a RESTCONF Stream (with a GET) allows you to specify a filter. Such filter is flexible to specify global and local objects (or a combination of both)
- Creating a TAPI additional stream (in addition to the default one) currently supports the specification of selected global objects (there is no requested-local-id)

With these two hypotheses, a client that subscribes to a global object (e.g., CONNECTIVITY_SERVICE) type **only**, would not be notified of changes in its local-objects (e.g., the CSEPs).

As a consequence, this RIA mandates that the subscription to a Global Object automatically implies the subscription to the respectively contained local objects.

RESTCONF notifications do not natively support flow control

It is understood that the NOTIFICATION system is not expected to ensure total consistency, and clients MUST be robust to missed notifications. In case of communication failures, the client is expected to address inconsistencies by complementary methods, such as a performing GET operations on the relevant part(s) of the context.

The NOTIFICATION system should not be used to synchronize state between client and server. Given the nature of TCP the server can only guarantee reliable delivery of given notification when the TCP connection is active. If a client is not connected at the moment that a notification is generated, such notification will not be received and there is no defined mechanism to retrieve it.

Assuming a finite set of notifications associated to a given operation, implementations SHOULD support a form of "eventual consistency": after a certain undefined time, the client shall reach a point where after the expected sequence of notifications there are no dangling references between TAPI objects.

Note that an implementation MAY choose to delay one or more Notifications in order to pack multiple changes in a single notification. For example, a Topology Notification MAY be delayed, to include as many Node and Link changes as affected by the network operation. On the other hand, an implementation MAY choose to Notify about partial changes as they happen. Clients MUST be prepared for both cases [assuming the network state once all notifications have been emitted is the same].

Note that an implementation MAY choose to reduce redundancy in one or more Notifications sequence by leaving empty relevant objects (e.g., in child lists with global objects with uuid as key) as long as the missing information is included in related (previous or subsequent) notifications for the relevant subscription AND it is possible to correlate the information (by means of uuid and the information of DRI).

3.2.9.3 TAPI Alarm Framework using alarm-info (deprecated)

TAPI alarms are a type of notifications emitted by the TAPI server (see Section 2.7). An alarm notification includes notification-type: ALARM_EVENT. This method is kept for simple migrations to TAPI 2.4. Implementations SHOULD use the unified Detected Condition

3.2.9.3.1 Relevant Parameters (tapi-fm:alarm-info)

Alarm Event notifications have parameters included inside in the “alarm-info” object. The table below defines the relevant parameters that apply to alarm notifications, as well as additional considerations.

Table 5: Alarm information (alarm-info) Relevant Parameters

Attribute	Allowed Values/Format	Mod	Sup	Information Recorded
alarm-name	Standard Alarm and TCA List See tapi-common:alr	RO	M	LOS, AIS, LOF, Etc.

native-alarm-name	string	RO	M	Alternative/Native/Local naming for the alarm event. Usually conveys the name used by the originator device.
native-alarm-info	string	RO	O	Additional Alarm related information as provided by the originator device (for example, obtained from direct mapping of other data models or SBI)
is-transient	boolean	RO	M	To indicate if the alarm event is related to a transient fault, that has an underlying cause that soon resolves itself.
perceived-severity	One of { CRITICAL, MAJOR, MINOR, WARNING, CLEARED }	RO	M	
service-affecting	One of { SERVICE_AFFECTING, NOT_ SERVICE_AFFECTING, UNKNOWN }	RO	O	
alarm-category	One of { ALARM_CATEGORY_EQUIPMENT ALARM_CATEGORY_ENVIRONMENT ALARM_CATEGORY_CONNECTIVITY ALARM_CATEGORY_PROCESSING ALARM_CATEGORY_SECURITY }	RO	O	Alarm Category
alarm-qualifier	Standard Alarm and TCA List column AlarmQualifier	RO	C	Note: this is used when the probable-cause of the alarm-info and the target-object-identifier of the wrapping notification are not enough to identify the unique source for the alarm. For example: for an OMS_OTS CEP (target-object) and a LOS alarm, the qualifier provides the actual layer (e.g. OTS).

3.2.9.4 TAPI Threshold Crossing Alerts using tca-info (deprecated)

TAPI Threshold Crossing Alerts (TCA) are a type of notifications emitted by the TAPI server (see Section 2.7).

A threshold crossing alert notification includes notification-type: THRESHOLD_CROSSING_ALERT. Unlike other types of notifications, TCA triggering conditions (threshold values) are open to be configured and activated by the user.

[**difference.alarm.tca**] this RIA differentiates between *alarms* that are reported by a device and emitted by the TAPI server, including when some operational parameters have been crossed (by upper / lower limits), and the *threshold crossing alarms* that have a limit (threshold) **configured by the user** (even if this configuration is not specified in this RIA). Note that, even though it can be argued that a TCA is-a kind of alarm, this RIA uses different notification types. For example, as shown in the figure below, the system could send an alarm related to the crossing of the red dotted line and could send a subsequent TCA notification if/when the parameter crosses a user configured threshold (yellow dotted line).

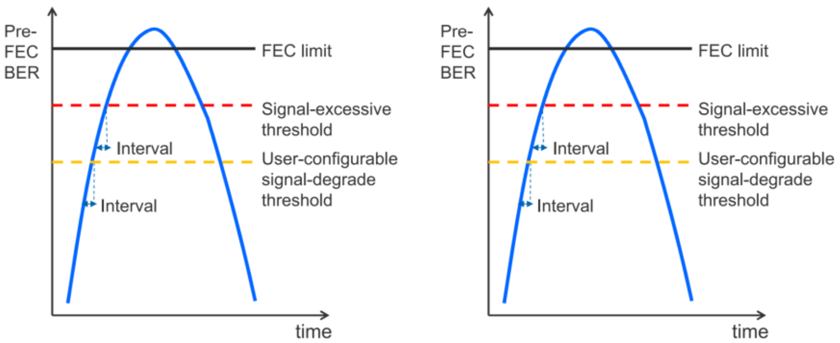


Figure 3-5 FEC function related thresholds

3.2.9.4.1 Relevant Parameters (tapi-fm:tca-info)

TCA Event notifications have parameters included inside in the “tapi-fm:tca-info” object.

Table 6: Threshold Crossing Alert information (tapi-fm:tca-info) Relevant Parameters

Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
threshold-indicator-name	tapi-common:pm	RO	M	Name of the TCA/PM metric
native-threshold-indicator-name	string	RO	M	
native-tca-info	string	RO	M	
is-transient	boolean	RO	M	To indicate if the TCA event is related to a transient condition.
perceived-tca-severity	One of PERCEIVED_TCA_SEVERITY_WARNING PERCEIVED_TCA_SEVERITY_CLEAR	RO	M	If the TCA is NOT transient implementations MUST send a notification with perceived-severity “CLEAR” when the threshold is no longer crossed.
threshold-observed-value	*Includes:* pm-parameter-value, pm-parameter-unit	RO	C	
threshold-configured-value	*Includes:* pm-parameter-value, pm-parameter-unit	RO	C	
oam-job	Reference to the Job UUID	RO	C	Applicable job
tca-qualifier	String conforming to TAPI Standard Alarm and TCA List column TCA Qualifier	RO	C	Note: this is used when the PM parameter and the target-object-identifier of the wrapping notification are not enough to identify the unique source for the TCA.
granularity-period	Includes value and unit	RO	C	Granularity period
tca-category	An identity that inherits from ALARM_CATEGORY	RO	O	TCA Category

3.2.9.5 TAPI Detected Condition (from 2.4)

Detected Conditions (alarms and TCAs) are defined in the tapi-fm.yang module. This module augments */tapi-common:context/tapi-notification:notification-context/tapi-notification:event-notification* for the purposes of transporting FM data.

3.2.9.5.1 Relevant Parameters (tapi-fm:detected-condition)

Table 7: detected-condition object definition

Notification	**/tapi-notification:event-notification/tapi-fm:detected-condition** **This augment only applies to FM notifications (ALARMs and TCAs)**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
detected-condition-name	Any identity that extends *tapi-common:dc* *Example:* ALR_BDI is a yang identity with base ALR with base DC). See tapi-common.yang for the definition of alarms. *Example:* PM_UAS is a yang identity with base PM with base DC) The name of the Condition, e.g., an alarm probable cause or the PM metric name which threshold crossing alert refers to.	RO	M	• Provided by *tapi-server*
detected-condition-native-name	Native Name used for the detected condition by the source of the information	RO	O	• Provided by *tapi-server*
detected-condition-native-info	Native Additional Info used for the detected condition	RO	O	• Provided by *tapi-server*
detected-condition-qualifier	String Further information necessary to precisely, uniquely and unambiguously identify the Condition Detector.	RO	O	• Provided by *tapi-server*
oam-job	UUID pointing to a OAM job associated with this dc.	RO	C	• Provided by *tapi-server* MUST appear if the detected condition relates to a Job
pm-metric-info	*Includes:* tapi-fm:threshold-observed-value (with pm-parameter-value and pm-parameter-unit) tapi-fm:threshold-configured-value (with pm-parameter-value and pm-parameter-unit) tapi-fm:granularity-period (with value and unit)	RO	C	• Provided by *tapi-server* MUST appear when the notification is a TCA
detector-info	*Includes:* perceived-severity (one of CRITICAL, MAJOR, MINOR or CLEARED) service-affecting (one of SERVICE_AFFECTING or NOT_SERVICE_AFFECTING) is-acknowledge, Boolean detector-category (one of DETECTOR_CATEGORY_{EQUIPMENT, ENVIRONMENT, CONNECTIVITY, PROCESSING, SECURITY, UNDEFINED}	RO	C	• Provided by *tapi-server*
simple-detector/simple-detector-state	One of: SIMPLE_DETECTOR_STATE_ACTIVE (M, alarm), SIMPLE_DETECTOR_STATE_CLEAR (M, alarm/tca, see note),	RO	C	• Provided by *tapi-server* Mandatory states are ACTIVE and CLEAR (in

	SIMPLE_DETECTOR_STATE_INTERMITTENT, SIMPLE_DETECTOR_STATE_FLEETING SIMPLE_DETECTOR_STATE_ACTIVE_NO_EXPLICIT_CLEAR			alarm and TCA when not automatically cleared) The rest are optional

3.2.10 Companion Documents

3.2.10.1 TAPI Standard Alarm and TCA List

This RIA uses the "TAPI Standard Alarm and TCA List" when identifying notifications related to alarms and threshold crossing alerts, notably related to the "alarm-name" and "threshold-parameter" data fields.

- The "TAPI Standard Alarm and TCA List" specifies terminology and identifiers related to alarms and TCA, with a description of established semantics and their relationships with specific technologies [derived from applicable standards as well as additional alarms not currently known to be standardized elsewhere].
- Implementations should align the representation of network behavior to entries in the list, without precluding that alarms or TCAs that do not align with any entry MUST still be raised using "alternative" or "native" names.
- This RIA does not mandate any behavior related to which specific or under which conditions such alarms are generated (no mandate on which Alarms or TCAs should or must be raised)

3.2.10.2 TAPI Notification and Streaming Sequence examples

This RIA provides a set of guidelines for state propagation and notification considerations (see Section 3.2.8.2). Some relevant examples are provided

3.2.10.3 Location

These normative documents are located at [CompDocs]. They are living documents (that will continue to be advanced independently from the RIA releases).

3.3 TAPI Data API

This specification does not mandate direct access to all data nodes defined by the YANG models. This section captures a minimal set of objects which shall provide full CRUD support according to the TAPI YANG model's specification (e.g., configurable objects should support all operations while non configurable objects shall support only the RETRIEVE operation). Please note that although the list of API entries is reduced here, the whole model MUST be supported, e.g., all child resources of the proposed list of objects need to be configurable.

The complete mandatory operation set of TAPI objects required here can be found in Table 8: Minimum subset required of TAPI RESTCONF Data API Table 8. [*Note: this API does not currently include items related to the equipment/physical and OAM models. This will change in a future version of the specification*].

Note that **in addition to** GET operations, TAPI Streaming (as described in [ONF TR-548]) MAY be supported as an alignment and change update mechanism.

Table 8: Minimum subset required of TAPI RESTCONF Data API

Note: Starting from RIA 1.1 API entries are mapped to use cases. Entries that are not strictly necessary or deemed inefficient for the listed use cases are tagged as <Optional> given that such entries appeared in previous versions of this specification

API Entry	RESTCONF Operations allowed	Use Case
/tapi-common:context Notes: the GET operation for the whole context has potential scalability issues. No current UC for GET, PUT and PATCH targeting the whole context object.	GET,PUT, PATCH	<Optional>
/tapi-common:context?depth=n Note: usage of depth in nodes, unless covered by a given UC may provide ambiguous responses (sliced and/or incomplete object fragments). Overall recommendation is to specify the list of requested fields and to perform more specific GET operations.	GET	<Optional>
/tapi-common:context?fields=name;uuid	GET	UC 0a
/tapi-common:context?fields=service-interface-point(uuid)	GET	UC 0a
/tapi-common:context/service-interface-point={uuid} Note: no current UC address the modification of SIPs. Further releases of this specification MAY add UCs for the modification of administrative-state and/or name list.	GET,PUT, PATCH	UC 0a
/tapi-common:context/tapi-topology:topology-context?fields=topology(uuid)	GET	UC 0b
/tapi-common:context/tapi-topology:topology-context/nw-topology-service Note: UC Ob provides alternative means to retrieve the topologies. There is no current use for the nw-topology-service.	GET	<Optional>
/tapi-common:context/tapi-topology:topology-context/nw-topology-service?fields=topology(uuid) Note: UC Ob provides alternative means to retrieve the topologies. There is no current use for the nw-topology-service.	GET	<Optional>
/tapi-common:context/tapi-topology:topology-context/topology={uuid} Notes: the GET operation for a whole topology has potential scalability issues.	GET	<Optional>
/tapi-common:context/tapi-topology:topology-context/topology={uuid}?depth=n Note: usage of depth in nodes, unless covered by a given UC may provide ambiguous responses (sliced and/or incomplete object fragments). Overall recommendation is to specify the list of requested fields and to perform more specific GET operations.	GET	<Optional>
/tapi-common:context/tapi-topology:topology-context/topology={uuid}?fields=uuid;name;layer-protocol-name	GET	UC 0b
/tapi-common:context/tapi-topology:topology-context/topology={uuid}?fields=node(uuid)	GET	UC 0b
/tapi-common:context/tapi-topology:topology-context/topology={uuid}?fields=link(uuid)	GET	UC 0b
/tapi-common:context/tapi-topology:topology-context/topology={uuid}/node={uuid}	GET	UC 0b
/tapi-common:context/tapi-topology:topology-context/topology={uuid}/node={uuid}?fields=owned-node-edge-point(uuid)	GET	UC 0b

/tapi-common:context/tapi-topology:topology-context/topology={uuid}/link={uuid}	GET	UC 0b
/tapi-common:context/tapi-topology:topology-context/topology={uuid}/node={uuid}/owned-node-edge-point={uuid}	GET	UC 0b
/tapi-common:context/tapi-topology:topology-context/topology={uuid}/node={uuid}/owned-node-edge-point={uuid}/name=INVENTORY_ID/value	GET	UC4
/tapi-common:context/tapi-topology:topology-context/topology={uuid}/node-uuid}/owned-node-edge-point={uuid}/inter-domain-plug-id-pac	GET	UC 0d
/tapi-common:context/tapi-topology:topology-context/topology={uuid}/node={uuid}/owned-node-edge-point={uuid}/tapi-connectivity:cep-list	GET	Future candidate if scale issue
/tapi-common:context/tapi-topology:topology-context/topology={uuid}/node={uuid}/owned-node-edge-point={uuid}/tapi-connectivity:cep-list/connection-end-point={uuid}	GET	UC 17b
/tapi-common:context/tapi-connectivity:connectivity-context Notes: the GET operation for the whole connectivity context has potential scalability issues. No UC addresses PUT or PATCH for the whole context.	~~GET~~, POST, ~~PUT, PATCH~~	All provisioning use cases.
/tapi-common:context/tapi-connectivity:connectivity-context?fields=connectivity-service(uuid)	GET	UC 0c
/tapi-common:context/tapi-connectivity:connectivity-context?fields=connection(uuid)	GET	UC 0c
/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service={uuid} Note: PATCH operation is unspecified	GET, PUT, DELETE, ~~PATCH~~	UC 0c, UC 10, UC 11a, UC 11b
/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service={uuid}/tapi-connectivity:topology-constraint/tapi-connectivity:include-path/path-uuid={puuid}	PUT	UC 6b
/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service={uuid}?fields=connection(connection-uuid)	GET	UC 0c
/tapi-common:context/tapi-connectivity:connectivity-context/connection={uuid}	GET	UC 0c
/tapi-common:context/tapi-connectivity:connectivity-context/connection={uuid}/physical-route-list	GET	UC 0c.1
/tapi-common:context/tapi-equipment:physical-context?fields=device(uuid)	GET (Added 1.1)	UC 4b
/tapi-common:context/tapi-equipment:physical-context/device={uuid}	GET (Added 1.1)	UC 4b
/tapi-common:context/tapi-equipment:physical-context?fields=physical-span(uuid)	GET (Added 1.1)	UC 4b
/tapi-common:context/tapi-equipment:physical-context/physical-span={uuid}	GET (Added 1.1)	UC 4b
/tapi-common:context/tapi-equipment:physical-context/device={uuid}?fields=equipment(uuid)	GET (Added 1.1)	UC 4b
/tapi-common:context/tapi-equipment:physical-context/device={uuid}/equipment={uuid}	GET (Added 1.1)	UC 4b
/tapi-common:context/tapi-path-computation:path-computation-context Notes: the GET operation for the whole context has potential scalability issues. No current UC for PUT and PATCH targeting the whole context object.	GET, POST, ~~PUT, PATCH~~	<Optional>
/tapi-common:context/tapi-path-computation:path-computation-context/path-comp-service={uuid} Note: PATCH operation is unspecified	GET, PUT, DELETE, ~~PATCH~~	<Draft> UC 12a, UC 12b, UC 12c
/tapi-common:context/tapi-path-computation:path-computation-context/path-comp-service={uuid}?fields=path	GET (see Use case 12.a)	<Draft>

		UC 12a, UC 12b, UC 12c
/tapi-common:context/tapi-path-computation:path-computation-context/path={uuid} Although "path computation service"-related use cases are considered draft, constrained provisioning of connectivity services MAY include TAPI path uuids (See [TAPI-CONN-MODEL-REQ-25]). In consequence, implementations MUST support the GET of a path object by its uuid.	GET	UC 3.X (Constrained provisioning)
/tapi-common:context/tapi-notification:notification-context	POST, GET	UC13a
/tapi-common:context/tapi-notification:notification-context/notif-subscription={uuid} Note: PATCH operation is unspecified	GET, PUT, DELETE, ~~PATCH~~	UC 13-16
/tapi-common:context/profile={{uuid}}	GET	UC12d UC17a
/tapi-common:context?fields=profile(uuid)	GET	UC17a
/tapi-common:context/tapi-oam:oam-context?fields=oam-service(uuid)	GET	UC 17a
/tapi-common:context/tapi-oam:oam-context/oam-service={{uuid}}	GET	UC 17a
/tapi-common:context/tapi-oam:oam-context?fields=oam-job(uuid)	GET	UC17a
/tapi-common:context/tapi-oam:oam-context/oam-job={{uuid}}	GET	UC17a
/tapi-common:context/tapi-oam:oam-context?fields=meg(uuid)	GET	UC17a
/tapi-common:context/tapi-oam:oam-context/meg={{uuid}}	GET	UC17a
/tapi-common:context/tapi-oam:oam-context/meg={{uuid}}?fields=mep(local-id)	GET	UC17a
/tapi-common:context/tapi-oam:oam-context/meg={{uuid}}?fields=mip(local-id)	GET	UC17a
/tapi-common:context/tapi-oam:oam-context/meg={{uuid}}/mip={{local-id}}	GET	UC17a
/tapi-common:context/tapi-oam:oam-context/meg={{uuid}}/mep={{local-id}}	GET	UC17a
/tapi-common:context	POST	UC17c
/tapi-common:context/tapi-oam:oam-context	POST	UC17d UC17e
/tapi-common:context/tapi-topology:topology-context/topology={{uuid}}/node={{node-uuid}}/owned-node-edge-point={{nep-uuid}}/tapi-connectivity:cep-list/connection-end-point={{cep-uuid}}/tapi-oam:mep-mip-list	GET	UC17b
/tapi-common:context/tapi-topology:topology-context/topology={topo-uuid}/node={node-uuid}/owned-node-edge-point={nep-uuid}/tapi-connectivity:cep-list/connection-end-point={cep-uuid}/tapi-digital-otn:odu-connection-end-point-spec/odu-term-and-adapter/odu-mep	GET	UC17b

NOTES:

1) RESTCONF allows a GET operation on a list (the target resource is a list or leaf-list, e.g., GET **/tapi-common:context/service-interface-point**) but it is only valid using JSON encoding, since well-formed XML does not allow multiple root elements. In view of this, this document no longer requires the implementation of GET directly targeting a list resource.

2) If a client wishes to retrieve a list, the implementation MUST support a GET operation on the list parent data node (e.g., usually a container) and the client MAY specify a ***fields*** and or ***depth*** query parameter. In consequence, while it is not mandatory to support e.g., GET **/tapi-common:context/service-interface-point** it is mandatory to support GET **/tapi-common:context?fields=service-interface-point** as shown.

3) In particular, the following calls are no longer mandatory. An implementation MAY chose to implement them assuming a JSON encoding.

API Entry	RESTCONF operation optionally allowed
/tapi-common:context/service-interface-point	GET
/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service	GET
/tapi-common:context/tapi-connectivity:connectivity-context/connection	GET
/tapi-common:context/tapi-topology:topology-context/topology	GET
/tapi-common:context/tapi-topology:topology-context/topology={uuid}/node	GET
/tapi-common:context/tapi-topology:topology-context/topology={uuid}/link	GET
/tapi-common:context/tapi-topology:topology-context/topology={uuid}/node={uuid}/owned-node-edge-point	GET
/tapi-common:context/tapi-path-computation:path-computation-context/path-comp-service	GET
/tapi-common:context/tapi-path-computation:path-computation-context/path	GET
/tapi-common:context/tapi-notification:notification-context/notif-subscription	GET

4) An implementation of TAPI/RESTCONF potentially allows / defines a much wider set of API entries /paths. The previous table aims a providing a reduced implementation scope.

5) The current minimum subset does not include calls related to OAM or inventory (equipment) aspects. The addition of additional entries is for further consideration.

4 Network Topology Model

Due to the need of composing a unified view of the network resources along different TAPI implementations, some guidelines are required to constrain the possibilities or interpretations of the models. The topology model MUST provide the explicit multi-layer topology representation of the L0-L1 network including Physical Media, OTS, OMS, MC, OTSiMC, OTSi/OTSiA, OTU, ODU, and DSR considerations.

Summary of changes for TAPI 2.4.0 and RIA 2.0 for layering (layer names and layer protocol qualifiers):

- The PHOTONIC_LAYER_QUALIFIER_MC and PHOTONIC_LAYER_QUALIFIER_OTSiMC layer-protocol-qualifier were introduced in TAPI v2.1.3, replacing PHOTONIC_LAYER_QUALIFIER_SMC and PHOTONIC_LAYER_QUALIFIER_NMC, respectively. In TAPI v2.4, OTSiMC is bound to a single OTSi while MC represents a generic media channel.

- The PHOTONIC_LAYER_QUALIFIER_{ SMC, OMSA, OTSA, OTS_OMS } layer qualifiers are **deprecated**.

- The PHOTONIC_LAYER_QUALIFIER_{ OCH, NMC, OTSi, OTSiA } layer qualifiers **are not used** *(candidates for future deprecation).* This RIA mandates the use of OTSiMC which integrates the ITU-T OTSi and MC concepts (as well as the OCH).

- The PHOTONIC_LAYER_QUALIFIER_{MCA, OTSiMCA} when applied to *ROADM-to-ROADM* scenarios are **left for further study**.

- The PHOTONIC_LAYER_QUALIFIER_{OTSiA, OTSiMCA} when applied to *Transceiver-to-Transceiver* scenarios are **left for further study.** This RIA only considers the provisioning of assemblies indirectly via the provisioning of client services (ODU/OTU). The direct provisioning of OTSiA, OTSiMCA services may apply in support of other clients not covered by this RIA.

- Use the new DIGITAL_OTN TAPI layer protocol name that models the OTU/ODU G.872 layers. The use of ODU TAPI layer protocol name is deprecated.

- Use the newly introduced tapi-digital-otn:OTU_TYPE identity (extending the LAYER_PROTOCOL_QUALIFIER) as well as OTU_TYPE_OTU1, OTU2, OTU3, OTU4 and OTU_CN identities.

- Use the newly introduced protocol qualifier PHOTONIC_LAYER_QUALIFIER_OTS_MEDIA. It is intended to replace and clarify the use of OTS and UNSPECIFIED protocol layer qualifiers while avoiding an excessive number of NEP/CEPs (i.e., avoid duplication of OTS and PHYSICAL MEDIA)

- The PHOTONIC_LAYER_QUALIFIER_OTSiMC protocol layer qualifier potentially includes information on the OTSi signal at the termination point (with electrical/optical conversion).

Based on ONF TAPI 2.4.0 models, a topology abstraction view is described for vendor agnostic integration across management/control systems in the frame of the proposed architecture in Section 3. The **TAPI Topology Flat Abstraction model** collapses *all layers in a single multi-layer topology instance*. The nomenclature **T0 – Multi-layer topology** and **T0** is used interchangeably to reference this topology in the remaining document.

4.1 Model Requirements

To properly describe the topology abstraction model proposed, the following requirements are listed. To help clarify such requirements, please consider the YANG tree snippet below.


```
module: tapi-topology
  augment /tapi-common:context:
    +--rw topology-context
       +--ro nw-topology-service
       |  +--ro topology* [topology-uuid]
       |  |  +--ro topology-uuid
       |          -> /tapi-common:context/tapi-topology:topology-context/topology/uuid
       |  +--ro uuid?       uuid
       |  +--ro name* [value-name]
       |  |  +--ro value-name    string
       |  |  +--ro value?        string
    +--ro topology* [uuid]
```

[TAPI-TOP-MODEL-REQ-1] The single topology **(T0 – Multi-layer topology)** includes all network layers, DSR, DIGITAL_OTN (including ODU and OTU), as well as PHOTONIC_MEDIA (including OTSiMC, MC, OMS, and OTS_MEDIA). T0 is explicitly modelled as a ***tapi-topology:topology*** object. This topology MUST appear within ***tapi-topology:topology-context/topology*** list, and MAY optionally be referenced by the ***topology*** list within the ***nw-topology-service*** container.

Note that in this version of the RIA there are no defined uses for ***nw-topology-service.***

[TAPI-TOP-MODEL-REQ-2] The TAPI server MAY implement other topologies. This RIA does not specify uses for topologies other than T0. In case there are multiple topologies present, the **T0 - Multi-layer topology** MUST be uniquely identified via the TOPOLOGY_NAME (in the name value-pair) prefixed with **T0_.**

[TAPI-TOP-MODEL-REQ-3] Each SIP MUST have at least one NEP related to it.

[TAPI-TOP-MODEL-REQ-4] A SIP is thus logically mapped to topology NEPs through the ***tapi-topology:owned-node-edge-point/mapped-service-interface-point*** attribute.

```
  augment /tapi-common:context:
       +--ro topology* [uuid]
          +--ro node* [uuid]
          |  +--ro owned-node-edge-point* [uuid]
          |  |  +--ro mapped-service-interface-point* [service-interface-point-uuid]
          |  |  |  +--ro service-interface-point-uuid -> .../service-interface-point/uuid
```

4.1.1 TAPI Node NEP Forwarding Rules

It is possible to represent constrained forwarding capabilities between the NEPs of a node. This is modelled by using one or more ***node-rule-groups*** that, in turn contain one or more ***rules*** with a ***forwarding-rule*** (see yang-tree snippet). This feature can be useful in the case where an external path computation entity is used.

```
module: tapi-topology
augment /tapi-common:context:
       +--ro topology* [uuid]
          +--ro node* [uuid]
          |  +--ro node-rule-group* [uuid]
          |  |  +--ro rule* [local-id]
          |  |  |  +--ro rule-type?                    rule-type
          |  |  |  +--ro forwarding-rule?              forwarding-rule
          |  |  |  +--ro override-priority?            uint64
          |  |  |  +--ro cep-direction*                tapi-common:port-direction
          |  |  |  +--ro cep-port-role* []
          |  |  |  |  +--ro port-role*        port-role
          |  |  |  |  +--ro port-role-rule*   port-role-rule-option
          |  |  |  +--ro connection-spec-reference* []
          |  |  |  |  +--ro connection-spec-name?   string
          |  |  |  |  +--ro connection-spec?        tapi-common:uuid
          |  |  |  +--ro layer-protocol-qualifier*    tapi-common:layer-protocol-qualifier
          |  |  |  +--ro signal-property
          |  |  |  |  +--ro signal-property-name?         string
```



```
      |  |  |  |  +--ro signal-property-value-rule?   signal-property-value-rule
      |  |  |  |  +--ro applicable-signal-value*      string
      |  |  |  |  +--ro number-of-signal-values?      uint64
      |  |  |  +--ro complex-rule*                 string
      |  |  |  +--ro local-id                      string
      |  |  |  +--ro name* [value-name]
      |  |  |     +--ro value-name    string
      |  |  |     +--ro value?        string
      |  |  +--ro node-edge-point* [topology-uuid node-uuid node-edge-point-uuid]
      |  |  |  +--ro topology-uuid           -> ... topology/uuid
      |  |  |  +--ro node-uuid               -> ... topology/node/uuid
      |  |  |  +--ro node-edge-point-uuid    -> ... topology/node/owned-node-edge-point/uuid
      |  |  +--ro node-rule-group* [topology-uuid node-uuid node-rule-group-uuid]
               ...
```

To illustrate a possible use case, consider a transponder (modelled as a TAPI node) with multiple client ports and line ports (NEPs). The node-rule-groups may be useful to allow forwarding between client and line NEPs (i.e., with different layer-protocol-name and/or qualifiers) and to restrict forwarding between a pair or either client or line NEPs (i.e., with the same layer-protocol-name and/or qualifiers).

- In the former case, the allowed NEPs are grouped in a node-rule-group (node/node-rule-group) that contains a NEP list (node/node-rule-group/node-edge-point) and the node-rule-group contains a rule (node/node-rule-group/rule) with its **forwarding-rule MAY_FORWARD_ACROSS_GROUP.**

- In the latter case, the restricted NEPs are grouped in a node-rule-group with a rule with the **forwarding-rule CANNOT_FORWARD_ACROSS_GROUP.**

The following sections introduce a set of requirements on the NEP / CEP stacking for different scenarios. Please cfr. Section 5.2 for a description of applicable scenarios and illustrating figures.

4.1.2 DSR/DIGITAL_OTN Layers

[TAPI-TOP-MODEL-REQ-5] TAPI Nodes considered in this RIA MAY include DSR and/or DIGITAL_OTN capabilities, representing the mapping between DSR and DIGITAL_OTN NEPs (multi-layer) and the multiplexing/de-multiplexing across different ODU rates (multi-rate). Examples of such nodes can be transponder, muxponders or digital OTN switching functions.

For such NEPs, implementations MUST have the following allowed combinations:

- For the **layer-protocol-name,** either **DSR,** or **DIGITAL_OTN** as applicable.
  - For the DSR NEPs, they must support the instantiation of CEPs with layer protocol qualifiers being identities with base **tapi-dsr:DIGITAL_SIGNAL_TYPE** as allowed by hardware capabilities.
  - For the DIGITAL_OTN NEPs, they must support the instantiation of CEPs with layer protocol qualifiers being identities with base **tapi-digital-otn:ODU_TYPE** for the ODU layer qualifier(s) and with base **tapi-digital-otn:OTU_TYPE** for the OTU layer qualifier(s).

4.1.3 Digital to optical transition

[TAPI-TOP-MODEL-REQ-6] **[DEPRECATED]** [**transitional-link**] Transitional links are deprecated in this version of the RIA.

[TAPI-TOP-MODEL-REQ-7] The *digital to optical* transitions/adaptations MUST be represented by including a NEP that supports CEP instances with ***tapi-photonic-media:PHOTONIC_LAYER_QUALIFER_OTSiMC***. In such *terminated* CEPs the OTSi PAC MUST be present, and the OTSiMC PAC MAY be present (for example, to project the MC information bound to the

OTSi to the node modeling a transceiver device). The OTSi PAC represents the Trail Termination Points (TTPs) of the OTSiMC connections.
This implies NEP / CEP stacking with terminated OTSiMC CEP (for example, at the line port of an optical terminal such as transponders or muxponders).

[TAPI-TOP-MODEL-REQ-8] This optical line interfaces representation in terms of PHOTONIC_MEDIA NEPs shall be available immediately after the Optical Terminals commissioning stage and prior to any service deployment over the optical line interfaces.

4.1.4 OTSiMC/MC/OMS/OTS Photonic Media Layers

[TAPI-TOP-MODEL-REQ-9] The physical connectivity between transponder/muxponder line ports and ROADM/FOADM's add/drop ports MUST be represented as UNIDIRECTIONAL or BIDIRECTIONAL **tapi-topology:links** between **PHOTONIC_MEDIA NEPs.**

[TAPI-TOP-MODEL-REQ-10] PHOTONIC_MEDIA NEPs representing potential OTSiMC connectivity MUST be BIDIRECTIONAL.

[TAPI-TOP-MODEL-REQ-11] PHOTONIC_MEDIA NEPs representing potential OTSiMC connectivity at the transponder line port MUST be clients of the **layer-protocol-qualifier:PHOTONIC_LAYER_QUALIFIER_OTS_MEDIA** CEP(s) (via ***tapi-connectivity:connection-end-point/client-node-edge-point***). For ROADM Add/Drop ports, client NEPs of the OTS_MEDIA CEPs may support either OTSiMC or MC CEP qualifiers. *Note: future versions of this RIA MAY explicitly include the MC layer. This is for further study.*

[TAPI-TOP-MODEL-REQ-12] PHOTONIC_MEDIA NEPs representing potential physical connectivity between transponder/muxponders line ports and ROADM/FOADM add/drop ports MUST support CEP(s) with OTS_MEDIA protocol qualifier. Such NEPs MUST NOT include the ***tapi-photonic-media:ots-node-edge-point-spec*** container (*there is no OTS section between a transponder line port and add/drop port*).

[TAPI-TOP-MODEL-REQ-13] PHOTONIC_MEDIA NEPs supporting OTSiMC CEPs SHOULD include the ***tapi-photonic-media:media-channel-node-edge-point-spec*** to represent the supportable, available, and occupied media channel spectrum pool resources.

[TAPI-TOP-MODEL-REQ-14] In case Optical Line Protection systems (OLPs) are present, OLP functionality MUST be represented in the Photonic Media layer. The OLP MUST appear as a single node, logically part of the Optical Line System (for further description please see Use Case 5b).

[TAPI-TOP-MODEL-REQ-15] Nodes representing OLP, ROADM/FOADM and ILA devices MUST be linked by PHOTONIC_MEDIA links. The corresponding NEPs MUST support CEPs with OTS_MEDIA protocol qualifier.

[TAPI-TOP-MODEL-REQ-16] Each NEP at the photonic media layer (**layer-protocol-name= PHOTONIC_MEDIA**) MUST support at least one of the following protocol layer qualifiers: PHOTONIC_LAYER_QUALIFIER_OTSiMC, PHOTONIC_LAYER_QUALIFIER_MC, PHOTONIC_LAYER_QUALIFIER_OMS, PHOTONIC_LAYER_QUALIFIER_OTS_MEDIA.

[TAPI-TOP-MODEL-REQ-17] Media-Channel (MC) constructs represent a reserved portion of the spectrum to route one or more OTSi signals. An OTSiMC represents the actual portion of the spectrum assigned to a given OTSi (see Figure 4-1).

PHOTONIC_MEDIA/PHOTONIC_LAYER_QUALIFIER_OTS_MEDIA CEPs at the ROADM add/drop ports MUST support a NEP which, in turn, *supports a CEP of either PHOTONIC_LAYER_QUALIFIER_MC or PHOTONIC_LAYER_QUALIFIER_OTSiMC.*

A NEP supporting one or more OTSiMC CEPs MAY be optionally represented on top of each of the PHOTONIC_LAYER_QUALIFER_MC CEPs. Such OTSiMC CEPs provide monitoring information of the spectrum of an individual OTSi, and its inclusion depends on the HW monitoring capabilities.

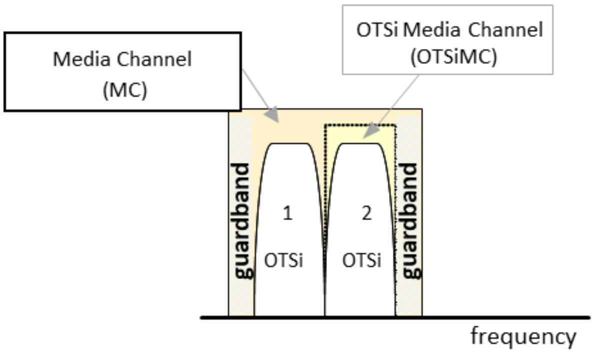


Figure 4-1 Media-channel entities relationship.

[TAPI-TOP-MODEL-REQ-18] Each **PHOTONIC_LAYER_QUALIFER_MC** NEP **MUST** include the ***tapi-photonic-media:media-channel-node-edge-point-spec*** to represent the media channel pool resources supportable, available. and occupied.

[TAPI-TOP-MODEL-REQ-19] This RIA mandates the representation of **tapi-topology:link** objects between PHOTONIC_MEDIA NEPs supporting PHOTONIC_LAYER_QUALIFIER_OTS_MEDIA CEPs. Such links MUST have **layer-protocol-name = PHOTONIC_MEDIA** as specified in Table 25. These links are not driven by services, they are configured in the network, and they MUST appear in the topology (in particular, in the absence of services). Note: other links (e.g., between NEPs at different protocol layers) MAY be present in the topology.

[TAPI-TOP-MODEL-REQ-20] In case OLP constructs are present for OMS or OTS protection, such construct MUST be represented in TAPI by instantiating a PHOTONIC_MEDIA link between involved ROADM degree ports and using ***tapi-topology:resilience-type/tapi-topology:protection-type*** link attribute (see UC.5a)

4.2 The use of INVENTORY_ID name in logical elements

Hardware identifiers currently stored in legacy OSS inventory systems MUST be correlated with TAPI UUID identifiers. This information will be provided by the SDN optical domain controller suppliers. For every inventory element represented as a logical element in TAPI by the SDN Domain controller, an **INVENTORY_ID** ***tapi-common:name*** property shall be included into the logical element construct.

The **INVENTORY_ID** tag SHALL be included for the following TAPI objects:

- ***tapi-topology:node-edge-point***
- ***tapi-common:service-interface-point***

The proposal for a common definition of the **INVENTORY_ID** tag, follows 2 main principles and it is based on [TMF-814] naming standards:

- It is explicit and clear: there is no ambiguity to which field each index correspond
- It can be augmented: if a new type of field needs to be inserted it does not break compatibility with the former format.

The generic format is the concatenation of *n* tuple elements “/<field>=<index>”. The supported fields for tuple elements are:

Table 9: Inventory-id fields format.

<field>	meaning
ne	Network Element
r	Rack
sh	Shelf
s_sh	Sub-shelf
sl	Slot
s_sl	Sub-slot
p	Port

The supported sequence for the tuple is the following and covers a variety of supported scenarios that may not all be applicable.

- [ ] means that may not be present
- […] means that multiple values can be specified (marked as green **x** in the matrix)

```
/ne=<nw-ne-name>[/r=<r_index>][/sh=<sh_index>[/s_sh=<s_sh_index>…]][[/sl=<sl_index>[/s_sl=<s_sl_index>…]][/p=<p_index> …]]
```

```
Inventory_ID ::= PortLocation... (separated by comma)

PortLocation ::= NetworkElement [Rack] [ Shell [ SubShell ] ] [Slot [SubSlot] ]  PortId
```

NOTE: An inventory ID is a list of port locations separated by comma

```
/ne=<nw-ne-name>         ;; Mandatory

         [/r=<r_index>]  ;; Rack

         [/sh=<sh_index>
                    [/s_sh=<s_sh_index> ...]
         ]
         [
                    [/sl=<sl_index>
                             [/s_sl=<s_sl_index> ...]
                    ]
                    [/p=<p_index> ...]
         ]
```

- **<nw-ne-name>** is the native **Network Element (NE)** name.
- **<r_index>** is the **Rack index**.
- **<sh_index>** is the **Shelf index**.
- **<s_sh_index>** is the **Sub-Shelf index**.
- **<sl_index>** is the **Slot index**.

- **<s_sl_index>** is the **Sub-Slot index**.
- **<p_index>** is the **Port index**.

Meaning for the port the following possible combinations depicted in the following matrix. Each column represents which tuples can be after the element listed in the first column.

Table 10: Inventory-id fields combination allowance.

	/r= <r_index>	/sh= <sh_index>	/s_sh= <s_sh_index>	/sl= <sl_index>	/s_sl= <s_sl_index>	/p= <p_index>
/ne=<nw-ne-name>	**X**	**x**	**-**	**x**	**-**	**x**
/r=<r_index>	**-**	**x**	**-**	**x**	**-**	**-**
/sh=<sh_index>	**-**	**-**	**x**	**x**	**-**	**-**
/s_sh=<s_sh_index>	**-**	**-**	**-**	**x**	**-**	**-**
/sl=<sl_index>	**-**	**-**	**-**	**-**	**x**	**x**
/s_sl=<s_sl_index>	**-**	**-**	**-**	**-**	**x**	**x**
/p=<p_index>	**-**	**-**	**-**	**-**	**-**	**-**

Some examples of INVENTORY_ID for the node-edge-points potentially mapped to the ports described in the examples shown in Figure 6-75 in Section 6.3.2.2 (the use of the INVENTORY_ID name does not exclude other value names that MAY be present):

Example 1:

```
"name": [
   {
    "value_name": "INVENTORY_ID",
    "value":"/ne=MadridNorte/r=1/sh=1/sl=1/s_sl=0"
    }
]
```

Example 2:

```
"name": [
   {
    "value_name": "INVENTORY_ID",
    "value": "/ne=MadridNorte/r=1/sh=2/sl=2/s_sl=1/p=3"
   }
]
```

Example 3:

```
"name": [
   {
```



```
      "value_name": "INVENTORY_ID",
      "value": "/ne=MadridNorte/r=1/sh=3/sl=7/s_sl=2/p=2"
     }
]
```

Example 4: (two ports) p=2 and p=7, may be different racks or not

```
"name": [
   {
     "value_name": "INVENTORY_ID",
     "value": "/ne=Barcelona/r=1/sh=3/sl=7/s_sl=2/p=2,/ne=Barcelona/r=1/sh=2/sl=4/s_sl=3/p=7"
    }
]
```


5 Connectivity service model

In this chapter, the complete connectivity service model will be described. The intention is to clarify some gaps which might not be clear just by reading the current description included in TAPI YANG models and to provide a uniform understanding on the use of the TAPI information models. Several reference design guidelines are stated to constrain the possibilities or interpretations of the current proposed models.

5.1 Model guidelines

The following guidelines MUST be met by all implementations compliant with the current specification. For the different guidelines and scenarios, this RIA follows the updated legend as per the Figure below:

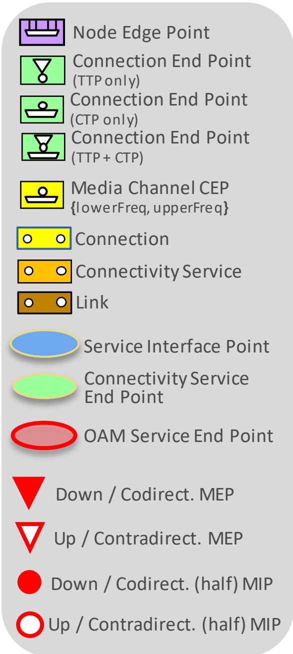


Figure 5-1 Legend used in the guidelines and scenarios

[TAPI-CONN-MODEL-REQ-1] [**creation**] The solution exposing the proposed NBI based on RESCONF/TAPI MUST expose the capability of creating Connectivity-Service(s) at the DSR, DIGITAL_OTN and PHOTONIC_MEDIA layers (see Section 3.2.1) as per the network capabilities. The provisioning of the Connectivity Service triggers the allocation of existing or newly created network resources by the TAPI server. Regarding the forwarding function, such allocation is modelled in terms of *Connections* at applicable Layer Protocol Name and Qualifiers [**supporting connections**].

[TAPI-CONN-MODEL-REQ-2] [**top-connection-def**] The connectivity model MUST include the concept of **Top Connection(s). A top connection is a connection object that represents connectivity at the highest level of partition (it is not a lower connection of another connection) and abstraction** for a given layer protocol name and qualifier supporting a given connectivity service**.**

- A top connection *commonly* spans two or more nodes (has *bounding CEPs* in different nodes) at the lowest partitioning level and usually represents end-end connectivity.

  It is possible that a top-connection spans a single node, such as a add/drop to add/drop local connection.

- Except in some specific cases, top-connections are explicitly partitioned into lower connections. In such case it is said the lower connections *support the top-connection*. This RIA only considers a direct partitioning of top-connection into “cross-connections” (which span only a given node) [**cross-connection**]. Note that the ONF Core IM contemplates the notion of “embedded” or “encapsulated” cross-connection as a fixed cross-connection that is internal to the CEP modelling. Those cross-connections are not explicitly represented in the data model and are not explicitly listed in the top-connection lower connections list corresponding to the aforementioned partitioning, see Figure 5-2.

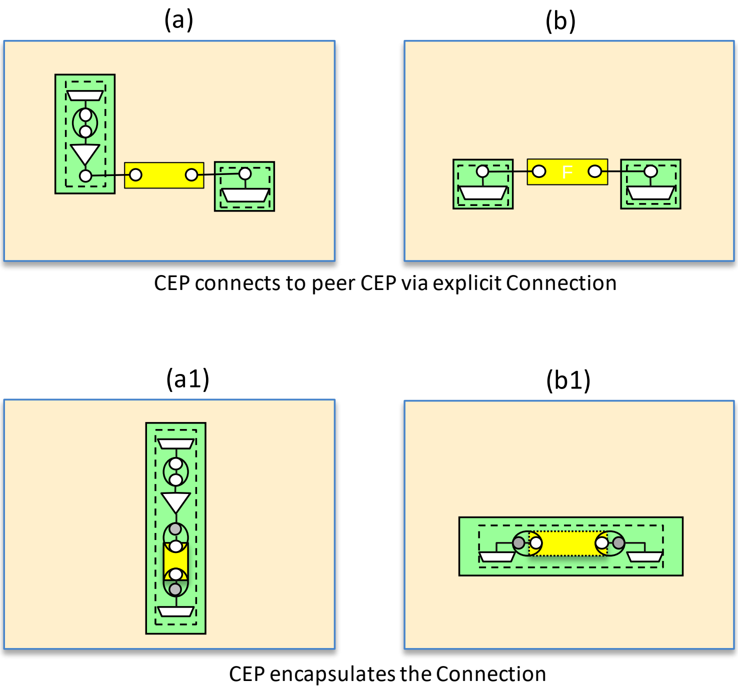


Figure 5-2 Explicit and encapsulated connections

Notes:

1) The partition in terms of cross-connections also applies to top-connections that spans only one node.

2) In some cases, a top connection may not have any lower connections. For example, a DSR top-connection where there is no switching flexibility at the DSR level, or an OTSiMC top-connection without explicit OTSiMC cross-connections at the ROADMs or OTU top-connections.

- Top-Connections can be either *terminated* (“infrastructure trails”), *non-terminated (*connecting client signals) or *semi-terminated* (asymmetric scenarios). A connectivity service for a given Layer Protocol Name and Qualifier relies on a **single** top-connection at that Layer Protocol Name and Qualifier [**immediate-top-connection**].[11] and may rely on an arbitrary number of top-connections for the server layers. For the former, immediate top-connection, each CEP is instantiated on top of the NEP/CEP stack that includes the NEP bound to the SIP that a CSEP references.

- In this line, some scenarios may involve, for example, a “terminated” top-connection that logically *extends* a “unterminated” top-connection at the same layer protocol name and qualifier. In this case, two top-connections exist yet both of them only list the corresponding cross-connections. In other words, there are no intermediate partitioning schemes in which the terminated top-connection refers to the

[11] Note that scenarios not covered by this RIA may address the 4-ended protected services, in which a connectivity service relies on multiple (e.g., 2) *immediate top-level connections*.

unterminated one as one of its lower connections. In other words, there is no explicit relationship between the non-terminated and the terminated top-connections.

For a given connectivity service, this RIA considers that it is supported by both top-connections and cross-connections.

[TAPI-CONN-MODEL-REQ-3] [**top-connection-ref**] A ***tapi-connectivity:connectivity-service*** MUST, after being successfully provisioned by the TAPI Server, include a reference to the *Immediate Top Connection* (***tapi-connectivity:connection)*** *and* MAY add additional supporting top-connections in its *connection list* (***tapi-connectivity:connectivity-service/connection***). These connections describe the end-to-end connectivity across the network at every network layer traversed by the connectivity-service (represented as the combination of the ***tapi-common:layer-protocol-name*** and ***tapi-common: layer-protocol-qualifier*** parameters). [*Note: In previous versions of the RIA, it was required to include all top-level connections – down to the MC layer --, this restriction has been relaxed. It is now preferred to use the* ***server-connection*** *attribute of each top-level connection if applicable.*].

EXAMPLE: Starting from time zero scenario of Figure 5-3, consider the TAPI client provisioning unterminated CS1 and unterminated CS2, which causes the instantiation of their corresponding immediate top-connections, see Figure 5-4. Each unterminated top-connection shall list only the relevant cross-connections of the forwarding domains (nodes) it spans. Later, the TAPI client provisions the over-arching, semi-terminated CS and refers to CS1 and CS2 in the coroute-inclusion constraints, see Figure 5-5. This triggers the instantiation of the semi-terminated immediate top-connection. The semi-terminated top-connection shall include all (pre-existing) cross-connections that support the unterminated top-connections, as well as the additional instantiated cross-connections (termination and stitching) only in its lower-connections list (and not the unterminated top-connections). At this point all (pre-existing) cross-connections are owned by both the semi-terminated CS as well as the corresponding unterminated CS (in other words, the cross-connections are listed as lower-connections by both the semi-terminated top-connection as well as the corresponding unterminated top-connection). The semi-terminated CS MUST list the semi-terminated immediate top-connection in its connection list (and MAY list additional server layer top-connections) but MUST NOT list the unterminated top-connections 1 and 2, which are only listed as immediate top-connection of their respective CS1 and CS2. Note that (see UC10 on service deletion) that it is possible to delete either unterminated CS1 or CS2 before deleting the semi-terminated CS and it would cause the deletion of the corresponding unterminated top-connection, yet the supporting cross-connections would not be removed since they are co-owned by the semi-terminated connectivity service.

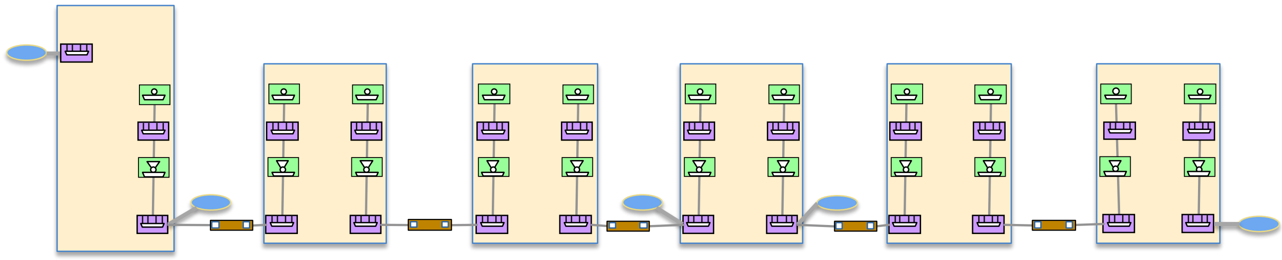


Figure 5-3 Unterminated Connection, time zero

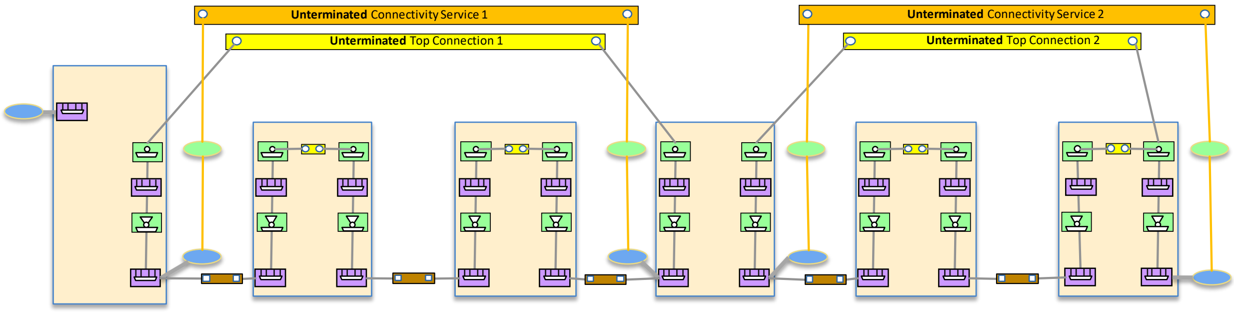


Figure 5-4 Unterminated Connection, unterminated CSs and Connections

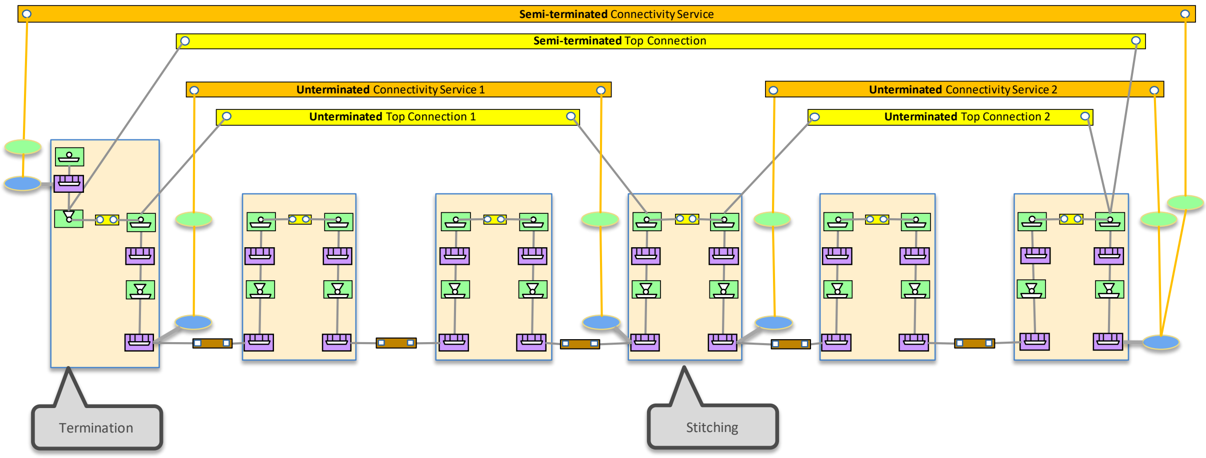


Figure 5-5 Unterminated Connection, semi-terminated CS and Connection

NOTES:

1/ When adding the list of top-level connections to a connectivity service, the RECOMMENDED order is to add items from the highest protocol and qualifier to the lowest and, for a given layer, from aEnd to zEnd.

2/ This RIA does not mandate the listing of layers below the MC.

3/ It is acknowledged that maintaining this list of supporting connections has redundancy and scalability issues: some connections (e.g., OMS/OTS) systematically appear, the same captured relationship(s) can also be obtained via the corresponding CEP/NEP/CEP stacking and, to comply to this requirement, implementations need to perform additional costly consistency checks when the underlying connections change (e.g., after a reroute).

```
module: tapi-connectivity
  augment /tapi-common:context:
 +--rw connectivity-context
        +--rw connectivity-service* [uuid]
```



```
        |  +--ro connection* [connection-uuid]
        |  |  +--ro connection-uuid -> ...connectivity:connectivity-context/connection/uuid
```

[TAPI-CONN-MODEL-REQ-4] [**route**] Each Top Connection object MUST represent how the requested service has been implemented within its network layer/qualifier. It shall include one or more ***tapi-connectivity:connection/route*** object containing the list of connection-end-points (CEPs) as per Section 3.2.7.3.

```
module: tapi-connectivity
  augment /tapi-common:context:
 +--rw connectivity-context
     +--ro connection* [uuid]
         +--ro route* [local-id]
          |  +--ro connection-end-point* [
                    topology-uuid node-uuid node-edge-point-uuid connection-end-point-uuid]
          |  |  +--ro topology-uuid
                      -> ...topology-context/topology/uuid
          |  |  +--ro node-uuid
                      -> ...topology-context/topology/node/uuid
          |  |  +--ro node-edge-point-uuid
                      -> ...node/owned-node-edge-point/uuid
          |  |  +--ro connection-end-point-uuid
                      -> ...tapi-connectivity:cep-list/connection-end-point/uuid
...
```

[TAPI-CONN-MODEL-REQ-5] [**route-order**] The ***tapi-connectivity:connection/route*** is modelled as a YANG List object of CEP References which is, by default, ordered by the system (i.e., the TAPI server which produces it). **The TAPI Server SHALL maintain the logical order of the CEP**, as defined by the signal flow and the knowledge of the Topology information (links) and the NEP and CEP associations.

[TAPI-CONN-MODEL-REQ-6] Lower and Server Connections:

[**lower-connection**] Each Top Connection MUST include a reference to all the lower connections supporting it (in the same network layer and qualifier). These references MUST be included within the ***tapi-connectivity:connection/tapi-connectivity:lower-connection*** list. *Please note that the use of the lower-connection attribute is used to represent the partitioning of the Top Connection and does not introduce any layering relationship.*

```
module: tapi-connectivity
  augment /tapi-common:context:
 +--rw connectivity-context
     +--ro connection* [uuid]
        +--ro lower-connection* [connection-uuid]
           +--ro connection-uuid -> ...connectivity-context/connection/uuid
```

[**server-connection**] Each Top Connection MAY be supported by one or more (immediate) server layer top connection(s). In such case, the (client) top connection MUST include a reference to each immediately supporting server top connection(s) within the ***tapi-connectivity:connection/tapi-connectivity:server-connection*** list.

```
module: tapi-connectivity
  augment /tapi-common:context:
 +--rw connectivity-context
     +--ro connection* [uuid]
        +--ro server-connection* [connection-uuid]
           +--ro connection-uuid -> ...connectivity-context/connection/uuid
```


[TAPI-CONN-MODEL-REQ-7] [**top-connection**] Top Connections MAY represent two different cases:

- **Non-terminated Top Connections**: between CEPs with parent-NEPs (**tapi-topology:owned-node-edge-point/tapi-connectivity:cep-list/connection-end-point/parent-node-edge-point**) directly associated to the SIPs which has been referenced by the Connectivity-Service-End-Points of the Connectivity-Service associated to this Top Connection.

```
augment /tapi-common:context/tapi-topology:topology-context/tapi-topology:topology/tapi-
topology:node/tapi-topology:owned-node-edge-point:
    +--ro cep-list
       +--ro connection-end-point* [uuid]
          +--ro parent-node-edge-point
          |  +--ro topology-uuid?
                   -> ...topology-context/topology/uuid
          |  +--ro node-uuid?
                   -> ...topology-context/topology/node/uuid
          |  +--ro node-edge-point-uuid?
                   -> ...topology-context/topology/node/owned-node-edge-point/uuid
```

- **Infrastructure Trails as defined in [ITU-T G.805]**: between CEPs representing Trail Termination Points (TTPs) which handover a signal of a given layer to a higher layer. These CEPs also produce associated client-NEPs (**tapi-topology:owned-node-edge-point/tapi-connectivity:cep-list/connection-end-point/client-node-edge-point**), to represent the generated pool of resources at a higher network layer or rate.

```
augment /tapi-common:context/tapi-topology:topology-context/tapi-topology:topology/tapi-
topology:node/tapi-topology:owned-node-edge-point:
    +--ro cep-list
       +--ro connection-end-point* [uuid]
          +--ro client-node-edge-point* [topology-uuid node-uuid node-edge-point-uuid]
          |  +--ro topology-uuid
                     -> ...topology-context/topology/uuid
          |  +--ro node-uuid
                     -> ...topology-context/topology/node/uuid
           |  +--ro node-edge-point-uuid
                     -> ...topology-context/topology/node/owned-node-edge-point/uuid
```

5.1.1 Multi-layer connectivity service provisioning and connection generation

The TAPI server MUST include a reference to the immediate layer Top Connection within a Connectivity Service's Connection list (referenced within the ***tapi-connectivity:connectivity-service/connection*** list attribute) and need not include other supporting top-level connections (optional). Therefore, the Connectivity Service routing across different layers (identification of all supporting connections) cannot be inferred *only* by means of such list along with their respective lower-connections, but also requires retrieving each top-level connection ***tapi-connectivity:connection/server-connection*** list.

Note that it is also possible to determine the supporting connections by the tapi-topology - tapi-connectivity model relationships (*known as NEP/CEP stacking*). These relationships are described in the following requirements:

[TAPI-CONN-MODEL-REQ-8] Every layer-protocol or layer-protocol-qualifier transition MUST be represented as a stack of ***tapi-topology:node-edge-point*** and ***tapi-connectivity:connection-end-points*** related to each other by ***tapi-connectivity:connection-end-point/parent-node-edge-point*** and ***tapi-connectivity:connection-end-point/client-node-edge-point*** parameters:

```
augment /tapi-common:context/tapi-topology:topology-context/tapi-topology:topology/tapi-
topology:node/tapi-topology:owned-node-edge-point:
    +--ro cep-list
       +--ro connection-end-point* [uuid]
```



```
          +--ro parent-node-edge-point
          |  +--ro topology-uuid?          -> ...topology-context/topology/uuid
          |  +--ro node-uuid?           -> ...topology-context/topology/node/uuid
          | +--ro node-edge-point-uuid? -> ...node/owned-node-edge-point/uuid
          |
          +--ro client-node-edge-point* [topology-uuid node-uuid node-edge-point-uuid]
          |  +--ro topology-uuid           -> ...topology-context/topology/uuid
          |  +--ro node-uuid               -> ...topology-context/topology/node/uuid
          |  +--ro node-edge-point-uuid    -> ...node/owned-node-edge-point/uuid
```

[TAPI-CONN-MODEL-REQ-9] Additionally, if a ***tapi-topology:link*** object is generated to represent the adjacency between a pair of NEPs that results from a Top-Connection object, such link MUST be referenced by the ***tapi-connectivity:connection/supported-client-link*** attribute.

```
module: tapi-connectivity
  augment /tapi-common:context:
 +--rw connectivity-context
     +--ro connection* [uuid]
           +--ro supported-client-link* [topology-uuid link-uuid]
            |  +--ro topology-uuid
            |  +--ro link-uuid
```

5.1.2 Relationship CS and Top-Level Connections for DSR Connectivity Services

The following set of guidelines detail the process when a ***DSR connectivity service*** has been requested, including the different layer connections and how they are *instantiated.*

Notes:

1) This process assumes the encapsulation of a DSR signal into a Low Order (LO)-ODU signal and the multiplexing of the (LO)-ODU signal into High Order (HO)-ODU signals.

2) In this section *instantiation* means the managed object appears in the RESTCONF datastore of the TAPI Server.

5.1.2.1 Initial considerations regarding connection creation order

Previous versions of this RIA specified the order in which connections were inserted in their respective lists (e.g., Connectivity Service *connections* list; connection *server-connection* lists, etc.) and the order they were expected to become operational. It is now acknowledged that connections may be appear on the datastore at arbitrary times and with diverse states.

This RIA only specifies that :

- After a successful POST (the server returns an *HTTP 201 Created* response code, including a “Location” header) it means that the connectivity service has been instantiated (in the RESTCONF sense, the arguments were valid and the datastore contains the CS). This stage does not necessarily include path computation or resource allocation (in other words, we do neither impose nor forbid a synchronous approach)

- State changes in the Connectivity Service (and supporting connections) are, by definition, asynchronous. Clients are expected to determine state (either by polling using subsequent GETs or via notification / streaming processing).

- When the connectivity service operational-state (***tapi-connectivity:connectivity-service/tapi-common:operational-state)*** changes to ENABLED, the client is informed that the service is OPERATIONAL. It is responsibility of the TAPI server to derive the state from the state of each supporting resources (*supporting top-connection(s)*, other connections, CEPs, NEPs...). The client is thus not required to check for the operational state of such supporting resources.

At DSR layer:

[TAPI-CONN-MODEL-REQ-10] The CS triggers the creation of the Top Connection at the DSR layer:

- The DSR top-connection **MUST** be inserted in the CS connection list.
- The DSR top-connection MUST include its route as per [TAPI-CONN-MODEL-REQ-5].

[TAPI-CONN-MODEL-REQ-11] If one or more DSR XC Connections are *instantiated* (describing the lower partitioning level of DSR Top Connection), they MUST be included within the top-connection lower-connection list.

At the DIGITAL_OTN layer the DSR CS triggers the creation of (or the reuse of):

[TAPI-CONN-MODEL-REQ-12] 1-N_LO Top Connections at the LO-ODUj rate (ODU-j) layer qualifiers (*e.g., due to intermediate DSR switching or DSR resilience*)

- The ODU-j Top Connection(s) MAY be included within the CS connection list.
- The ODU-j Top Connection(s) MUST be included within the DSR top-connection server-connection list.
- Each ODU-j Top Connection MUST include the corresponding list of ODU-j lower connections.
- After the instantiation of the ODU-j Top-Connection(s) the immediately upper layer adjacency is defined (a higher layer NEP is created "over" the CEP) allowing the upper layer Top Connection to be realized.
- After the instantiation of the ODU-j Top-Connection(s), a new ***tapi-topology:link*** at the DSR layer (**layer-protocol-name=DSR**) **MAY be** generated between the DSR NEPs on top of the ODU-j CEPs (Trail Termination Points) and referenced by the ***tapi-connectivity: supported-client-link*** attribute of such top-connections**.**

[TAPI-CONN-MODEL-REQ-13] 1-N_HO Top Connection(s) at the HO-ODUk rate (ODU-k), which describe the highest order ODU which are transported by the OTU layer.

- The ODU-k Top Connection(s) MAY be included within the CS connection list.
- The ODU-k Top Connection(s) MUST be included within the *corresponding* ODU-j top-connection server-connection list.
- Each ODU-k Top Connection MUST include the corresponding list of ODU-k lower connections.
- After the instantiation of the ODU-k Top-Connection(s) the immediately upper layer adjacency is defined (a higher layer NEP is created "over" the CEP) allowing the upper layer Top Connection to be realized.
- After the instantiation of the ODU-k Top-Connection(s), a new ***tapi-topology:link*** at the DIGITAL_OTN layer **MAY be** generated between the DIGITAL_OTN NEPs on top of the ODU-k CEPs (Trail Termination Points) and referenced by the ***tapi-connectivity:supported-client-link*** attribute of such top-connections**.**

[TAPI-CONN-MODEL-REQ-14] 1-N_OTU Top Connection(s) at the OTU, which describe the OTU which are transported by the optical OTSiMC layer.

- The OTU Top Connection(s) MAY be included within the CS connection list.
- The OTU Top Connection(s) MUST be included within the *corresponding* ODU-k top-connection server-connection list.
- After the instantiation of the OTU Top-Connection(s) the immediately upper layer adjacency is defined (a higher layer NEP is created "over" the CEP) allowing the upper layer Top Connection to be realized.
- After the instantiation of the OTU Top-Connection(s), a new ***tapi-topology:link*** at the DIGITAL_OTN layer **MAY be** generated between the DIGITAL_OTN NEPs on top of the OTU CEPs (Trail Termination Points) and referenced by the ***tapi-connectivity: supported-client-link*** attribute of such top-connections**.**

At the PHOTONIC_LAYER_QUALIFIER_OTSiMC layer the CS triggers the creation of:

[TAPI-CONN-MODEL-REQ-15] One or more Top Connection(s) between the OTSiMC CEPs over PHOTONIC_MEDIA NEPs (the OTSiMC CEPs supporting the DIGITAL_OTN NEPs).

- The OTSiMC Top Connection(s) MAY be included within the CS connection list.
- The OTSiMC Top Connection(s) MUST be included within the corresponding OTU top-connection server-connection list.
- Each OTSiMC Top Connection MUST include the corresponding list of OTSiMC lower connections.
- After the instantiation of the OTSiMC Top-Connection(s) the immediately upper layer adjacency is defined (a DIGITAL_OTN NEP supporting OTU CEPs is created "over" the OTSiMC CEP) allowing the upper layer Top Connection to be realized.
- After the instantiation of the OTSiMC Top-Connection(s), a DIGITAL_OTN ***tapi-topology:link*** between the related DIGITAL_OTN (OTU) NEPs **MAY be** generated. If generated, the new link MUST be referenced by the OTSiMC Top-Connection(s), which realizes it, as a ***tapi-connectivity: supported-client-link.***

[TAPI-CONN-MODEL-REQ-16] **[DEPRECATED]** This version of the RIA covers multiple OTSiMC cross-connections.

At the PHOTONIC_LAYER_QUALIFER_MC layer the DSR CS triggers the creation (or reuse) of:

[TAPI-CONN-MODEL-REQ-17] Zero or more PHOTONIC_LAYER_QUALIFER_MC Top Connections. Note that it is possible to have a scenario with only OTSiMC switching (see, for example, Figure 5-32).

- The MC Top Connection(s) MAY be included within the CS connection list.
- The MC Top Connection(s) MUST be included within the corresponding OTSiMC top-connection server-connection list.
- Each MC Top Connection MUST include the corresponding list of MC lower connections.

- After the instantiation of the MC Top-Connection(s) the immediately upper layer adjacency is defined (a PHOTONIC_MEDIA NEP supporting OTSiMC CEPs is created "over" the MC CEP) allowing the upper layer Top Connection to be realized.
- After the instantiation of the MC Top-Connection(s), a PHOTONIC_MEDIA ***tapi-topology:link*** between the related PHOTONIC_MEDIA (OTSiMC) NEPs **MAY be** generated. If generated, the new link MUST be referenced by the MC Top-Connection(s), which realizes it, as a ***tapi-connectivity: supported-client-link.***

[TAPI-CONN-MODEL-REQ-18] **[DEPRECATED]** This version of the RIA covers multiple MC cross-connections.

Note that OTSiMC layer representation, including Top Connections, XCs and CEPs on top of an MC layer may be useful to reflect OTSiMC monitoring capabilities. When both layers are present both layers are congruent (see ROADM1 in Figure 5-21).

At the PHOTONIC_LAYER_QUALIFER_OMS layer

[TAPI-CONN-MODEL-REQ-19] Zero or more PHOTONIC_LAYER_QUALIFER_OMS Top-Connections are reused.

- The OMS Top Connection(s) MAY be included within the CS connection list.
- The OMS Top Connection(s) MUST be included within the corresponding MC top-connection server-connection list or in the OTSiMC top-connection server-list, as appropriate (see Figure 5-32).
- Each OMS Top Connection MUST include the corresponding list of OMS lower connections.
- For each of the OMS Top-Connection(s) the immediately upper layer adjacency is defined (a PHOTONIC_MEDIA NEP supporting MC CEPs is created "over" the OMS CEP) allowing the upper layer Top Connection to be realized.
- For each of the OMS Top-Connection(s), a PHOTONIC_MEDIA ***tapi-topology:link*** between the related PHOTONIC_MEDIA (MC or OTSiMC supporting) NEPs **MAY have been** generated. If generated, the new link MUST be referenced by the OMS Top-Connection(s), which realizes it, as a ***tapi-connectivity: supported-client-link.***

At the PHOTONIC_LAYER_QUALIFER_OTS_MEDIA layer

[TAPI-CONN-MODEL-REQ-20] Zero or more PHOTONIC_LAYER_QUALIFER_OTS_MEDIA Top-Connections are reused.

- The OTS_MEDIA Top Connection(s) MAY be included within the CS connection list.
- The OTS_MEDIA Top Connection(s) MUST be included within:
  - Where applicable, the corresponding OTSiMC top-connection server-connection list (e.g., in case of transceiver to transceiver), along with the supporting MC top-connections, if any (see Figure 5-32).
  - Where applicable, the corresponding MC top-connection server connection list (in case an MC connection starts at the transceiver line port).
  - The corresponding OMS top-connection server-connection list.
- For each of the OTS_MEDIA Top-Connection(s) the immediately upper layer adjacency is defined :
  - a PHOTONIC_MEDIA NEP supporting OMS CEPs (for example, in the case of ROADM degree ports) is created where applicable.

  - a PHOTONIC_MEDIA NEP supporting MC CEPs (for example, in the case of ROADM add/drop ports) is created where applicable.
  - a PHOTONIC_MEDIA NEP supporting OTSiMC CEPs (for example, in the case of ROADM add/drop port or Transceiver line port) where applicable.

  NEPs are created "over" the OTS_MEDIA CEP allowing the upper layer Top Connection to be realized (see Figures in Section 5.2)

- For each of the OTS_MEDIA Top-Connection(s), a PHOTONIC_MEDIA ***tapi-topology:link*** between the related PHOTONIC_MEDIA NEPs **MAY have been** generated. If generated, the new link MUST be referenced by the OTS_MEDIA Top-Connection(s), which realizes it, as a ***tapi-connectivity: supported-client-link.***

5.1.2.2 Example of encoding

The next fragment shows a partial view of a TAPI context highlighting a specific DSR connectivity-service as well as the involved connections, to clearly identify the connection hierarchy and navigation association described by the previous set of requirements.

```
{"tapi-common:context":{
 "tapi-connectivity:connectivity-context":{
        "connectivity-service":[
               {"uuid" : "CS_UUID",
                "end-point":[
                       {
                          "local_id" : "LOCAL_ID_A",
                          "service-interface-point":{
                            "service-interface-point-uuid" : <SIP_UUID_A>
                          }
                       },
                       {
                          "local_id" : "LOCAL_ID_B",
                          "service-interface-point":{
                            "service-interface-point-uuid" : <SIP_UUID_B>
                          }
                       }
                     ],
                "connection":[
                     {"connection-uuid":"DSR_TOP_1"},    /* mandatory */

                     {"connection-uuid":"ODUj_TOP_1"},   /* optional */
                      ...
                     {"connection-uuid":"ODUk_TOP_1"},
                      ...
                     {"connection-uuid":"OTSIMC_TOP_1"},
                      ...
                     {"connection-uuid":"MC_TOP_1"}
                      ...
                     {"connection-uuid":"OMS_TOP_1"}
                      ...
                     {"connection-uuid":"OTS_MEDIA_TOP_1"}

                ]
              }
        ],

        "connection":[
               {"uuid": "DSR_TOP_1",
                "lower-connection":[                     /* flexibility DSR switching */
```



```
                    {"connection-uuid":"DSR_XC_1"},
                    {"connection-uuid":"DSR_XC_2"}
              ],
              "server-connection":[
                    {"connection-uuid":"ODUj_TOP_1"},
                     ...
              ]
             },

             {"uuid": "ODUj_TOP_1",
              "lower-connection":[
                    {"connection-uuid":"ODUj_XC_1"},
                    {"connection-uuid":"ODUj_XC_2"},
                     ...
              ] ,
              "server-connection":[
                    {"connection-uuid":"ODUk_TOP_1"},
                     ...
              ]
             },

               … (repeated for N_LO ODUj layer rates)

               … (repeated for N_HO ODUk layer rates)


             {"uuid": "OTSiMC_TOP_1",
              "lower-connection":[
                    {"connection-uuid":"OTSiMC_XC_1"},
                    {"connection-uuid":"OTSiMC_XC_2"},
                           …
                    {"connection-uuid":"OTSiMC_XC_N"}
              ],
              "server-connection":[
                    {"connection-uuid":"MC_TOP_1"},
                     ...
              ]
             } ,
             {"uuid": "MC_TOP_1",
              "lower-connection":[
                    {"connection-uuid":"MC_XC_1"},
                    {"connection-uuid":"MC_XC_2"},
                           …
                    {"connection-uuid":"MC_XC_N"}
              ],
              "server-connection":[
                    {"connection-uuid":"OMS_TOP_1"},
                     ...
              ]
             },
               … (repeated for OMS layers)

               … (repeated for OTS layers)
       ]}
       }
}
```


[TAPI-CONN-MODEL-REQ-21] The relationship between client / server CS and the procedures and guidelines for CS deletion are given in Section 6.2 and UC-10 (service deletion).

5.1.3 Resiliency mechanism at connectivity service

[TAPI-CONN-MODEL-REQ-22] To implement different protection mechanisms the TAPI Server MUST support the following protection and restoration policies (***tapi-topology:protection-type***) at the Connectivity Service level as per applicable Use Cases and hardware capabilities:

- ONE_PLUS_ONE_PROTECTION
- ONE_PLUS_ONE_PROTECTION_WITH_DYNAMIC_RESTORATION
- ONE_PLUS_ONE_PROTECTION_WITH_PRE_COMPUTED_RESTORATION
- PERMANENT_ONE_PLUS_ONE_PROTECTION
- ONE_FOR_ONE_PROTECTION
- DYNAMIC_RESTORATION
- PRE_COMPUTED_RESTORATION

```
+--rw connectivity-context
      +--rw connectivity-service* [uuid]
          |  +--rw resilience-type
          |  |  +--rw restoration-policy?   restoration-policy
          |  |  +--rw protection-type? protection-type
```

[TAPI-CONN-MODEL-REQ-23] The TAPI server, for all protected services with restoration capabilities, SHALL implement the PER_DOMAIN_RESTORATION policy by default, which implies it is responsible of activating the required control mechanisms to guarantee the restoration of the service autonomously.

[TAPI-CONN-MODEL-REQ-24] At the Connection level, the switch control, which implements the route diversity for the different levels of protection policies listed above, MUST be implemented by the TAPI server. The TAPI server MUST be able to describe these mechanisms by the ***tapi-connectivity:connection/switch-control.***

```
module: tapi-connectivity
augment /tapi-common:context:
+--rw connectivity-context
   +--rw connection* [uuid]
      +--ro switch-control* [uuid]
          |  +--ro sub-switch-control* [connection-uuid switch-control-uuid]
          |  |  +--ro connection-uuid        -> ...connection/uuid
          |  |  +--ro switch-control-uuid    -> ...connection/switch-control/uuid
          |  +--ro switch* [local-id]
          |  |  +--ro selected-connection-end-point* [topology-uuid node-uuid ...]
          |  |  |  +--ro topology-uuid                -> ...topology-context/topology/uuid
          |  |  |  +--ro node-uuid                    -> ...topology-context/topology/node/uuid
          |  |  |  +--ro node-edge-point-uuid         -> ...topology-context/topology/node/owned-node-edge-point/uuid
          |  |  |  +--ro connection-end-point-uuid    -> ...tapi-connectivity:cep-list/connection-end-point/uuid
          |  |  +--ro selected-route* [connection-uuid route-local-id]
          |  |  |  +--ro connection-uuid    -> .../connection/uuid
          |  |  |  +--ro route-local-id     -> .../connection/route/local-id
          |  |  +--ro selection-reason?                selection-reason
          |  |  +--ro switch-direction?                tapi-common:direction
          |  |  +--ro local-id                         string
          |  |  +--ro name* [value-name]
          |  |     +--ro value-name    string
          |  |     +--ro value?        string
          |  +--ro control-parameters
          |  |  +--ro resilience-type
          |  |  |  +--ro restoration-policy?   restoration-policy
          |  |  |  +--ro protection-type?      protection-type
          |  |  +--ro restoration-coordinate-type?          coordinate-type
          |  |  +--ro fault-condition-determination?        fault-condition-determination
          |  |  +--ro restore-priority?                     uint64
          |  |  +--ro reversion-mode?                       reversion-mode
          |  |  +--ro wait-to-revert-time                   uint64
          |  |     +-- value?     uint64
          |  |     +-- time-unit? time-unit
          |  |  +--ro hold-off-time?                        uint64
```



```
        |   |   +--ro is-lock-out?                          boolean
        |   |   +--ro is-frozen?                            boolean
        |   |   +--ro is-coordinated-switching-both-ends?   boolean
        |   |   +--ro max-switch-times?                     uint64
        |   |   +--ro preferred-restoration-layer*          tapi-common:layer-protocol-name
        |   |   +--ro selection-control?                    selection-control
        |   |...
```

5.1.4 Connectivity, Routing, Topology and Resiliency constrains for connectivity services

[TAPI-CONN-MODEL-REQ-25] To implement different use cases that imply constraints on the connectivity service, several parameters of the ***tapi-connectivity: connectivity-service*** object MUST be supported, as required per each use case. See Section 6.2.1 for an overall definition of constraints and the different use cases.

5.2 TAPI overall network models

The following figures illustrate common scenarios including, for example, partial disaggregation. Note that this RIA does not specify layers above the DSR layer (e.g., UNI) thus DSR CEPs always being unterminated. It is shown the scenario at “time zero”, i.e., the model of logical resources made available by the server controller before any provisioning is performed by client controller, followed by examples of the possible provisioning scenarios.

5.2.1 Scenario 1 : Optical Line System Controller

Figure 5-6 illustrates a possible layering for an OLS controller at time zero. The OLS is composed of 3 ROADM nodes. The ROADMs are connected (degree to degree) via PHOTONIC links. There are OTS_MEDIA and OMS top connections between the ROADM degrees. There is no OMS CEP at the ROADM add/drop ports. SIPs are associated to PHOTONIC NEPs at ROADM add/drop ports.

Note that since the scope of the OLS controller is limited to the OLS/ROADMs, the link to the (undefined) clients is not available.

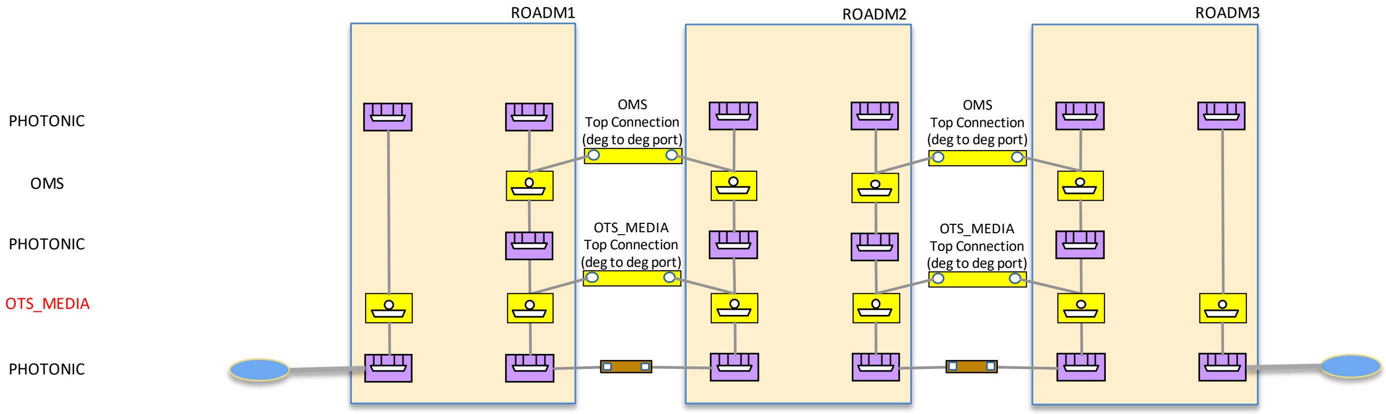


Figure 5-6 Scenario 1 : Optical Line System Controller, time zero

Figure 5-7 shows similar scenario including an In Line Amplifier. The amplifier node has an OMS cross-connection.

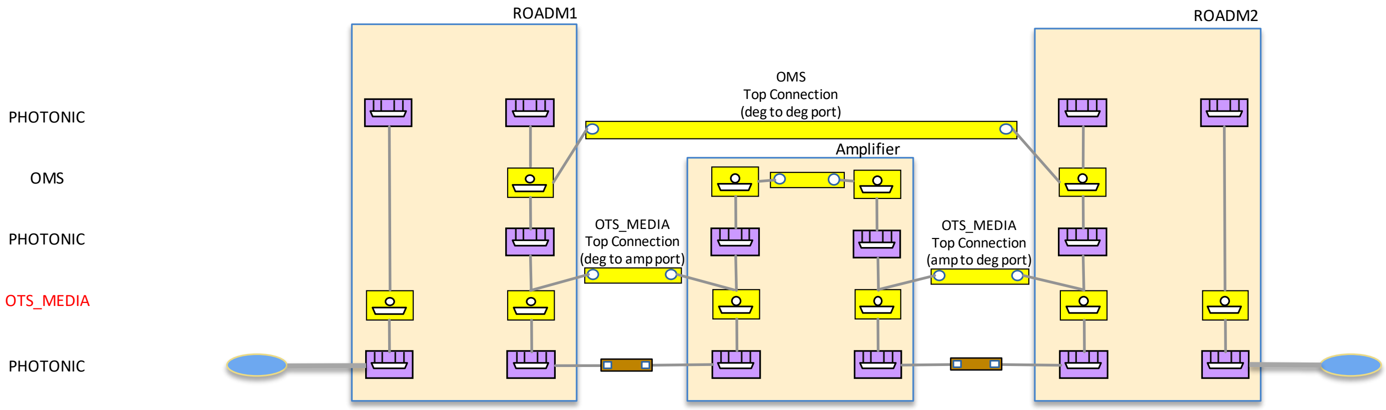


Figure 5-7 Scenario 1 : Optical Line System Controller, time zero, In Line Amplifier

Figure 5-8 shows the result of the provisioning of a MC connectivity service between add/drop ports. The MC top level connection starts and ends at the ROADM1 and ROADM3 add/drop ports.

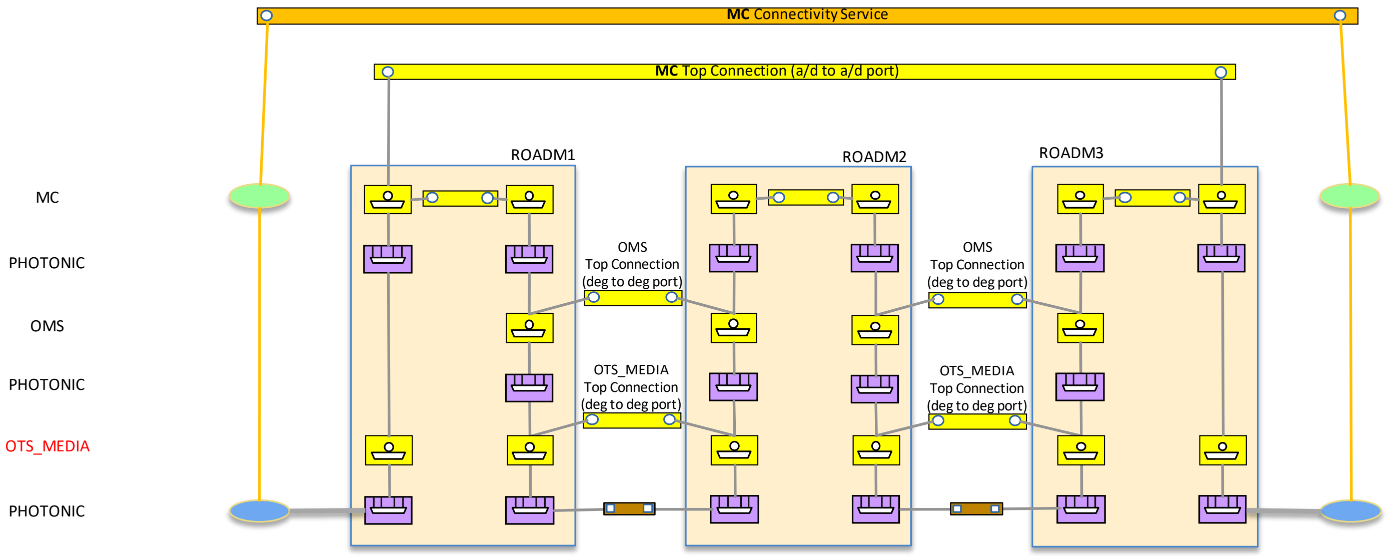


Figure 5-8 Scenario 1 : Optical Line System Controller, MC CS

Figure 5-9 shows the result of a provisioning of an OTSiMC connectivity service between add/drop ports. Note that the same SIPs are addressed for both MC and OTSiMC connectivity service provisioning. Note the greyed OTSiMC CEPs, which may or may not be available depending on e.g. monitoring capabilities.

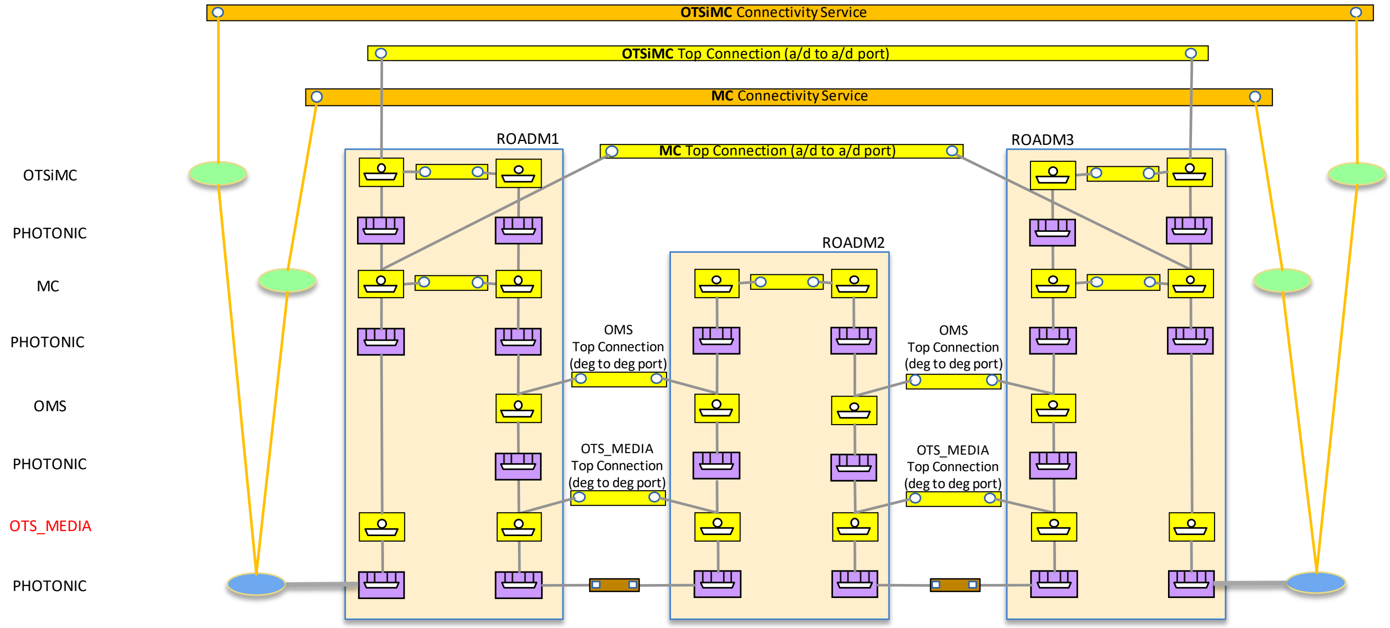


Figure 5-9 Scenario 1 : Optical Line System Controller, MC and OTSiMC CSs

Note that subsequent OTSiMC services may be established reusing or not the existing MC connections depending on their respective allocated spectrum ranges.

Figure 5-10 illustrates a possible layering for an OLS controller at time zero. In this case the SIPs for MC "express media channel" provisioning are available at the degree ports of ROADM1 and ROADM3.

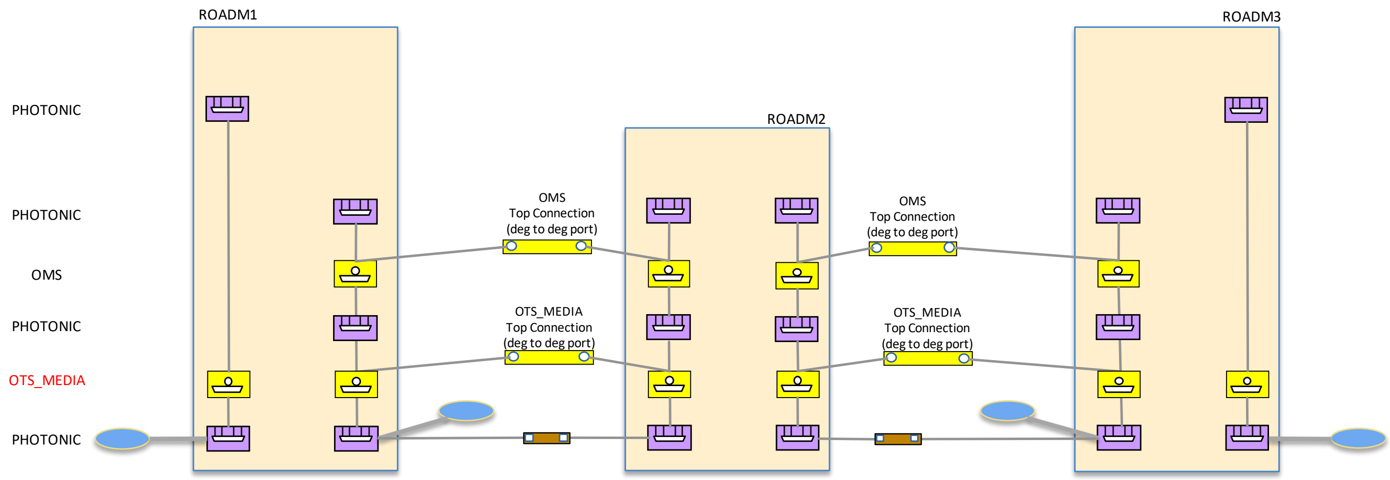


Figure 5-10 Scenario 1 : Optical Line System Controller, time zero, SIPs also on degree ports

Figure 5-11 shows the result of a provisioning of an MC connectivity service between degree ports and Figure 5-12 the subsequent provisioning of an OTSiMC connectivity service between add/drop ports. Note that in case of possible regeneration, the OTSiMC connectivity service shall be replaced by an unterminated OTSiMC+ODU connectivity service, to allow the provisioning of digital OTN parameters, see Figure 5-16. Note that this RIA only considers regeneration functions implemented as OTN.

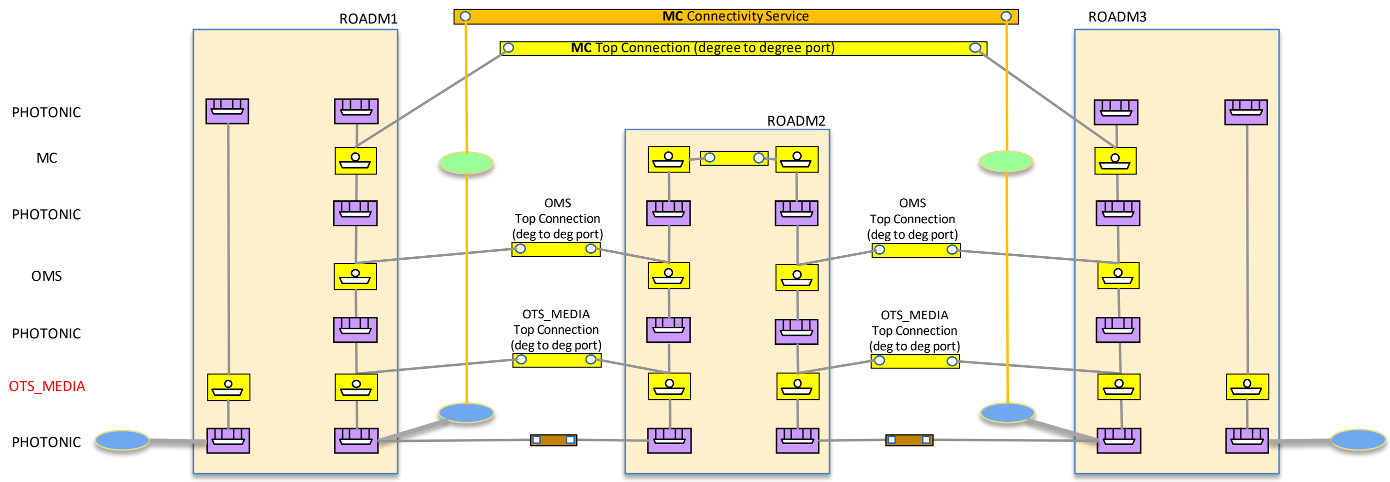


Figure 5-11 Scenario 1 : Optical Line System Controller, MC CS

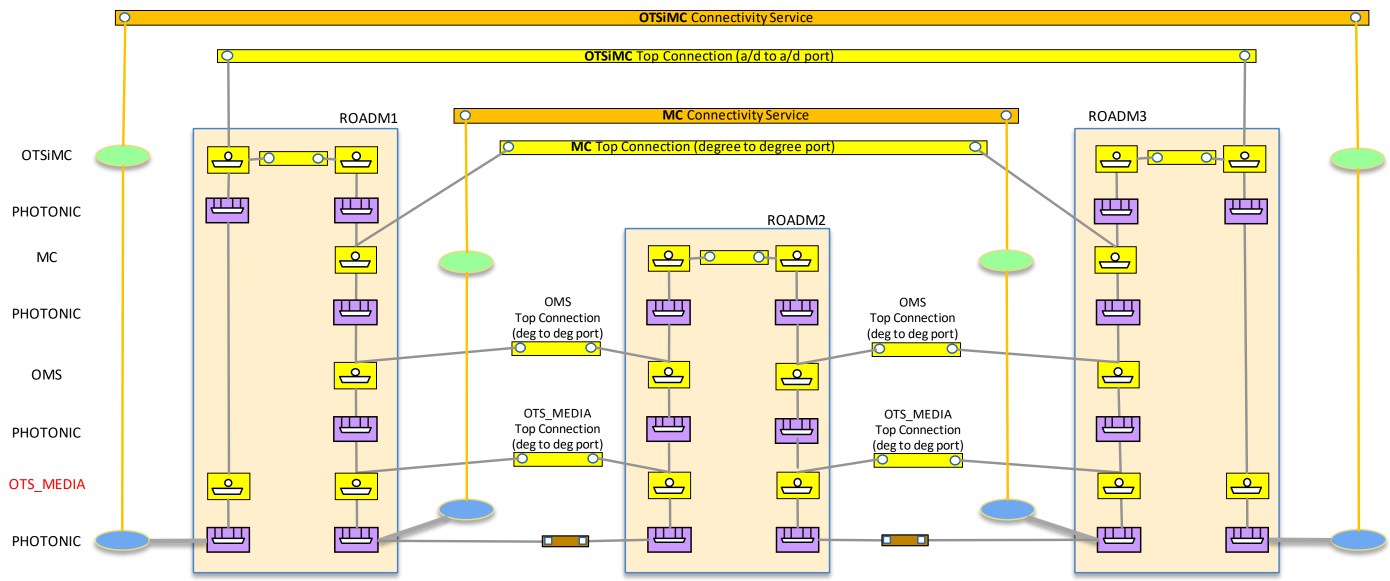


Figure 5-12 Scenario 1 : Optical Line System Controller, OTSiMC and MC CSs

Figure 5-13 shows a hybrid scenario with (MC) SIP at ROADM1 degree port and ROADM3 add/drop port.

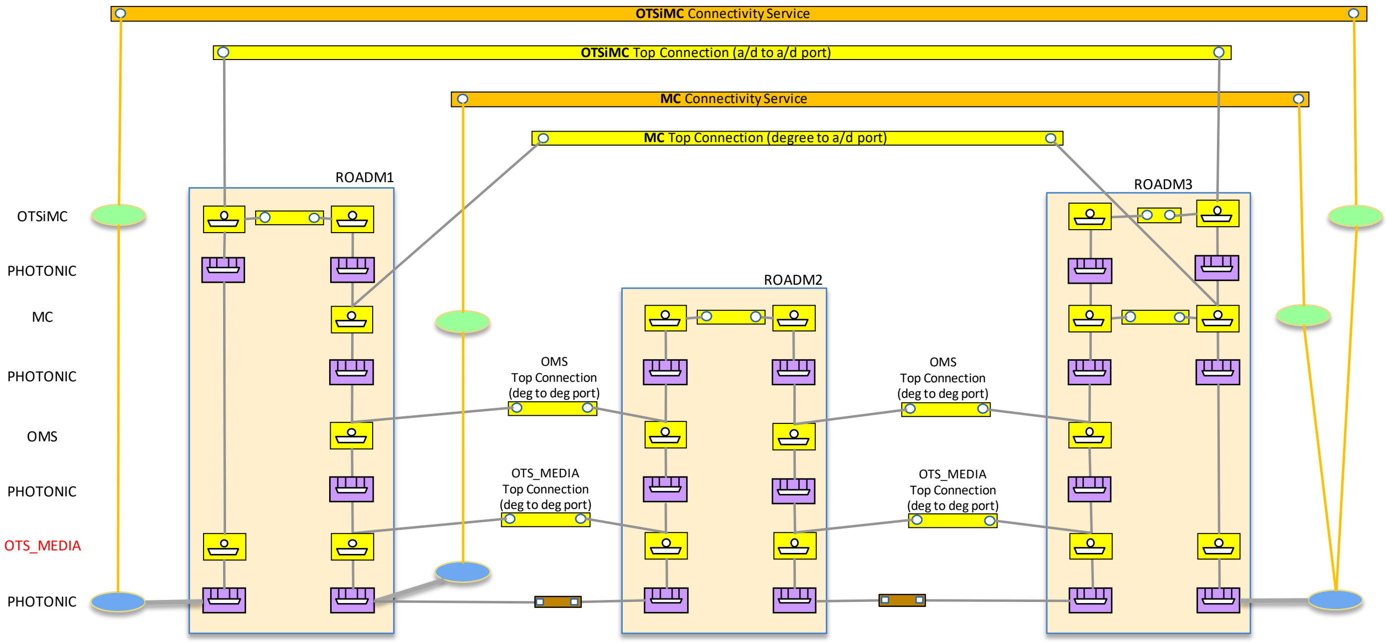


Figure 5-13 Scenario 1 : Optical Line System Controller, SIPs at both degree and a/d ports

Figure 5-14 shows a scenario with multiple optical bands. This RIA does not mandate any specific behavior related to optical band representation and/or OMS instances, allowing maximum flexibility. In particular, implementations MAY reflect bands having multiple OMS instances (one per optical band) or a single OMS instance with the management of MC pools within the same instance.

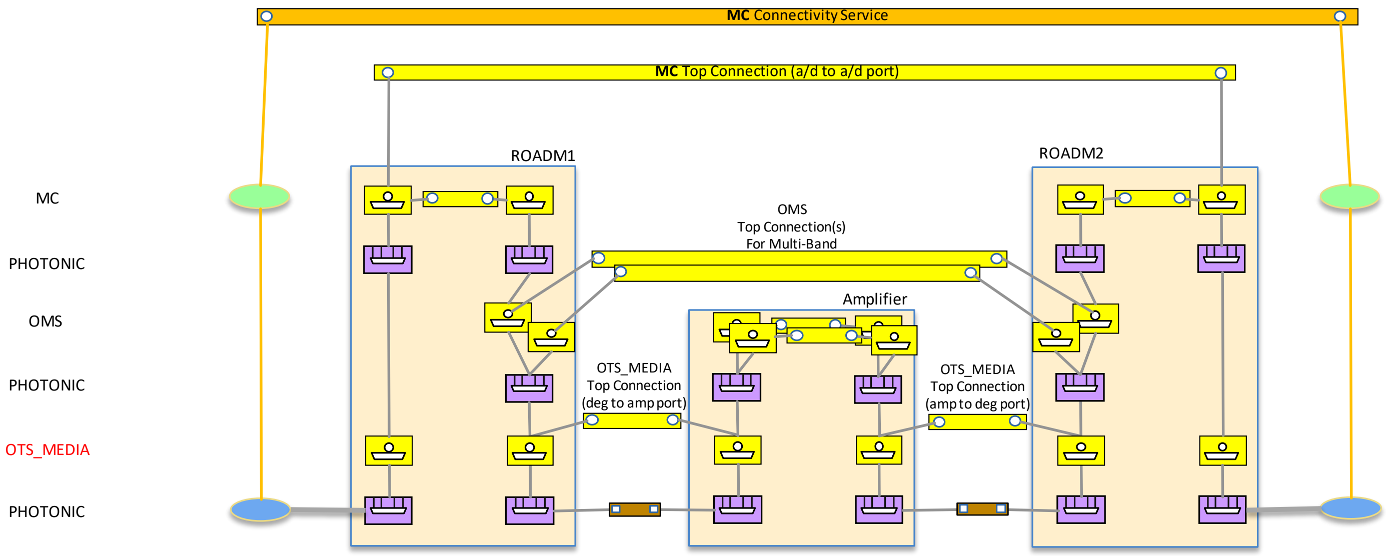


Figure 5-14 Scenario 1 : Optical Line System Controller, multi-band (note: not all MC NEPs are represented)

Figure 5-15 shows a scenario with multiple optical bands, in case the (MC) SIPs are available at ROADM degree ports.

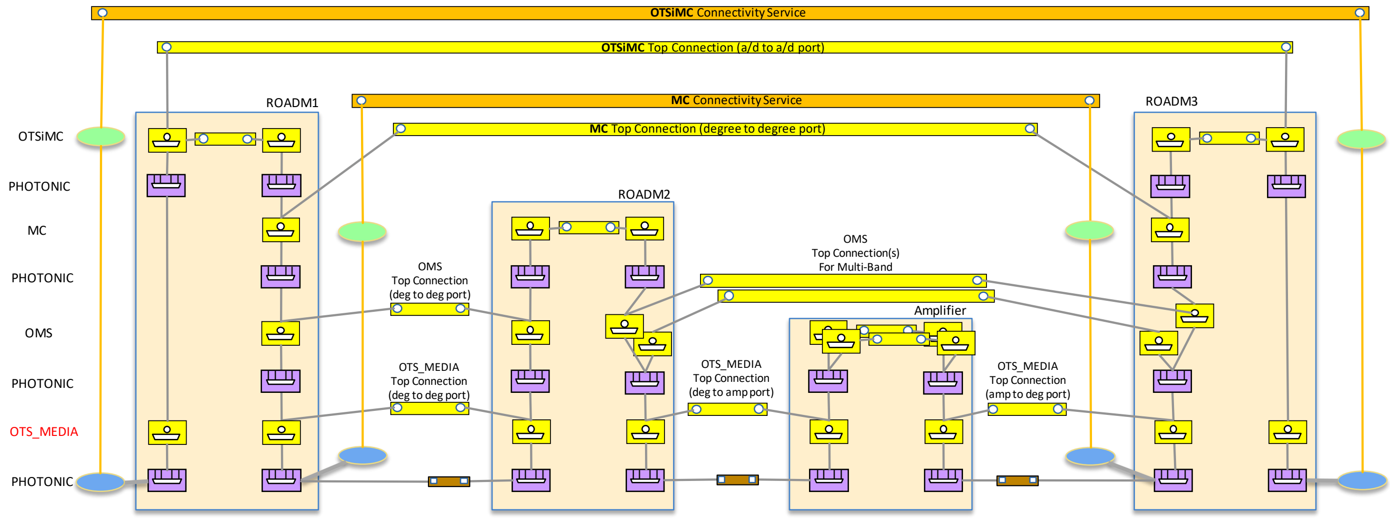


Figure 5-15 Scenario 1 : Optical Line System Controller, multi-band, and SIPs at degree ports

Figure 5-16 shows the provisioning of an unterminated OTSiMC+ODU connectivity service in case of regeneration, which leads to the creation of multiple OTSiMC top-connections between the ROADM add/drop ports and the regenerator ports, plus an ODU top-connection between the unterminated OTSiMC CEPs of the ROADM add/drop ports. SIPs are not shown. This is an example of Transit Scenario.

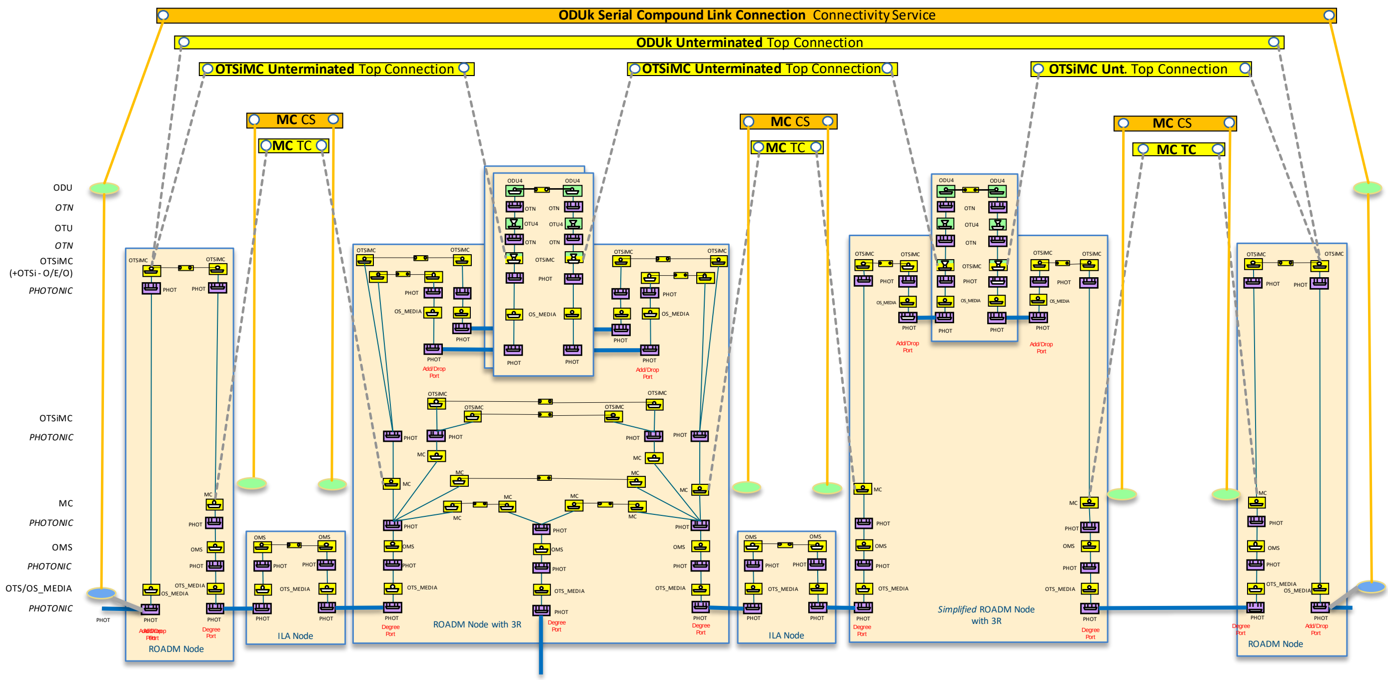


Figure 5-16 Scenario 1 : Optical Line System Controller, regeneration

5.2.2 Scenario 2 : Integrated Management

Figure 5-17 illustrates a possible layering for an integrated management scenario at time zero.

There are OTS_MEDIA top-connections between the transceiver line port and the ROADM add/drop ports as well as between ROADM degree ports. There is an OMS top-connection between ROADM degree ports as well.

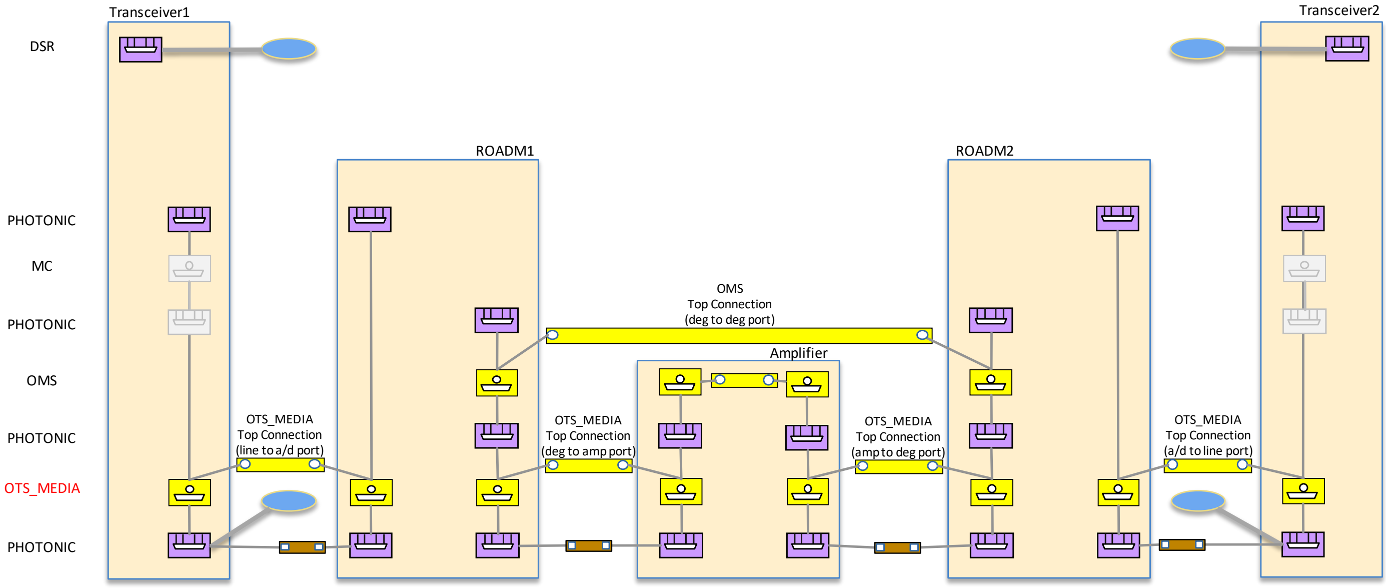


Figure 5-17 Scenario 2 : Integrated Management, time zero

Figure 5-18 illustrates a possible distinction between OTS and simpler OS layer protocol qualifiers.

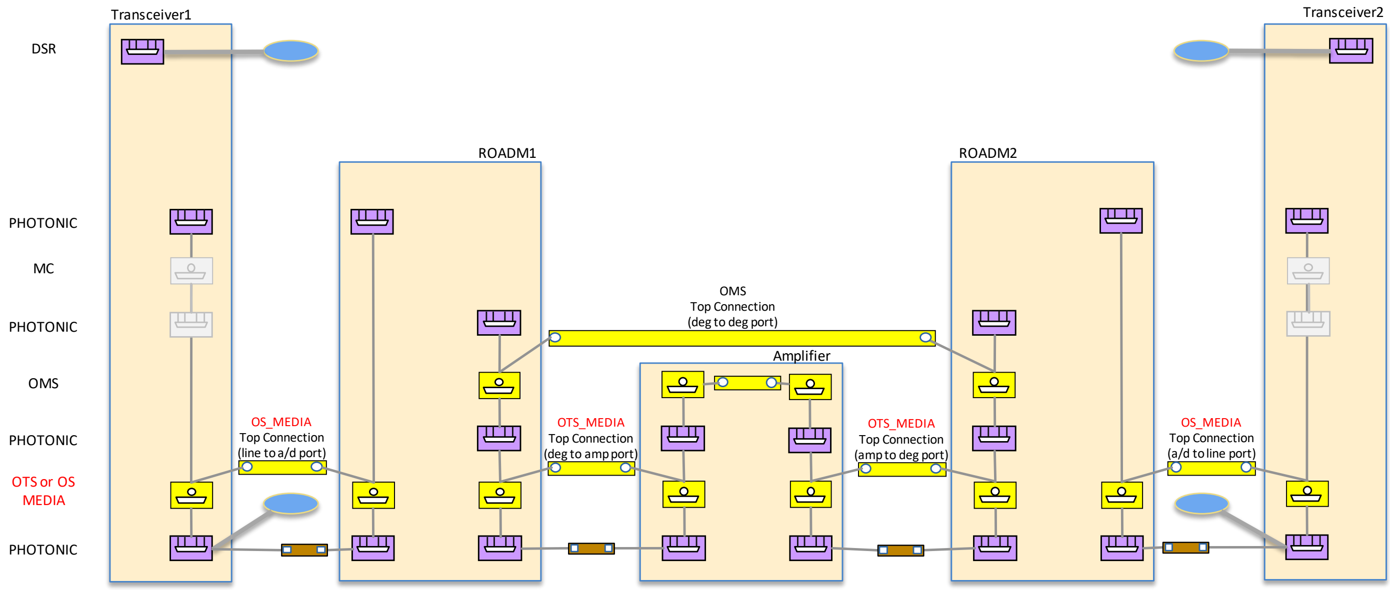


Figure 5-18 Scenario 2 : Integrated Management, time zero, OS_MEDIA

Figure 5-19 illustrates a similar scenario, with in addition the SIPs at ROADM add/drop ports.

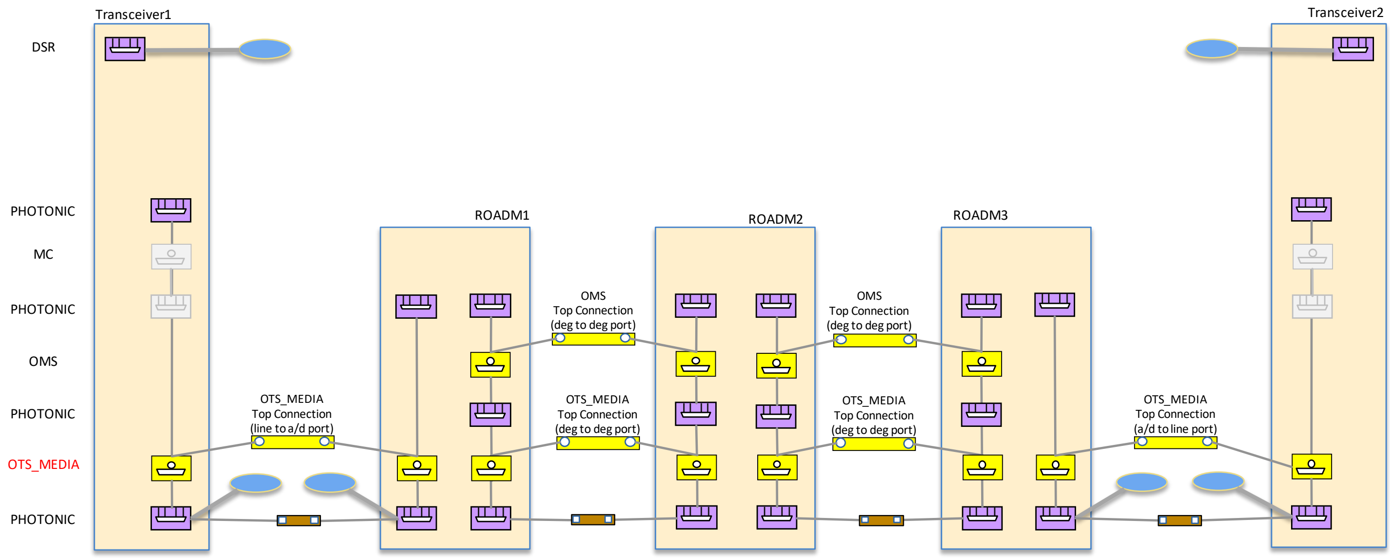


Figure 5-19 Scenario 2 : Integrated Management, time zero, SIPs at a/d ports

Figure 5-20 shows the MC connectivity service and its MC top-connection which start and end at the ROADM add/drop ports.

Note that the MC connection MAY be projected (extended) to the transceivers line ports to highlight that the transponder may have more than one OTSi instance and/or the band that is available to the transponder may be restricted by configuration in the attached ROADM. In other words, the presence of the MC sub-layer in the transceiver line port is

OPTIONAL and not recommended in the case there is a single OTSiMC. In this case, the MC top-connection will start and end in the transceivers line ports.

Figure 5-20 Scenario 2 : Integrated Management, MC CS

Figure 5-21 adds the OTSiMC+ODU connectivity service, which leads to the creation of an OTSiMC top-connection between the transceivers line ports plus an ODU top-connection between the unterminated ODU CEPs.

There may be no SIPs on ROADMs (and associated connectivity service) in a case where the controller has the capability of creating MC connections driven by OTSiMC+ODU service creation and some associated MC creation policy.

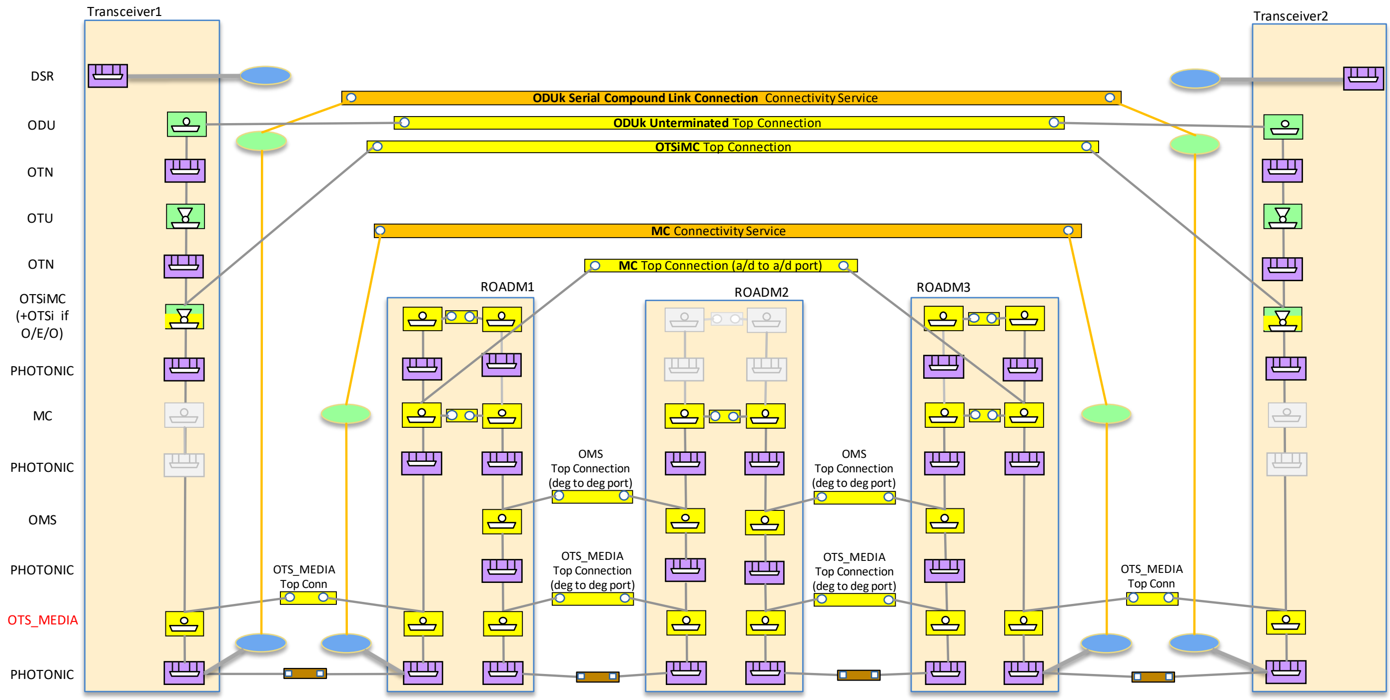


Figure 5-21 Scenario 2 : Integrated Management, MC and OTSiMC+ODU CSs

Figure 5-22 adds the DSR connectivity service, which leads to the creation of an ODU top-connection between the terminated ODU CEPs plus a DSR top-connection between the transceiver client ports.

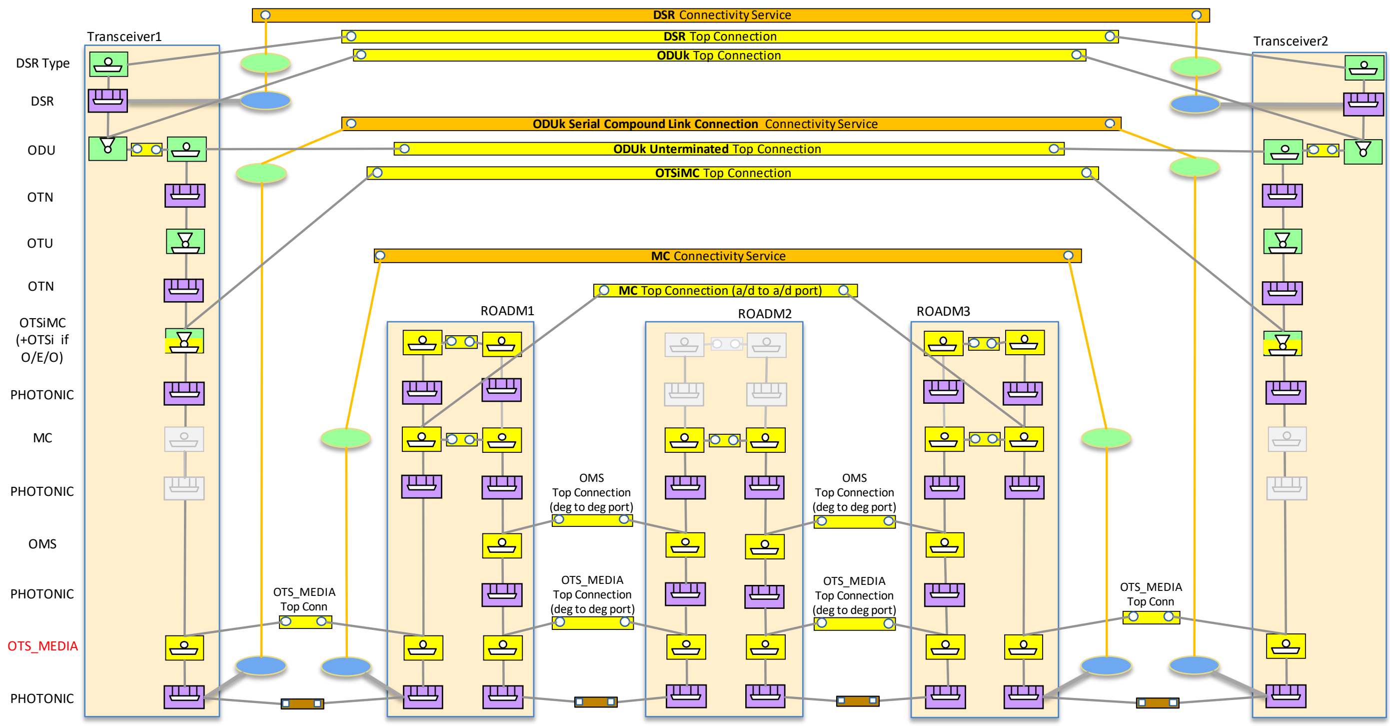


Figure 5-22 Scenario 2 : Integrated Management, MC and OTSiMC+ODU and DSR CSs

Figure 5-23 extends the previous scenario to show the OTSiMC cross-connections (e.g., to support monitoring points for individual OTSiMC) in gray also in ROADM 1 and 3. The switching happens at the MC level (switching of OTSiMC is congruent). Note that the effective frequency slot width of the MC connection may be greater than the OTSiMC frequency slots it supports.

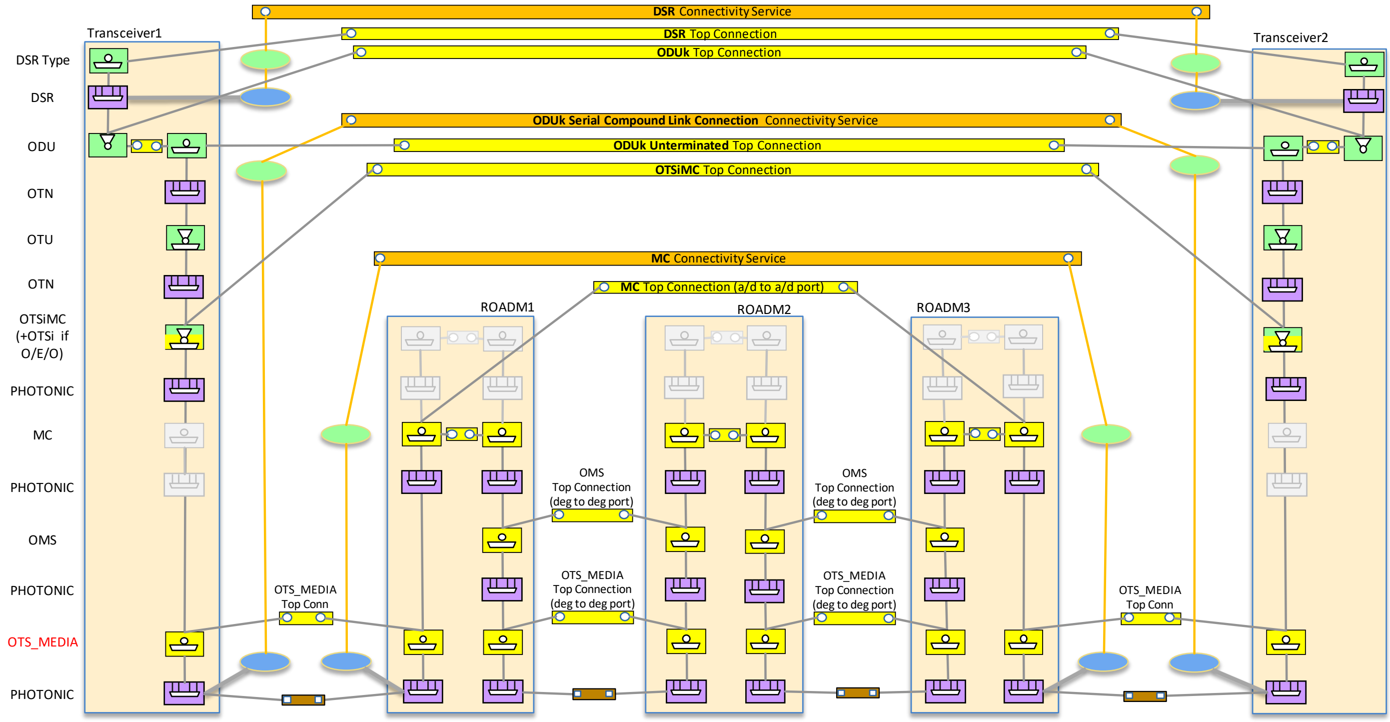


Figure 5-23 Scenario 2 : Integrated Management, MC and OTSiMC+ODU and DSR CSs, OTSiMC CEPs

Figure 5-24 illustrates a possible alternative scenario with respect to Figure 5-23, where the unterminated OTSiMC CS is created in the OLS and then the provisioning of OTSiMC+ODU connectivity service leads to the creation of the terminated OTSiMC top-connection between the transceivers. See also [TAPI-CONN-MODEL-REQ-3].

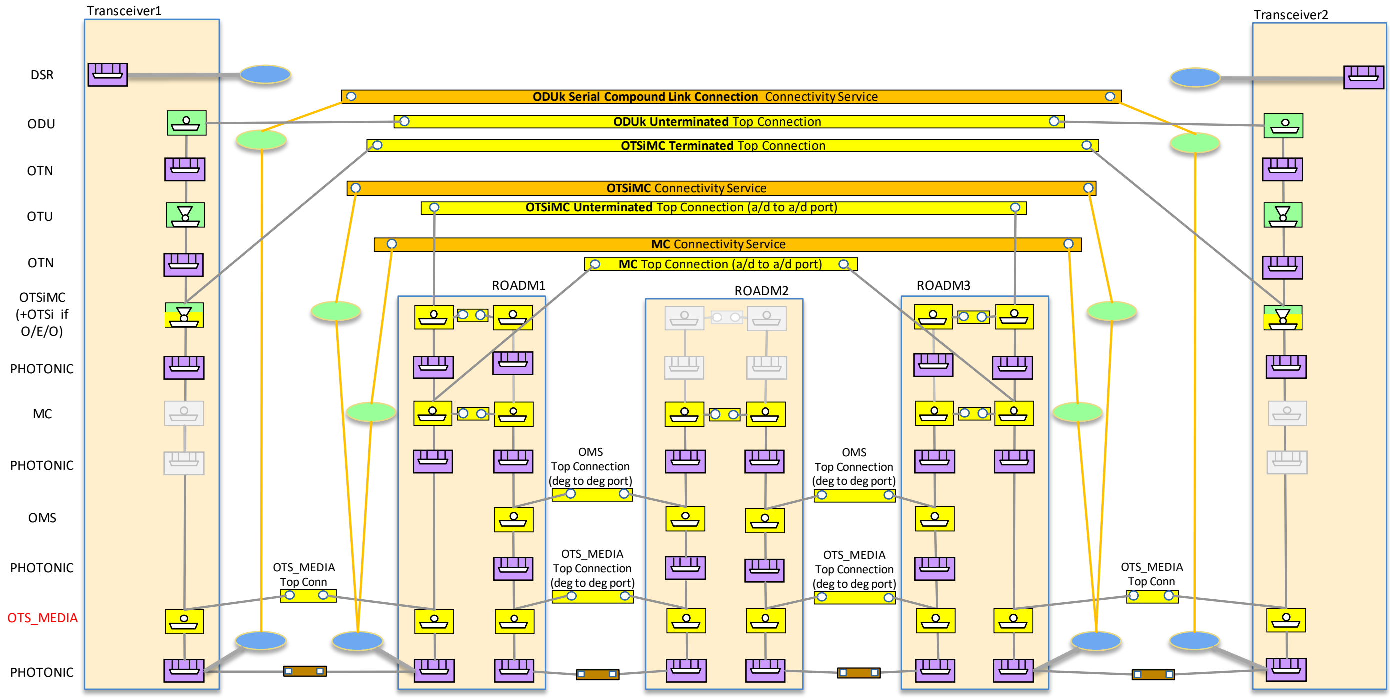


Figure 5-24 Scenario 2 : Integrated Mng, MC CS, unterminated OTSiMC CS, terminated OTSiMC +ODU CSs

illustrates a possible alternative scenario with respect to , where an unterminated OTSiMC+ODU connectivity service is provisioned to manage regeneration functions in the route along the OLS.

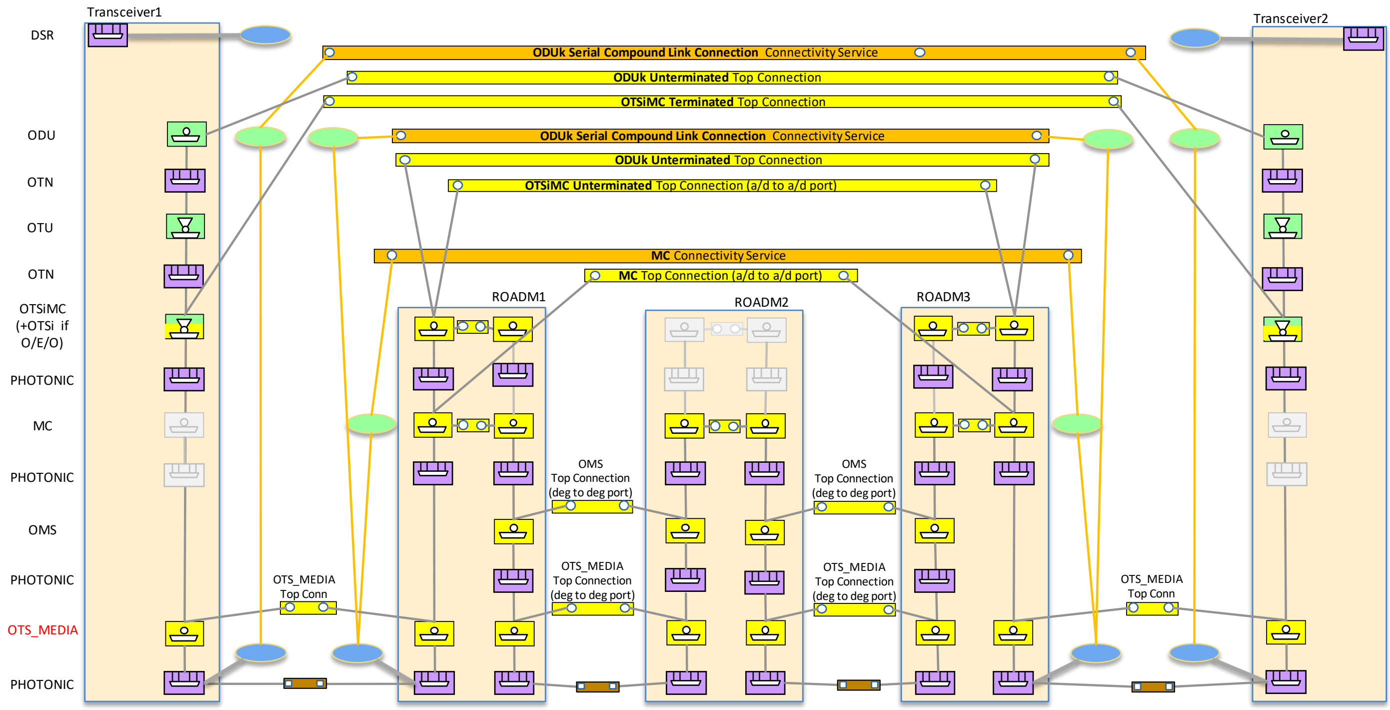


Figure 5-25 Scenario 2 : Integrated Mng, MC CS, unterminated OTSiMC +ODU CS, terminated OTSiMC +ODU CSs

Figure 5-26 illustrates a possible layering for an integrated management scenario at time zero, with the (MC) SIPs at ROADM1 and ROADM3 degree ports.

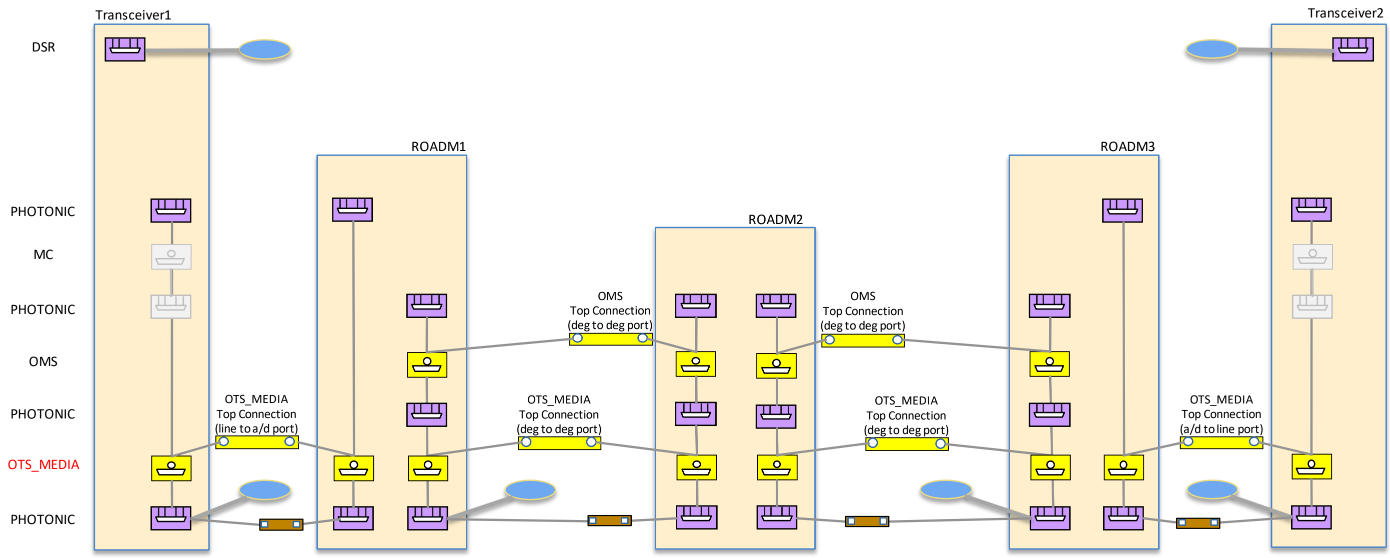


Figure 5-26 Scenario 2 : Integrated Management, time zero, SIPs at ROADM degree ports

Figure 5-27 shows an "express media channel" between the edge ROADMs with a given effective frequency slot width. The MC express media channel starts and ends at the ROADM degree ports and the intermediate ROADMs switch the MC channel (coarse granularity).

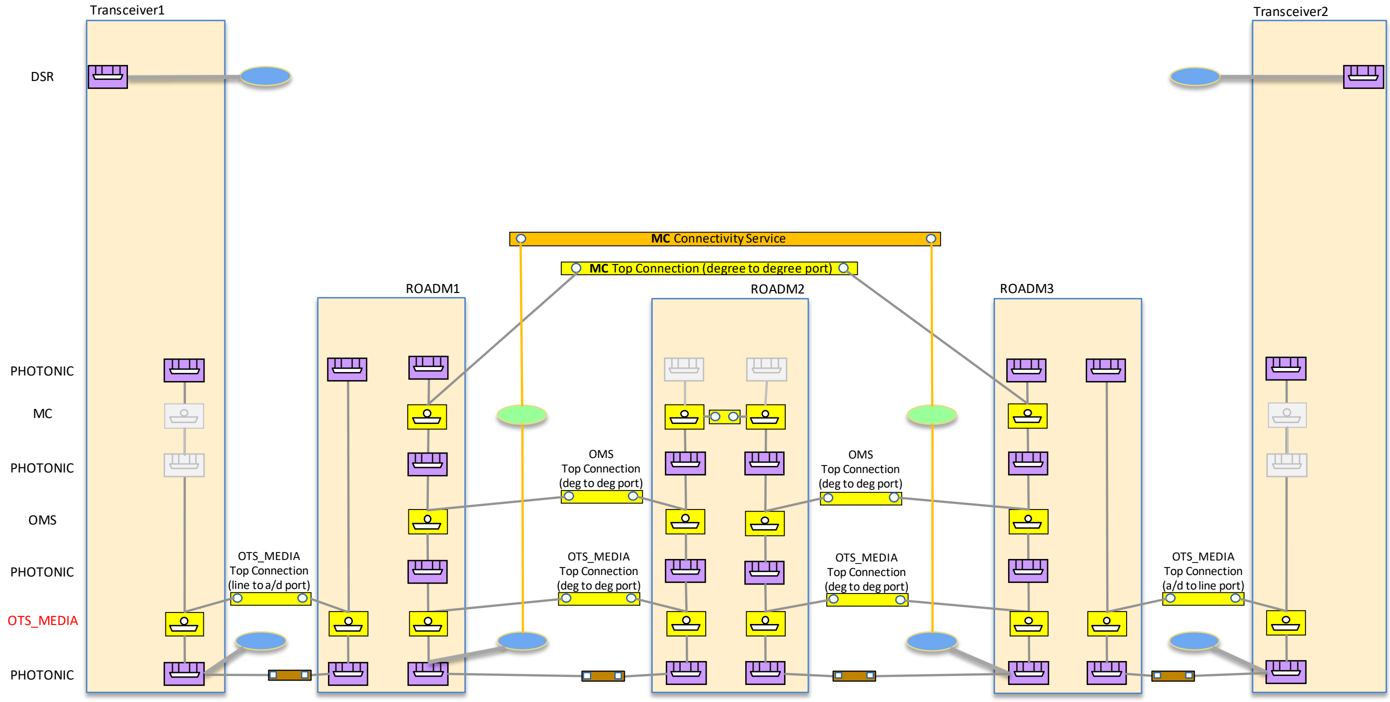


Figure 5-27 Scenario 2 : Integrated Management, MC CS

Figure 5-28 adds the OTSiMC+ODU connectivity service, which leads to the creation of an OTSiMC top-connection between the transceivers line ports plus an ODU top-connection between the unterminated ODU CEPs.

Note that multiple OTSiMC connections may share the same MC. Individual OTSiMC connections may be explicit and monitored at intermediate nodes (gray NEP/CEPs). Note that switching happens at the MC layer, OTSiMC switching represents individual OTSiMC forwarding but it is congruent with the MC.

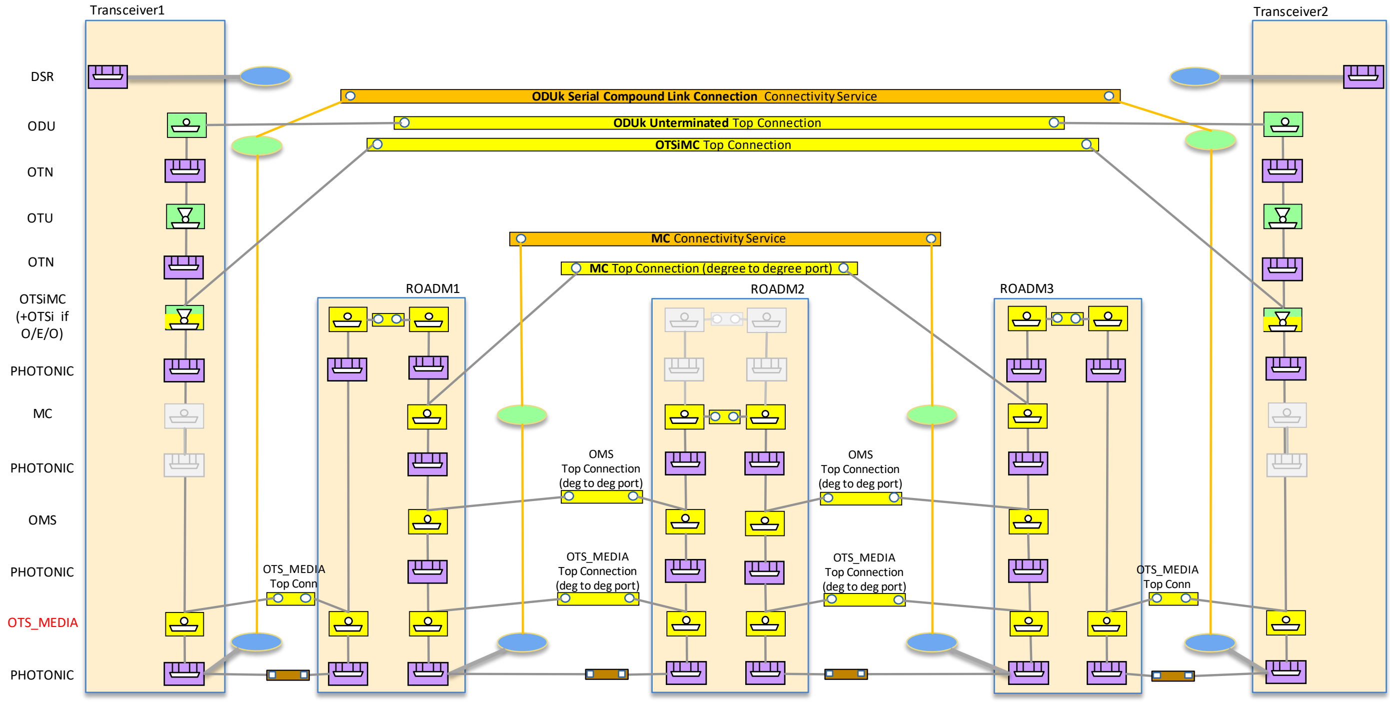


Figure 5-28 Scenario 2 : Integrated Management, MC and OTSiMC+ODU CSs

Figure 5-29 adds the DSR connectivity service, which leads to the creation of an ODU top-connection between the terminated ODU CEPs plus a DSR top-connection between the transceiver client ports.

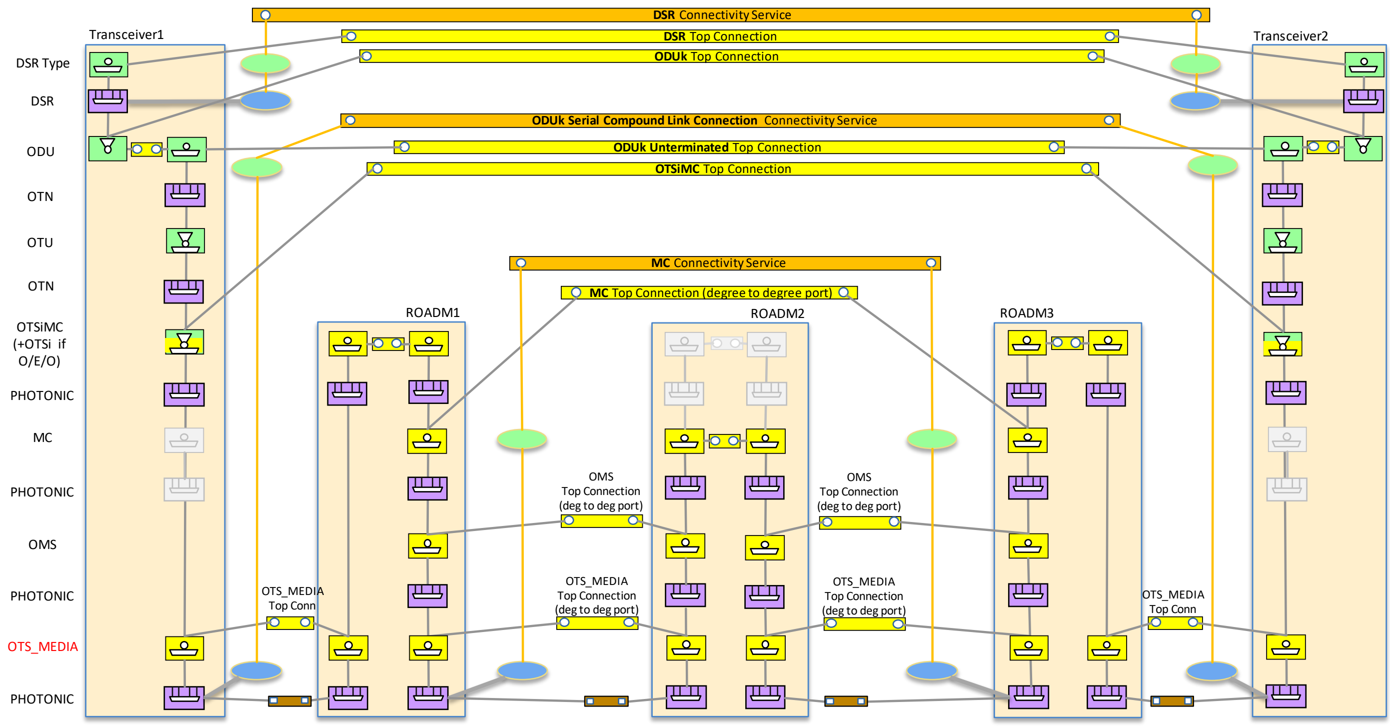


Figure 5-29 Scenario 2 : Integrated Management, MC and OTSiMC+ODU and DSR CSs, OTSiMC CEPs

Figure 5-30 is a variation of the previous scenarios where it is shown multiple add/drop port tributary signals being forwarded to a common express media channel.

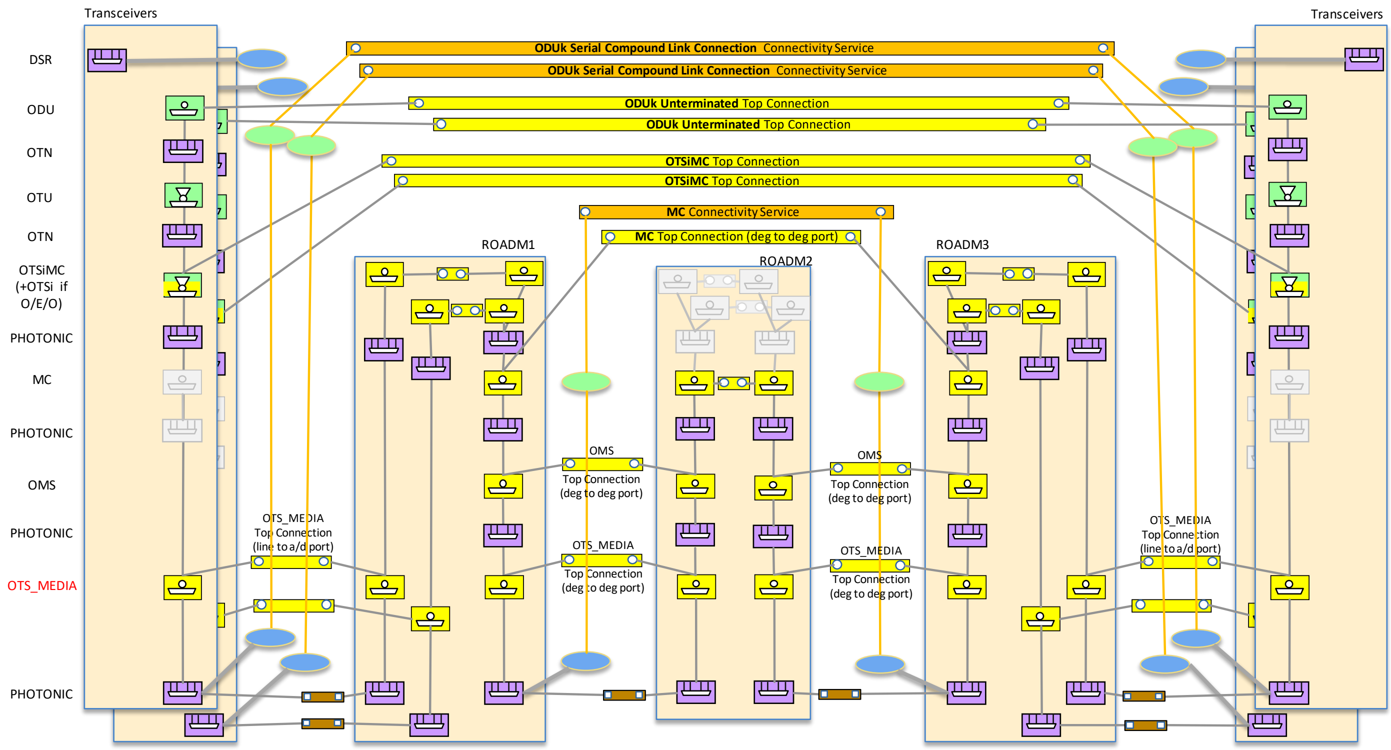


Figure 5-30 Scenario 2 : Integrated Management, MC and OTSiMC+ODU CSs, more OTSiMCs on MC

Figure 5-31 shows another variation with more optical carriers (OTSi) multiplexed on the transceiver line ports.

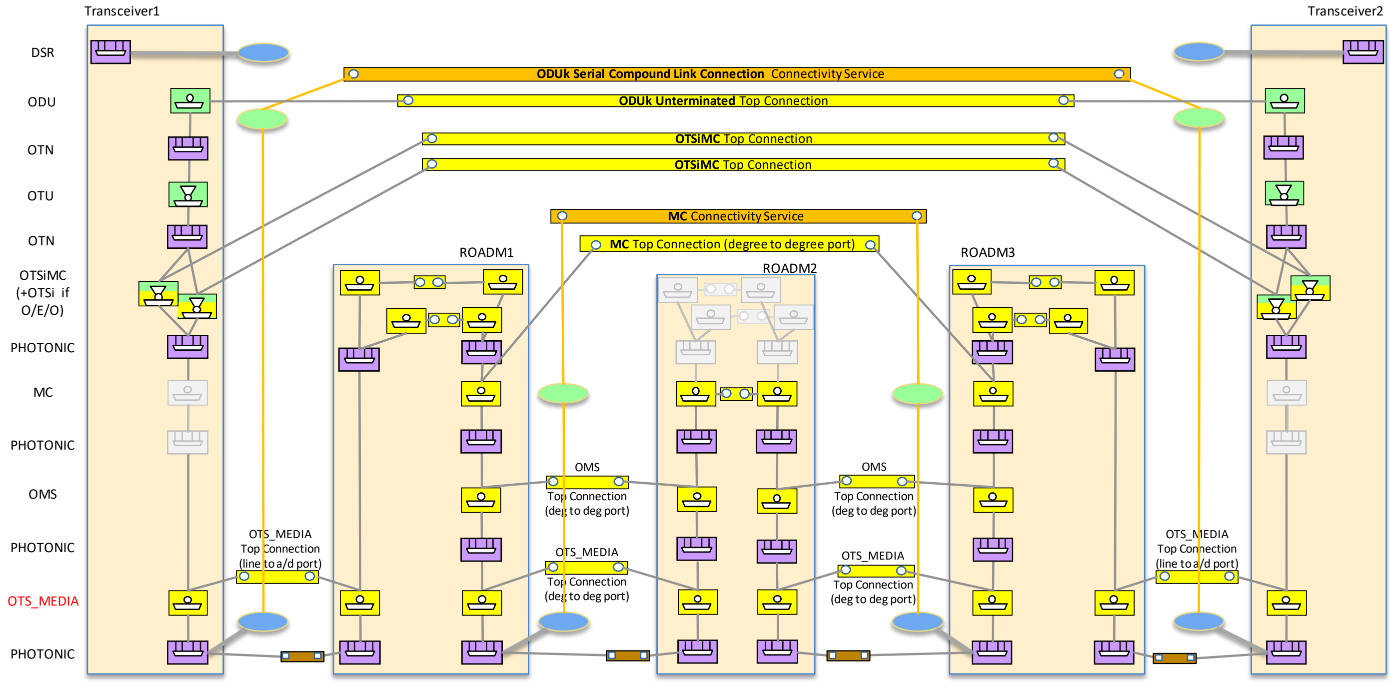


Figure 5-31 Scenario 2 : Integrated Management, MC and OTSiMC+ODU CSs, more OTSiMCs on MC, single line port

Figure 5-32 shows a simplification where the MC channels are not explicitly represented. Only the OTSiMC protocol qualifier switching is present at the ROADM nodes, thus switching individual OTSiMC.

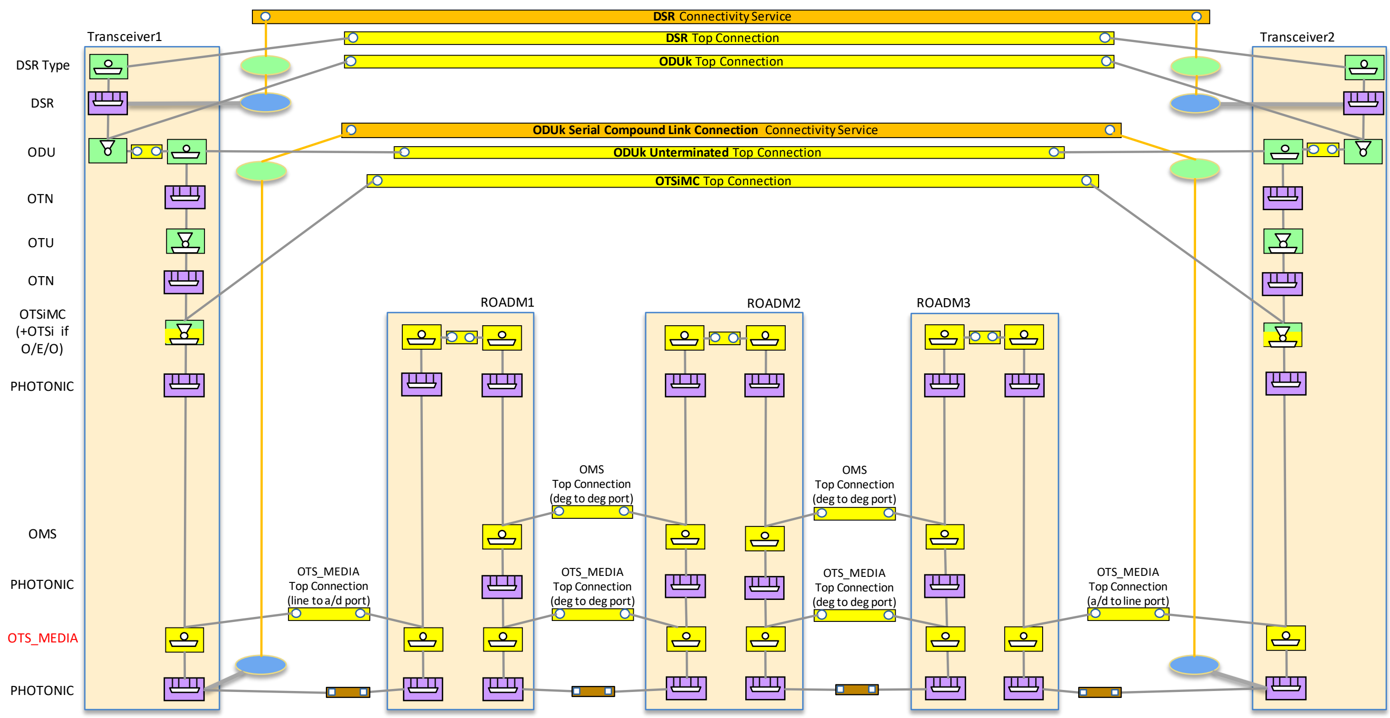


Figure 5-32 Scenario 2 : Integrated Management, MC and OTSiMC+ODU CSs, MC not represented

Figure 5-33 shows a sequence of MC Connections, where at ROADM2 the flexibility is at OTSiMC granularity. In other words, the MC connections are shorter than the span between edge ROADM1 and ROADM4.

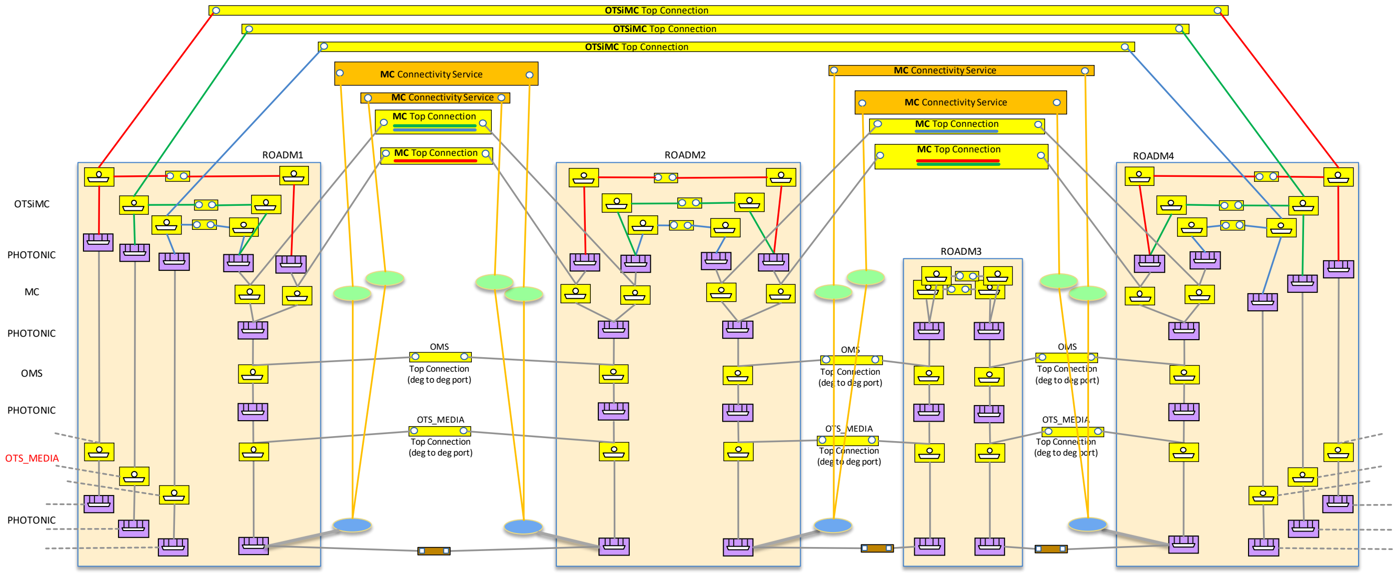


Figure 5-33 Scenario 2 : Integrated Management, sequence of MC top-connections

Figure 5-34 shows the provisioning of OTSiMC+ODU connectivity service, which leads to the creation of multiple OTSiMC top-connections between the transceivers line ports and the regenerator ports, plus an ODU top-connection between the unterminated ODU CEPs. SIPs are not shown.

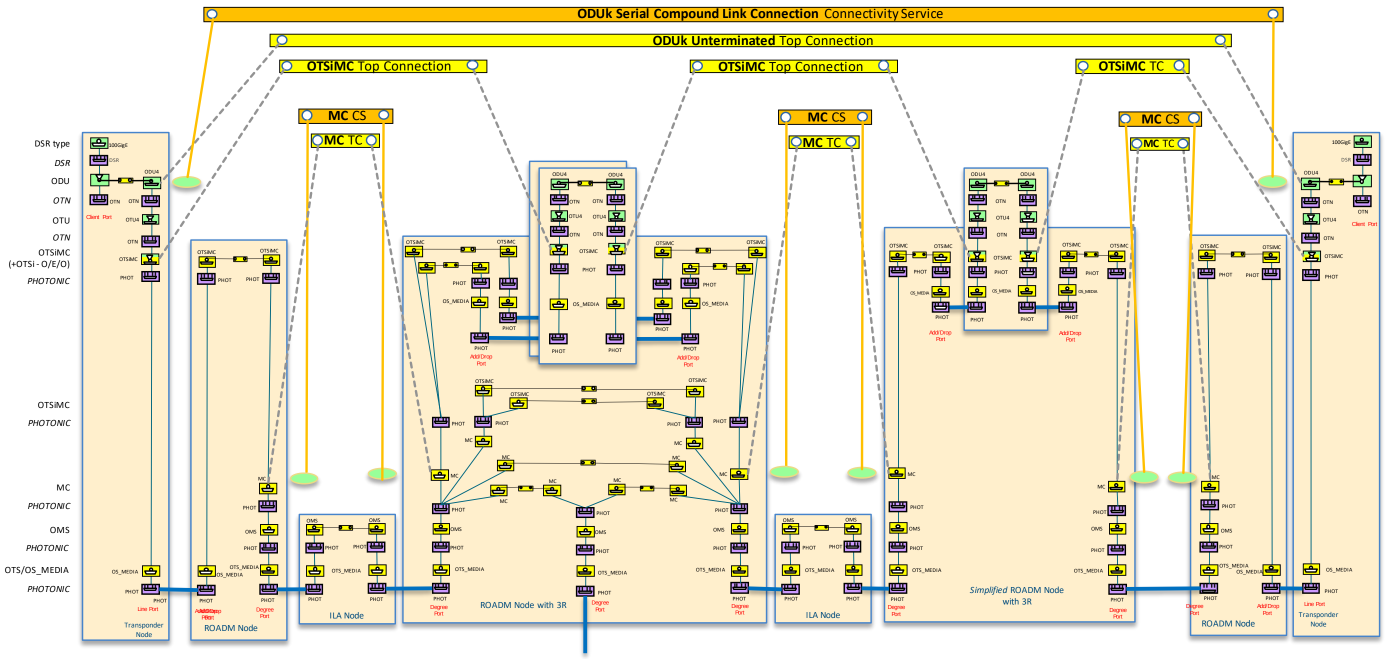


Figure 5-34 Scenario 2 : Integrated Management, regeneration

5.2.3 DSR UNI and OTN ENNI considerations

This RIA considers DSR based UNI and OTN based ENNI interfaces. ENNI interfaces are especially relevant in asymmetric scenarios.

5.2.3.1 UNI (DSR)

To model DSR UNI, several options are available based on the level of detail that is presented to TAPI clients as well as the coalescing on functions into CEPs. This section presents different modelling options along with considerations for implementations to select the most suitable ones. Each option differs in the assumptions in terms of flexibility at the DSR and/or DIGITAL_OTN layers.

5.2.3.1.1 Option: Explicit DSR cross-connection

This option (Figure 5-35) does not include lower layers at the UNI below the DSR NEP. The explicit DSR/10GE cross-connection is used to reflect the decapsulation of the DSR signal from the UNI NEP and its encapsulation into the ODU function. It can appear even if the association is fixed but it may also reflect existing flexibility in DSR switching between the UNI NEP and intermediate or NNI NEPs. This option also reflects switching flexibility at the ODU2 and ODU4 switching levels and the existence of ODU2 and ODU4 cross-connections.

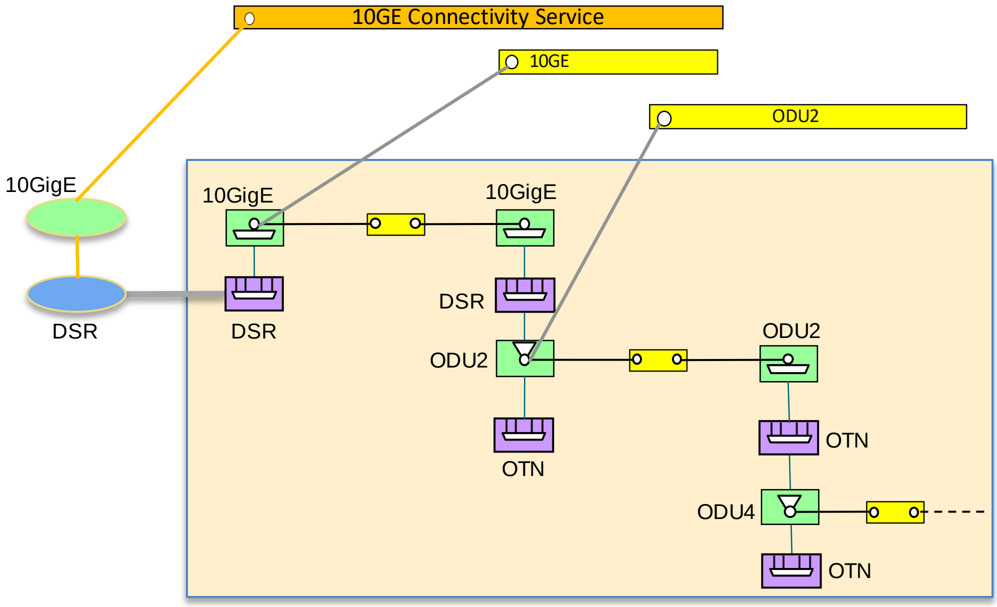


Figure 5-35 Option: Explicit DSR cross-connection

5.2.3.1.2 Option: Explicit DSR cross-connection, no ODU-LO cross-connection

This option (Figure 5-36) is analog to the previous one but does not include ODU-LO cross-connection. It usually means that the ODU2 is used for framing the DSR/Eth signal.

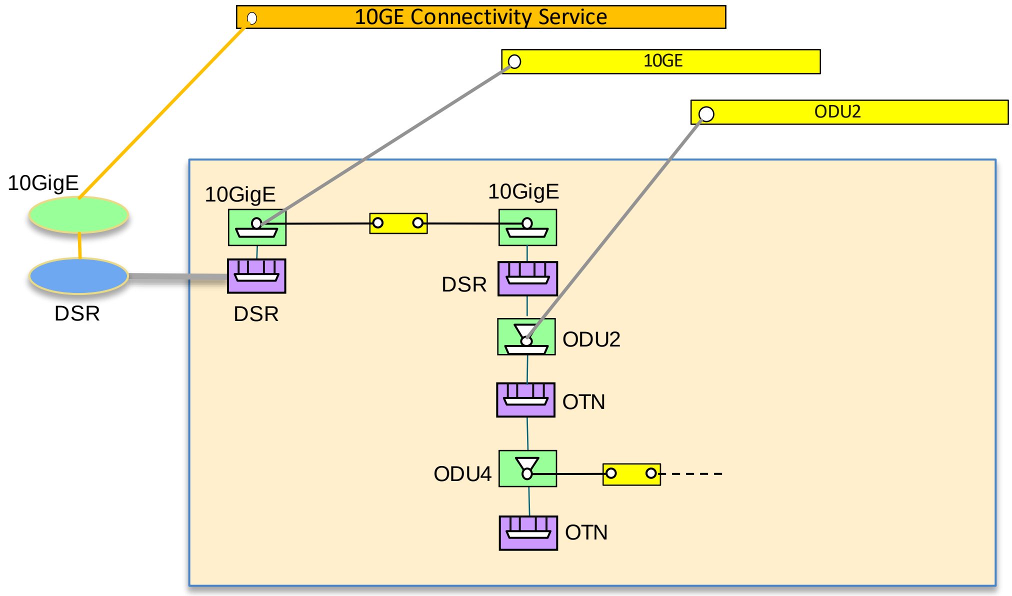


Figure 5-36 Option: Explicit DSR cross-connection, no ODU-LO cross-connection

5.2.3.1.3 Option: No DSR cross-connection

This option (Figure 5-37) does include the ODU2 cross-connection but does not reflect 10GE cross-connections.

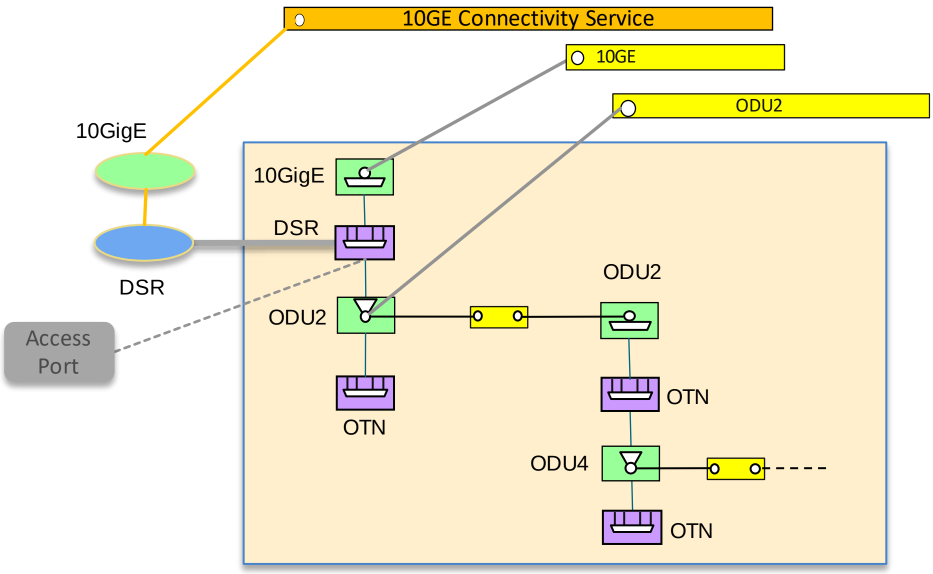


Figure 5-37 Option: No DSR cross-connection, with ODU-LO cross-connection

5.2.3.1.4 Option: No cross-connection

This option (Figure 5-38) does not include cross-connections neither at the DSR nor at the ODU-LO level, showing no flexibility in switching. It should be used only to model simple devices that e.g., frame the client signal and multiplex multiple ODU-LO into an ODU-HO with a single line port or with static mappings of UNI to NNI ports.

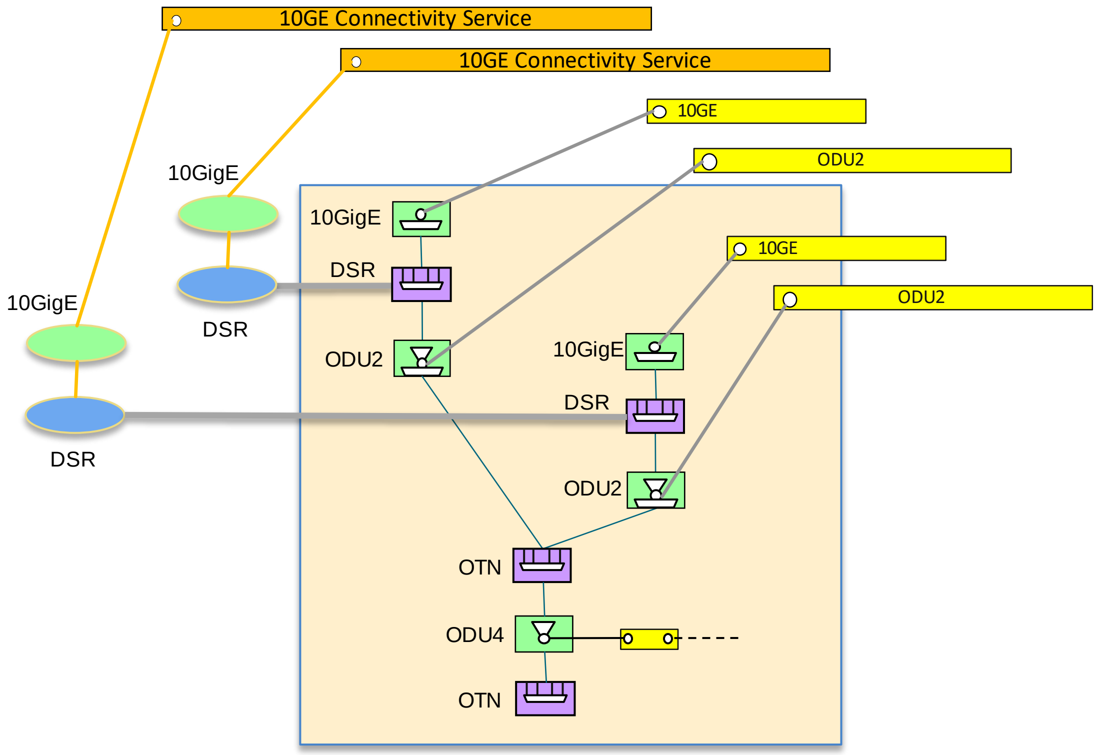


Figure 5-38 Option: No DSR/ODU-LO cross-connections

5.2.3.1.5 Option: Simplified DSR UNI

In view of the systematic use of the same pattern in terms of NEPs and CEPs, this RIA allows the use of a simplified representation, in which a single CEP instance (*coalesced CEP*) models the different involved (embedded) functions. For example, Figure 5-39 shows a single CEP encapsulating the 10GE CTP and the ODU2 TTP functions. Consider:

1) The CEP LPQ is, by convention, the “top-most” LPQ of the involved functions (i.e., 10GE).

2) A connection (both top-connections and cross-connections) has its own LPN/LPQ which may be different of the LPN/LPQ of the connected CEPs. This is the case of the DIGITAL_OTN/ODU2 top-connection which starts in a DSR/10GE CEP.

3) The termination state of the CEP refers to its LPQ so, in this case, the 10GE is not terminated whereas the encapsulated ODU2 function is terminated.

4) The coalesced CEP MUST appear in the 10GE top-connection as well as in the ODU2 top-connection.

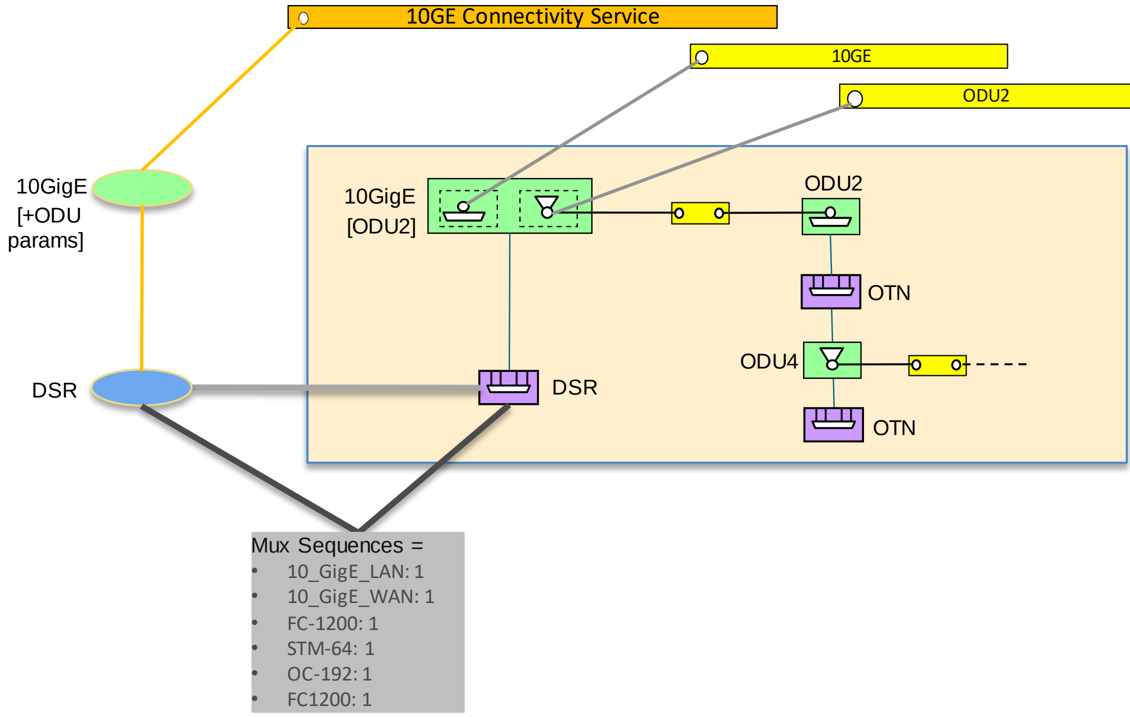


Figure 5-39 Option: Simplified DSR UNI

This simplification may be used to embed additional functions (see Figure 5-40), which otherwise would need an explicit modeling of functions by means of additional NEPs and CEPs.

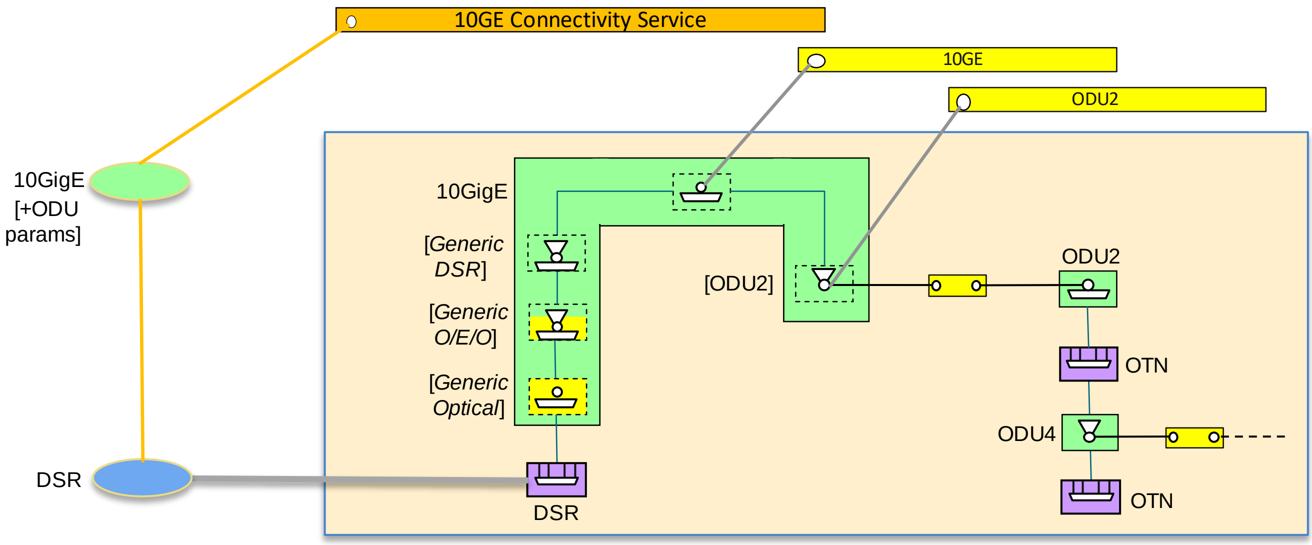
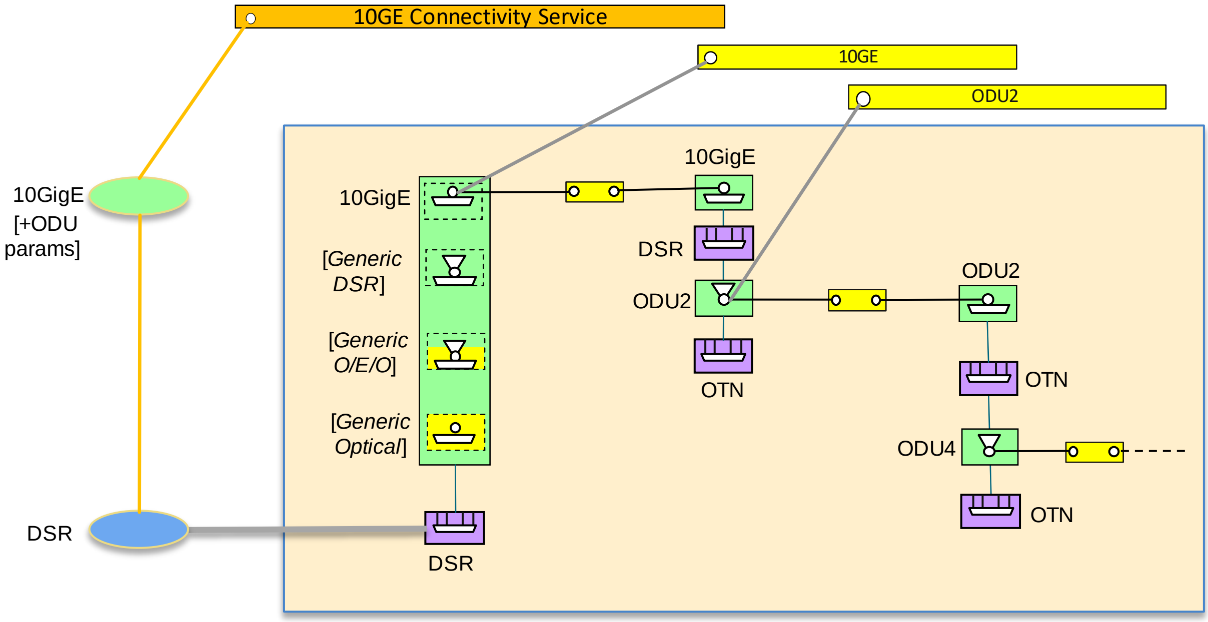
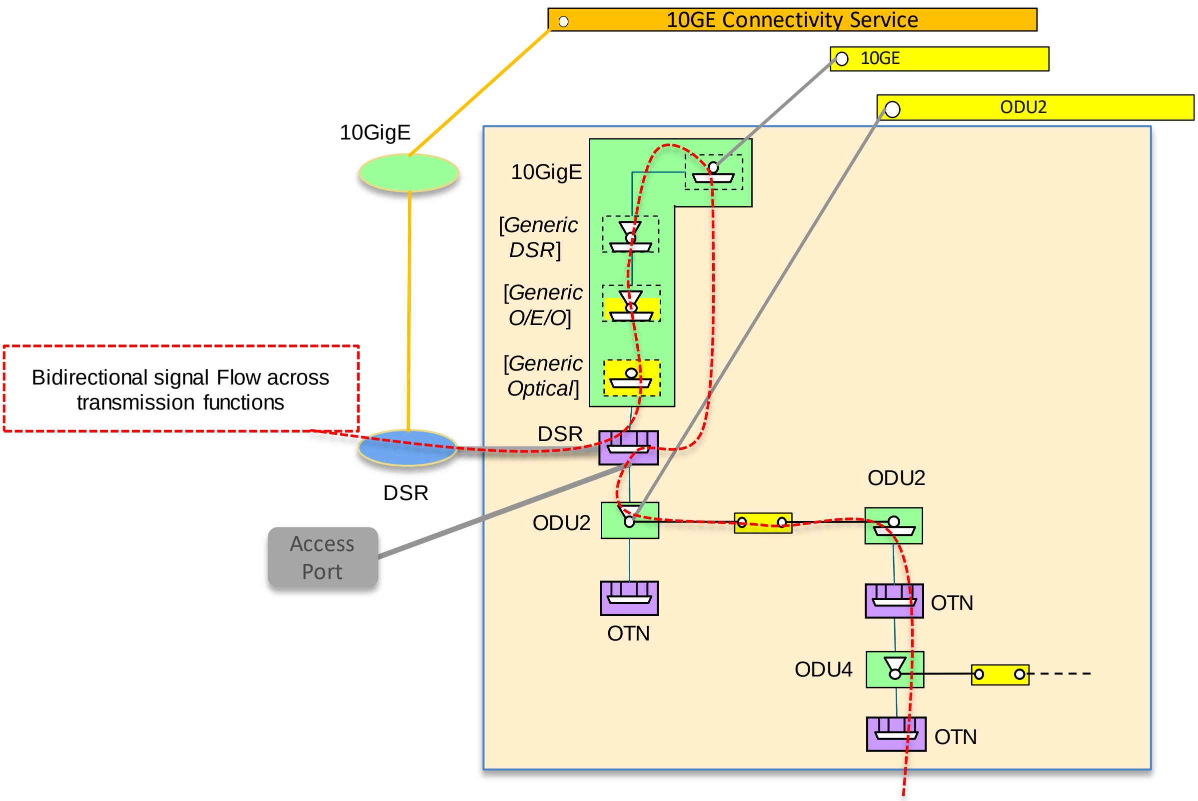


Figure 5-40 Option: Simplified DSR UNI with additional embedded functions

The possibility of embedding functions and attributes of lower layers to the DSR CEP also applies in the cases where the 10GE cross-connection is explicit or not with simplified DSR, as shown in Figure 5-41:


10GE Connectivity Service
10GE
ODU2
10GigE
[+ODU
params]
DSR
10GigE
[Generic
DSR]
[Generic
O/E/O]
[Generic
Optical]
DSR
10GigE
DSR
ODU2
OTN
ODU2
OTN
ODU4
OTN


10GE Connectivity Service
10GE
ODU2
10GigE
Bidirectional signal Flow across
transmission functions
DSR
Access
Port
10GigE
[Generic
DSR]
[Generic
O/E/O]
[Generic
Optical]
DSR
ODU2
OTN
ODU2
OTN
ODU4
OTN

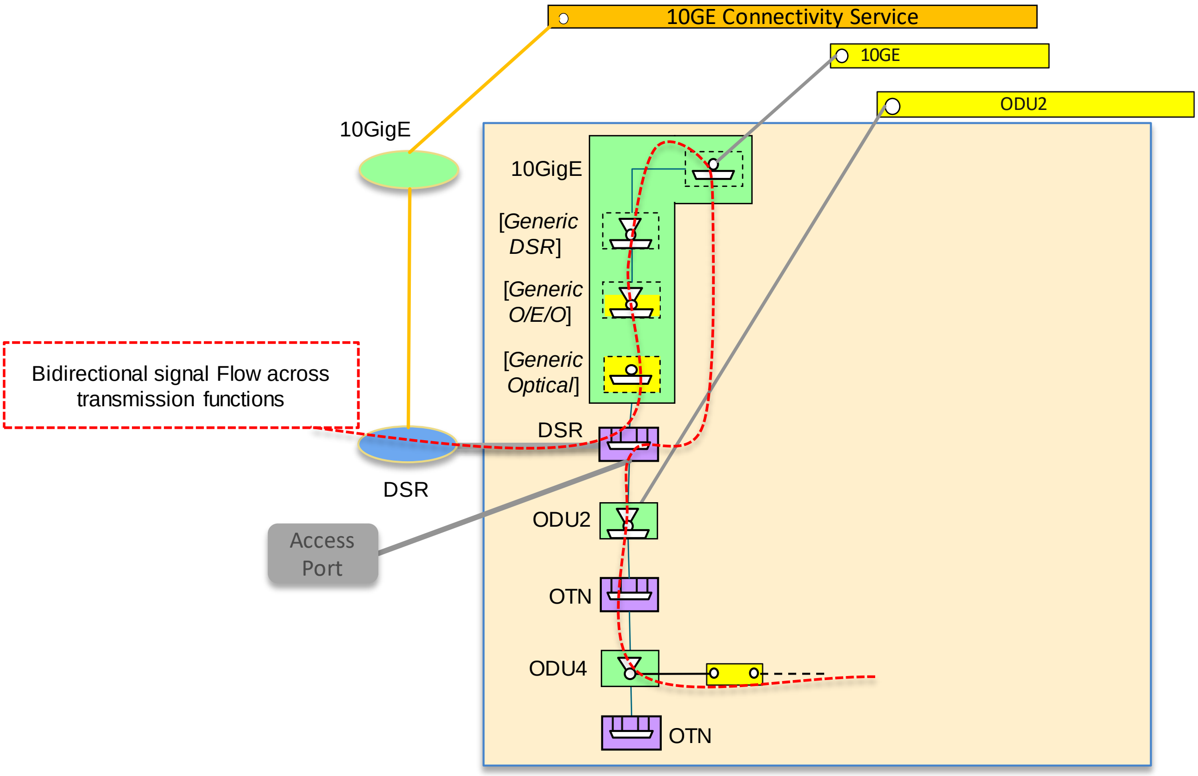


Figure 5-41 Option: DSR UNI with additional embedded functions with explicit DSR and ODU cross-connections (top), simplified without DSR cross-connection (middle), and simplified without cross-connections (bottom)

5.2.3.1.6 Explicit model of functions (electrical)

Implementations MAY also make explicit the layers below the DSR NEP at the UNI level. For the case of electrical media (e.g., 10GBASE-T, or IEEE 802.3an-2006) the Figure 5-42 shows the presence of additional ELECTRICAL_MEDIA NEPs and the corresponding generic DSR and electrical CEPs. At this stage, this version of the RIA does not model specific aspects of such layers.

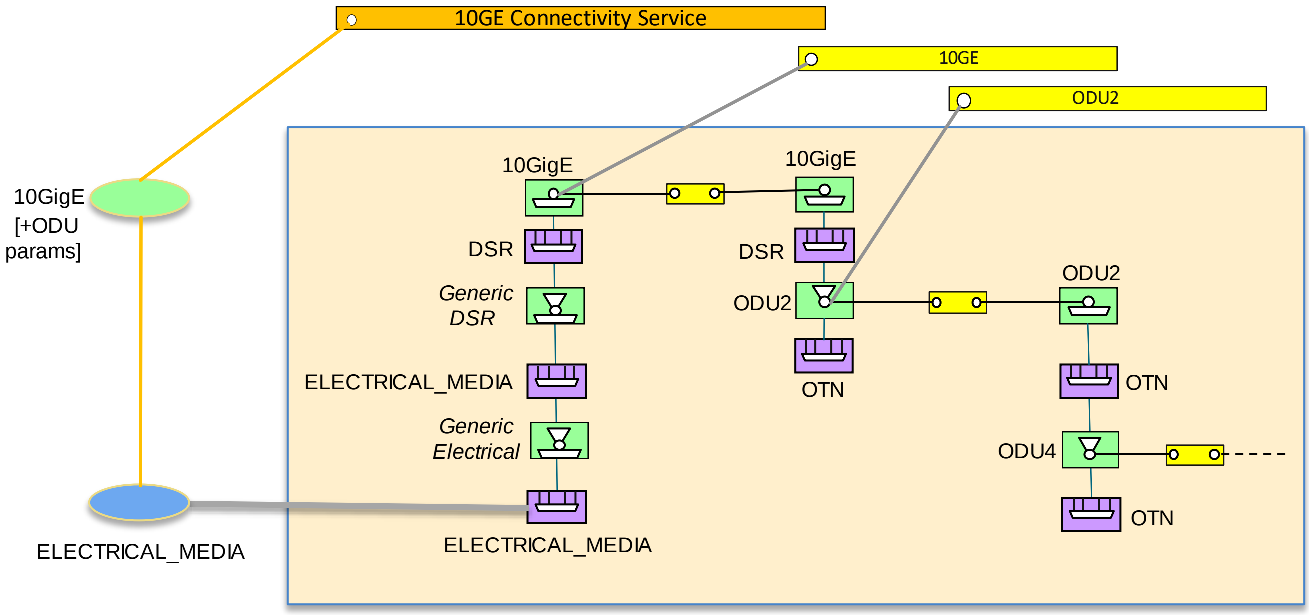


Figure 5-42 DSR UNI, explicit model of functions (electrical)

5.2.3.1.7 Explicit model of functions (optical)

Similarly, for physical layer modules based on optical transmission, Figure 5-43 represents the layer model involved below the DSR NEP.

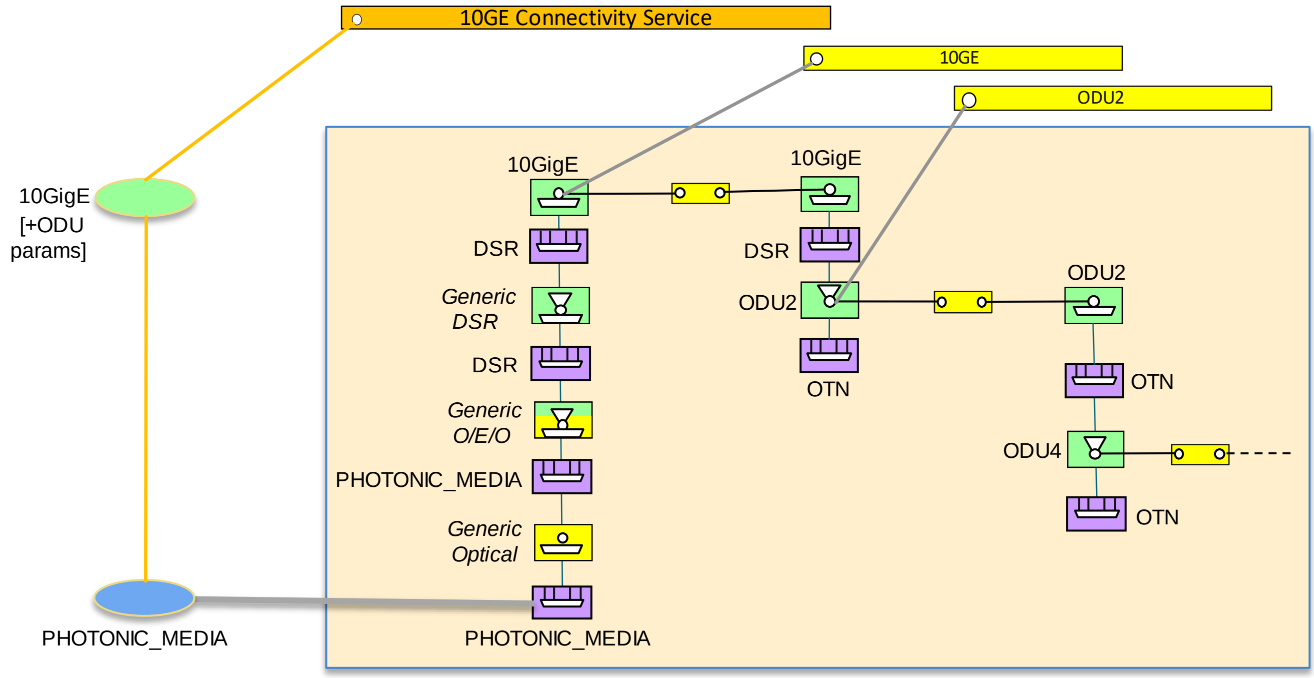


Figure 5-43 DSR UNI, explicit model of functions (optical)

NOTE: Since, at the time of writing, the Generic DSR, Generic O/E/O and Generic Optical Layer Protocol Qualifiers as well as the ELECTRICAL_MEDIA Layer Protocol Name have no defined attributes, the explicit model UNI options are presented for illustrative purposes only. Future versions of the RIA may address additional considerations as needed by the use cases.

5.2.3.2 ENNI (OTN)

To model OTN (E)NNI, several options are available based on the level of detail that is presented to TAPI clients as well as the coalescing on functions into CEPs. This section presents different modelling options along with considerations for implementations to select the most suitable ones. The options are based on the *ENNI Handoff Type*s defined by [MEF 64]. This section is to be considered as complementary with the *asymmetric connectivity service* use cases. All options include two cases:

- DSR connectivity service, in case of asymmetric DSR connectivity service (the interface at the other end is a DSR UNI)
- ODU connectivity service, in case of ODU connectivity service (the interface at the other end is another OTN ENNI).

5.2.3.2.1 Option: Directly Mapped Client Protocols

In this option (Figure 5-44), the client protocols, specifically the DSR rates, are mapped into Lower Order OTN containers of corresponding rate. For the client protocols in FIGURE, there are corresponding physical interfaces supporting the Optical Transport Unit (OTU), therefore no multiplexing is required (dotted lines on MEF 64 figure).

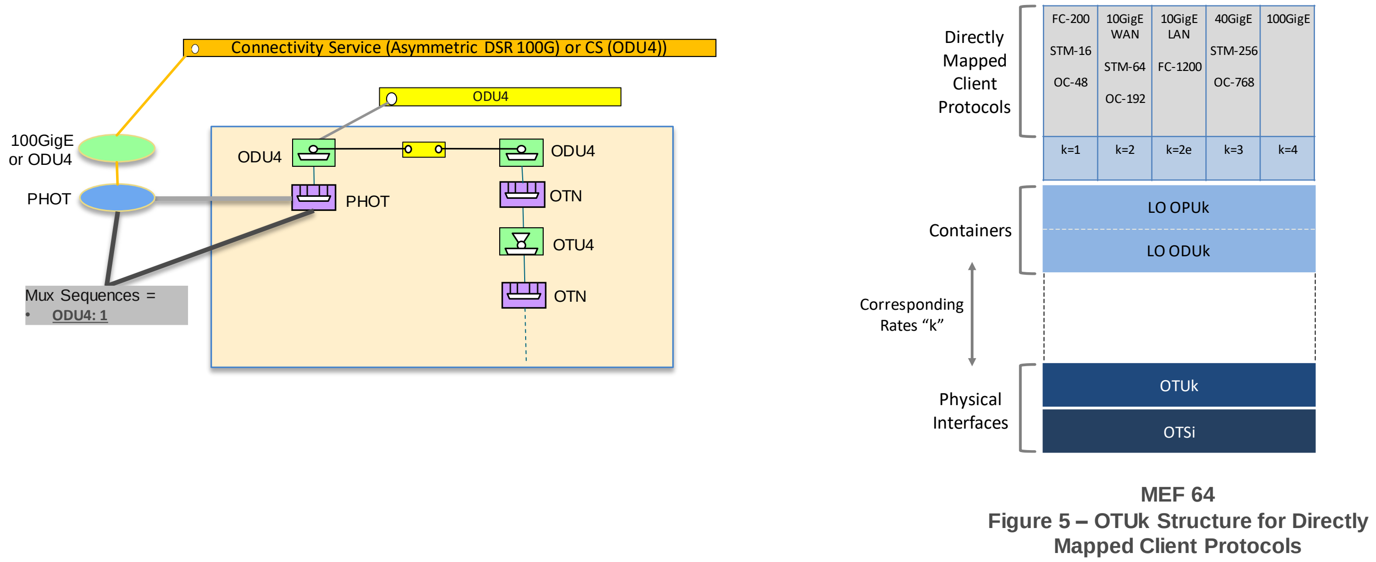


Figure 5-44 OTN ENNI, directly mapped client protocols

Figure 5-45 is a variation with the explicit instance of the OTU CEP.

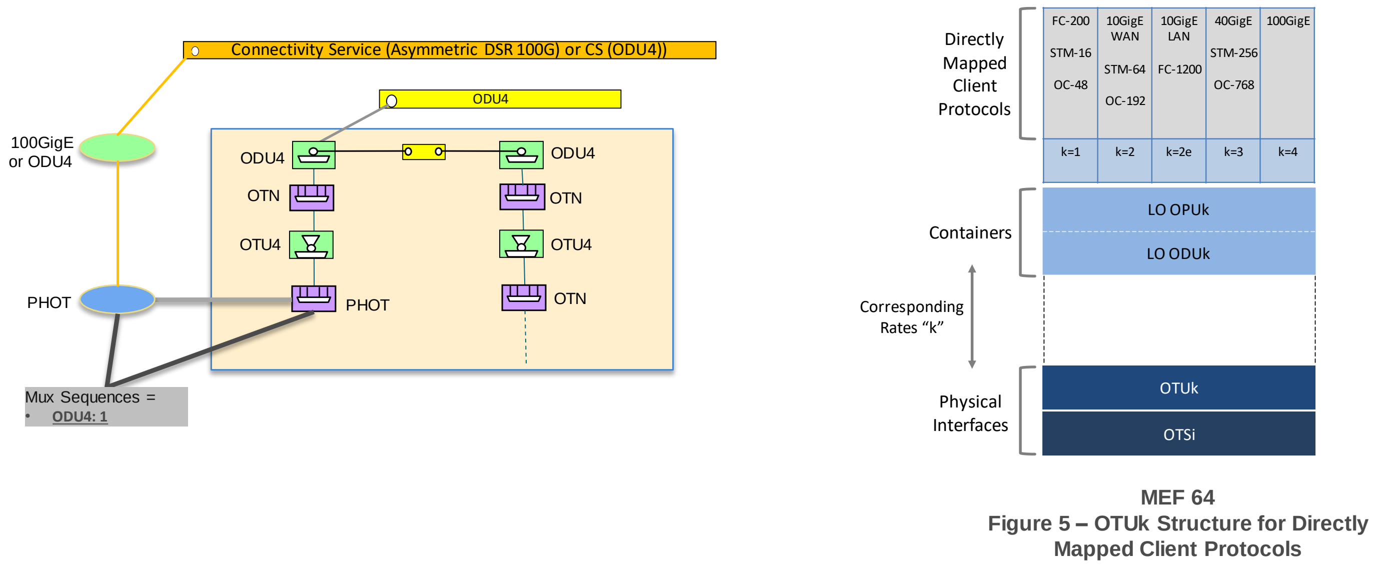


Figure 5-45 OTN ENNI, directly mapped client protocols, with OTU CEP

Figure 5-46 shows the possible embedded transmission functions.

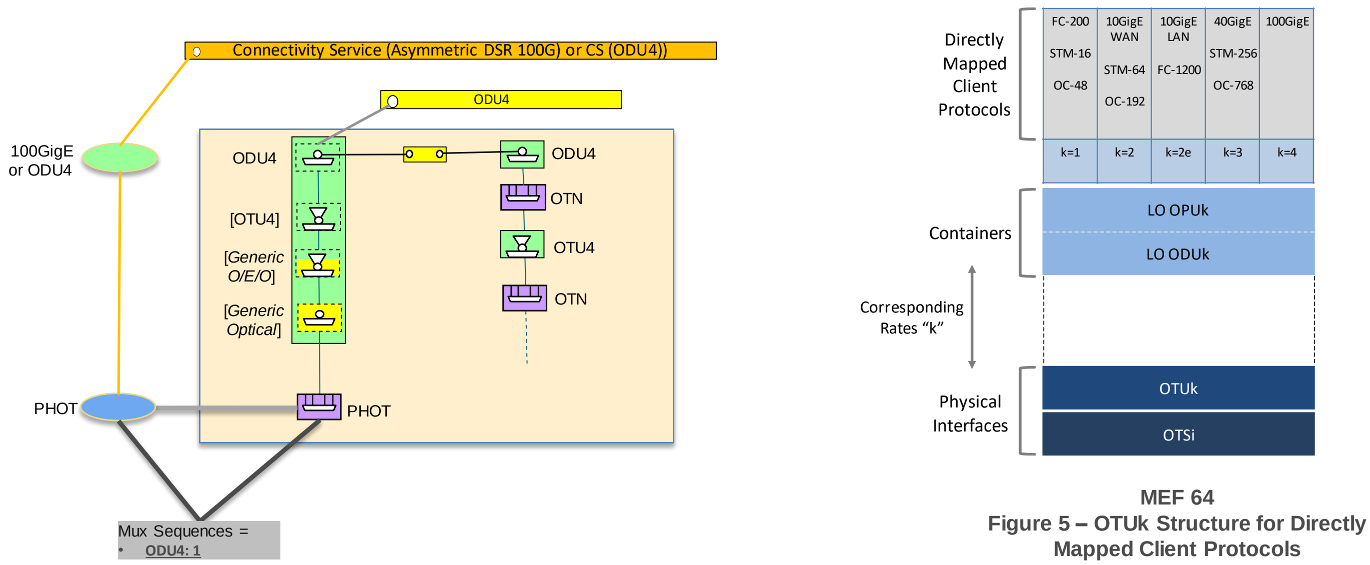


Figure 5-46 OTN ENNI, directly mapped client protocols, with additional embedded functions

Figure 5-47 shows a variation with 10GE/ODU2 layers:

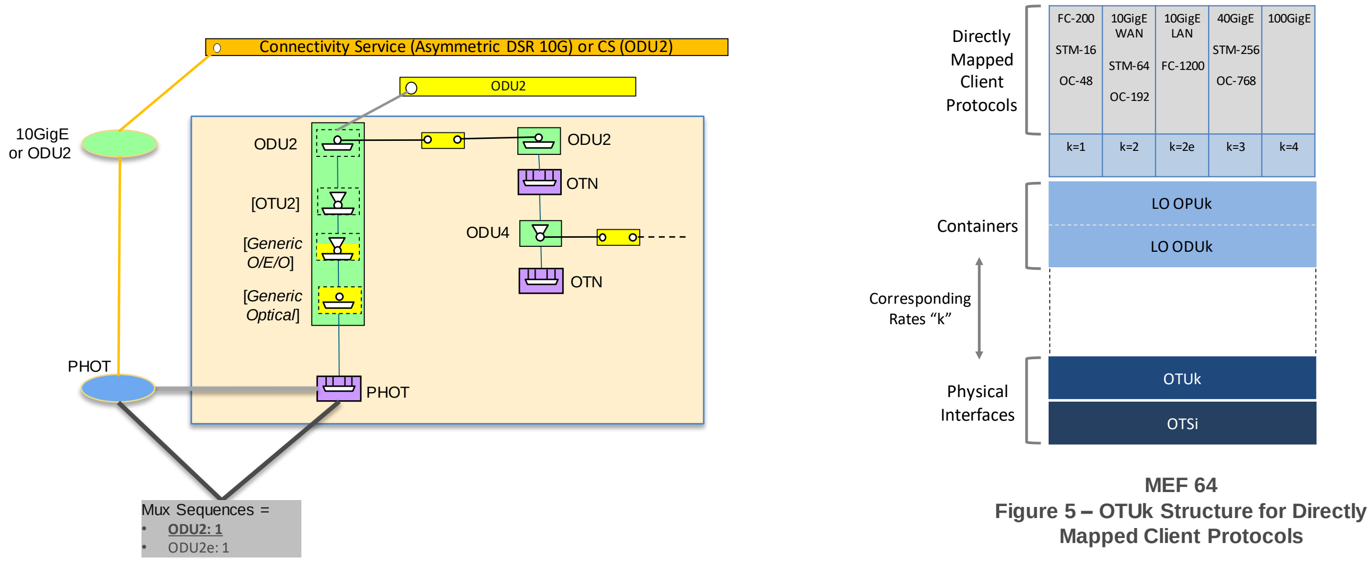


Figure 5-47 OTN ENNI, directly mapped client protocols, with additional embedded functions, 10GE/ODU2

5.2.3.2.2 Option: Mapped & Multiplexed Client Protocols

In case for the client protocols, specifically the DSR rates, no physical interfaces are defined at the same rate, the LO ODU must be multiplexed into a Higher Order ODU which do have defined physical interfaces, see Figure 5-48. Note that in this case the provisioning of the *handoff* HO ODU connectivity service shall be allowed, which depending on multiplexing feature support, can be terminated on the ENNI or more internally in the network. The other termination is located outside the domain of the TAPI management interface instance.

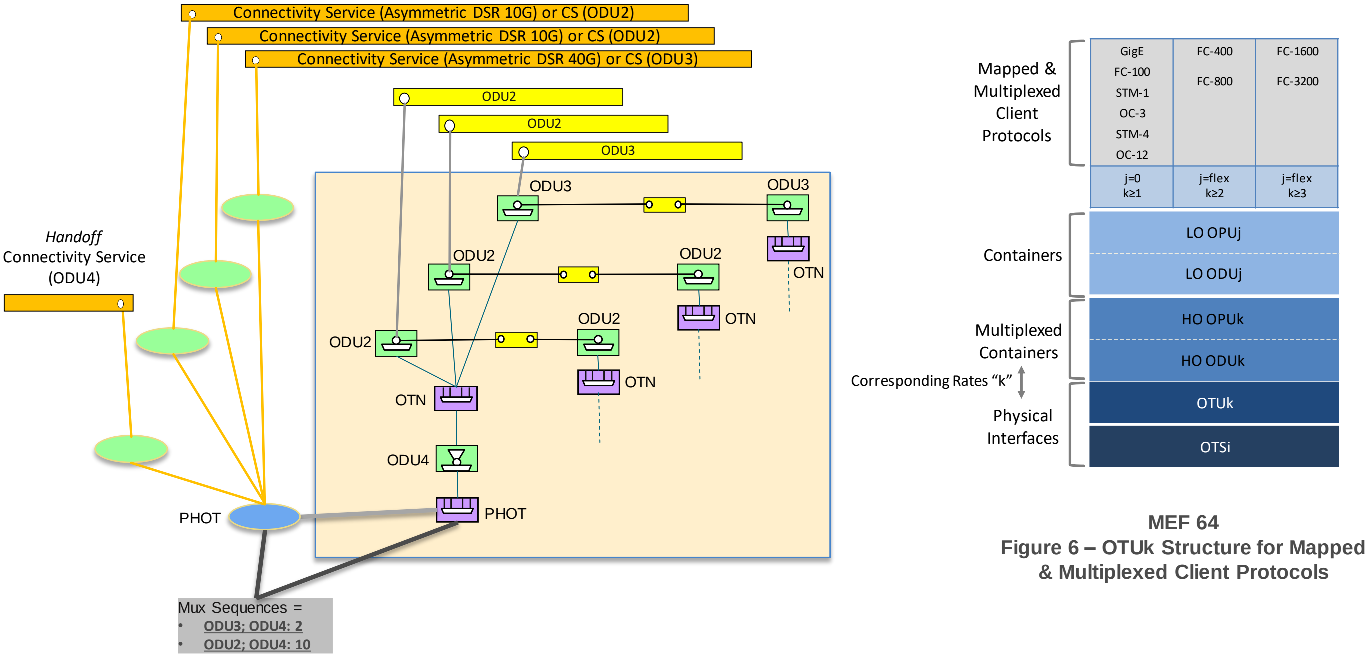


Figure 5-48 OTN ENNI, mapped & multiplexed client protocols

Figure 5-49 is a variation with the explicit instance of the OTU CEP.

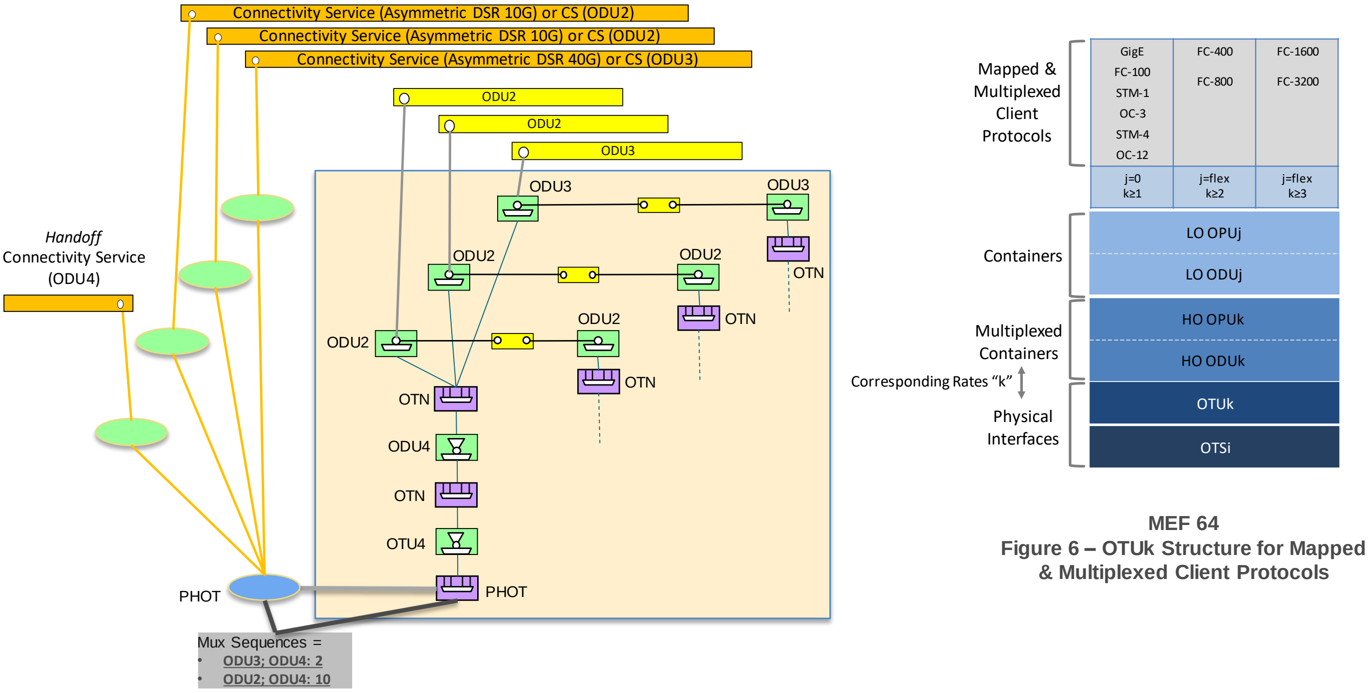


Figure 5-49 OTN ENNI, mapped & multiplexed client protocols, with OTU CEP

Figure 5-50 shows the possible embedded transmission functions.

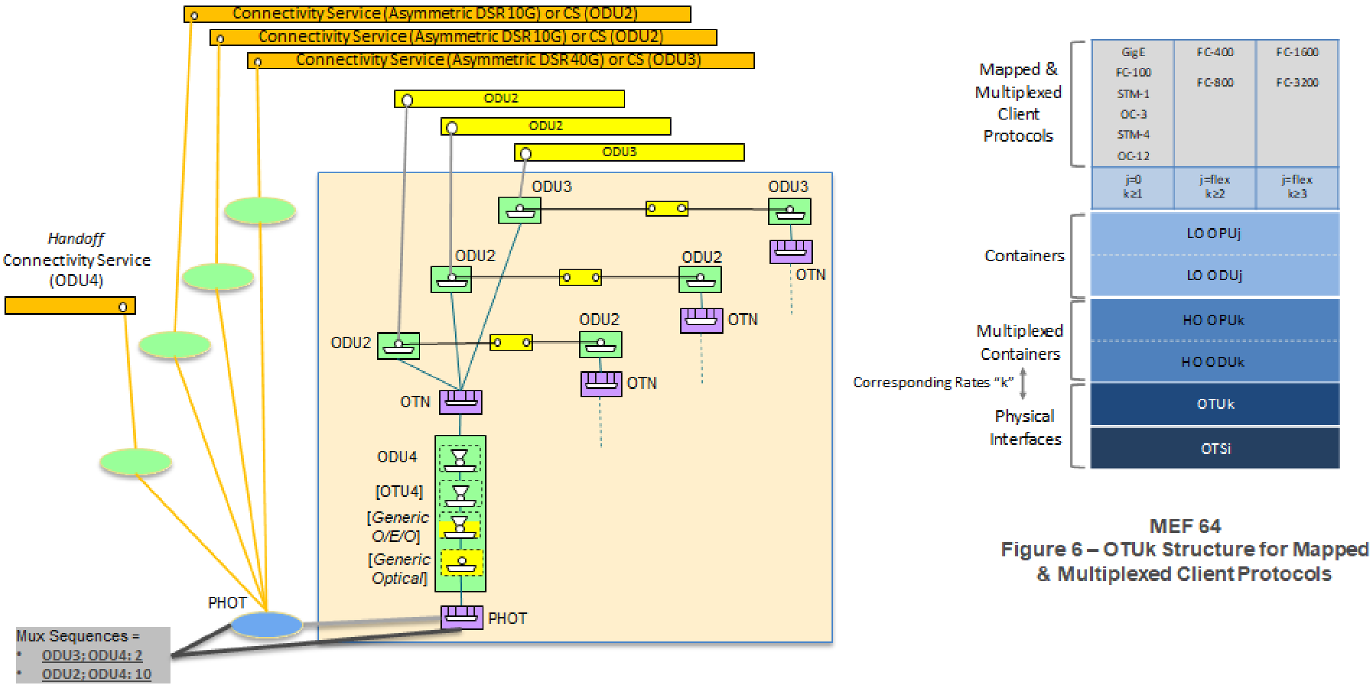


Figure 5-50 OTN ENNI, mapped & multiplexed client protocols, with additional embedded functions

Figure 5-51 shows a variation with the explicit instance of the OTU CEP.

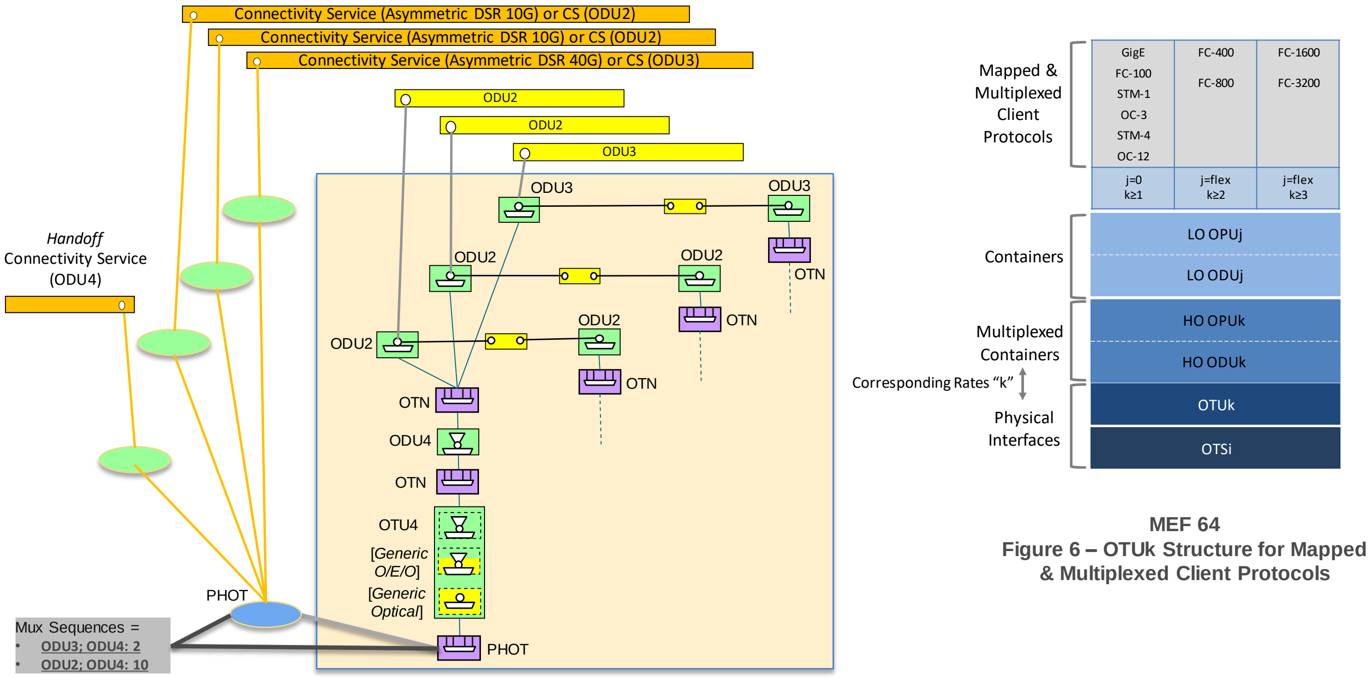


Figure 5-51 OTN ENNI, mapped & multiplexed client protocols, with additional embedded functions of OTU CEP

Figure 5-52 shows the case where the multiplexing is not supported by the edge node.

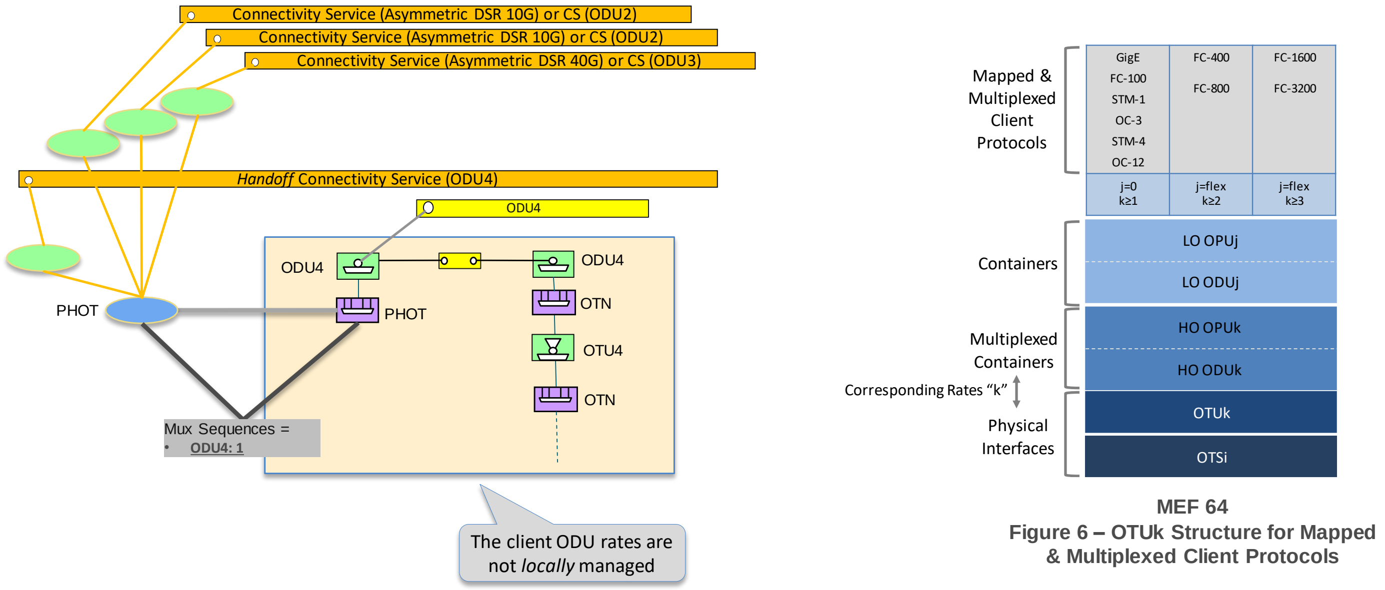


Figure 5-52 OTN ENNI, not locally mapped & multiplexed client protocols

Figure 5-53 shows a variation with the explicit instance of the OTU CEP.

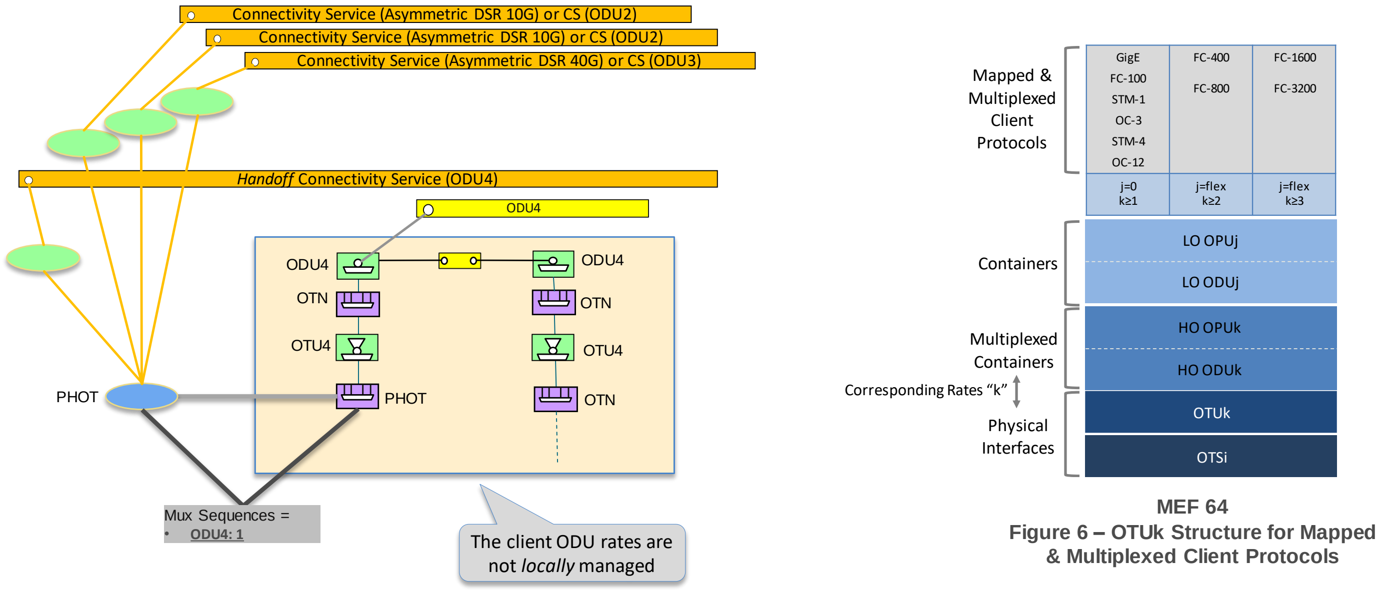


Figure 5-53 OTN ENNI, not locally mapped & multiplexed client protocols, with OTU CEP

Figure 5-54 shows the possible embedded transmission functions.

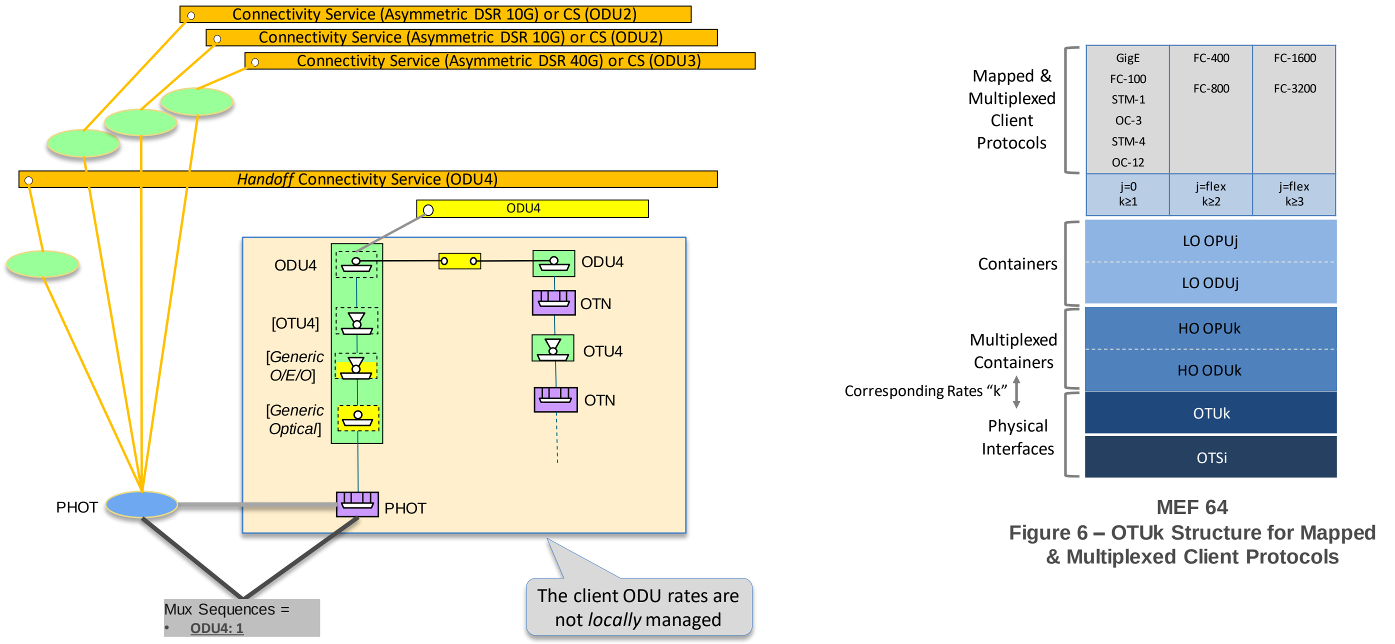


Figure 5-54 OTN ENNI, not locally mapped & multiplexed client protocols, with additional embedded functions

5.2.3.2.3 Explicit model of functions

Implementations MAY also make explicit the layers below the OUT, see Figure 5-55. At this stage, this version of the RIA does not model specific aspects of such layers.

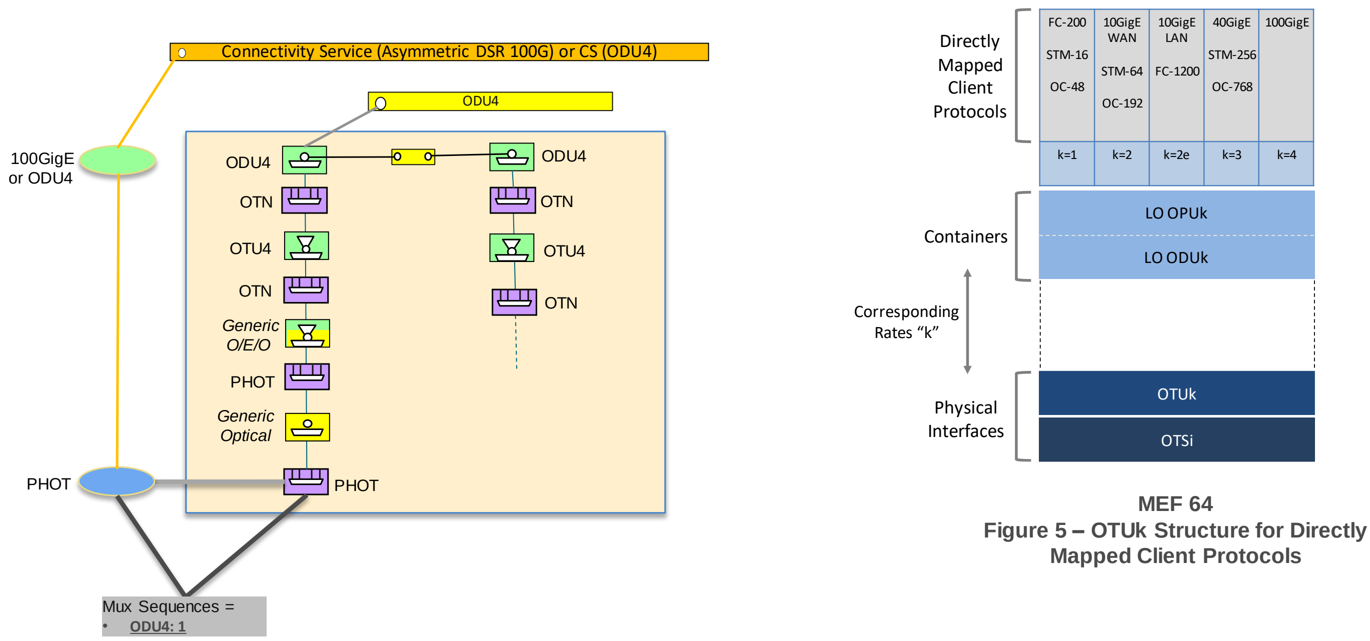
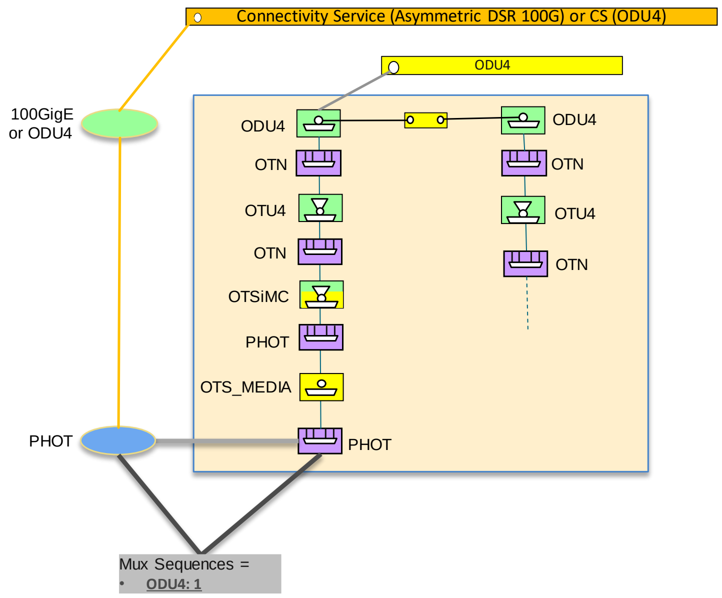
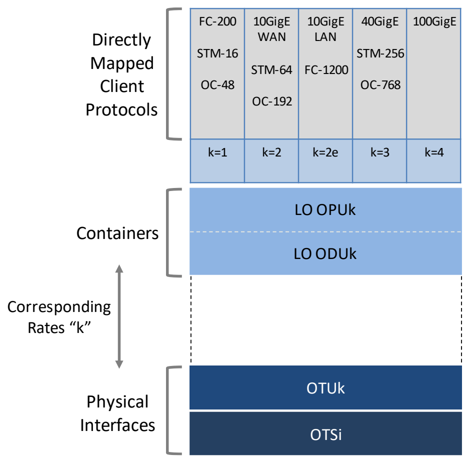


Figure 5-55 OTN ENNI, directly mapped client protocols, explicit model of functions

Figure 5-56 shows a variation with currently defined LPQs.





MEF 64
Figure 5 – OTUk Structure for Directly Mapped Client Protocols

Figure 5-56 OTN ENNI, directly mapped client protocols, explicit model of defined functions

Same explicit model can be applied to the case of mapped & multiplexed client protocols.

5.2.3.3 Multi-technology Network Interface

Some interfaces can support both DSR and OTN layers, configurable at connectivity service creation.

Figure 5-57 shows the two possible evolutions from time zero. Note that the difference with respect to fixed interfaces is the usage of generic PHOTONIC_MEDIA (or ELECTRICAL_MEDIA) instead of DSR or OTN layer protocol names.

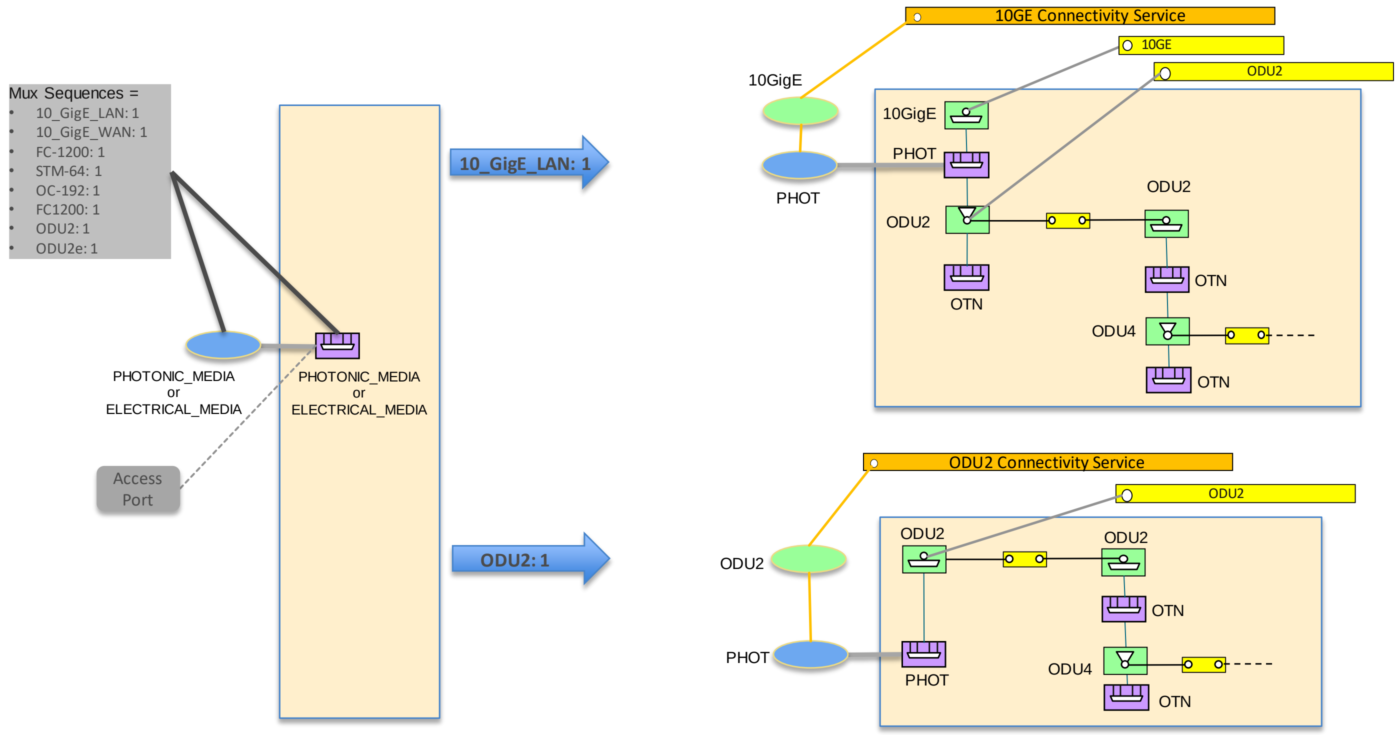


Figure 5-57 DSR/OTN NI, multi-technology interface

Figure 5-58 shows a variation with the explicit instance of the OTU CEP in the OTN case.

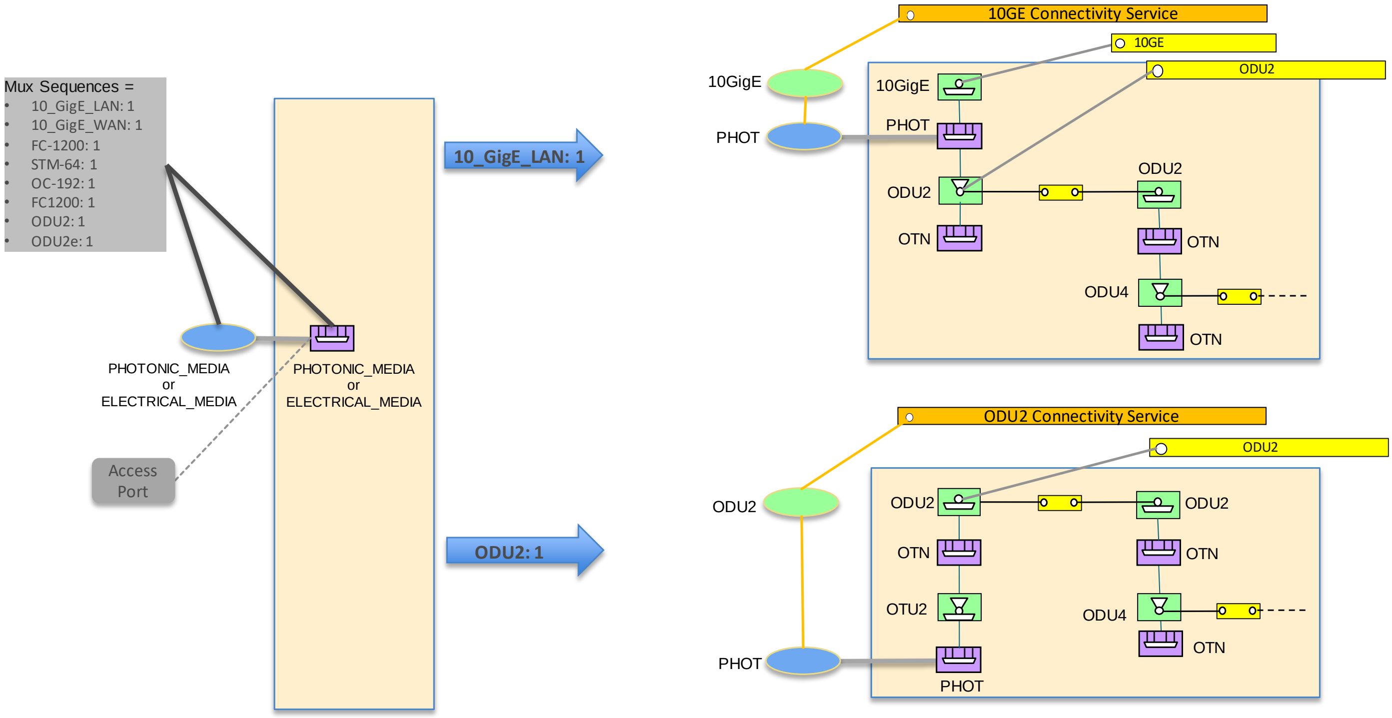


Figure 5-58 DSR/OTN NI, multi-technology interface, with OTU CEP in the OTN case

5.3 RESTCONF Responses for Common operations

NOTE: This section is experimental and waiting for feedback from implementations. At this stage this RIA does not mandate any explicit behavior.

It is acknowledged that due to the nature of optical networks a fully synchronous approach may not be suitable in all cases, and future versions of TAPI/RIA will consider two step approaches in which input validation and initial checks can be synchronous while the correct establishment of the service needs to be deferred and a subsequent asynchronous message (e.g., by means of notifications) provides an update on the status of the requested service.

The RESTCONF Server MUST implement the following responses in the RESTCONF data resources ({+RESTCONF}/data/). [Note: the first column of the table will list the error-tag specified in rfc8040#section-7 Error Reporting. A future version of this specification will add a TAPI specific sub-qualifier (complementing or in addition to the error-app-tag for such purpose). Error tags are specified in rfc6241#appendix-A (NETCONF).

Table 11: Responses for GET Operations

Error-tag	**TAPI *error-app-tag***	**HTTP Response**	**Error-info**	**Description**

		status code		
		200		Get OK response or Patch successfully modified without body
invalid-value		400, 404 or 406	<bad-attribute>: name of the missing attribute <bad-element>: name of the element that is supposed to contain the missing attribute	The request specifies an unacceptable value for one or more parameters.
(response)too-big		400	<bad-attribute>: name of the missing attribute <bad-element>: name of the element that is supposed to contain the missing attribute	The request specifies an unacceptable value for one or more parameters. An expected attribute is missing. An unexpected namespace is present. A message could not be handled because it failed to be parsed correctly. For example, the message is not well-formed XML or it uses an invalid character set.
missing-attribute		400	<bad-attribute>: name of the missing attribute <bad-element>: name of the element that is supposed to contain the missing attribute	And expected attribute is missing.
bad-attribute		400	<bad-attribute> : name of the attribute w/ bad value <bad-element> : name of the element that contains the attribute with the bad value	An attribute value is not correct; e.g., wrong type, out of range, pattern mismatch.
unknown-attribute		400	<bad-attribute> : name of the unexpected attribute <bad-element> : name of the element that contains the unexpected attribute	An unexpected attribute is present.
bad-element		400	<bad-element> : name of the element w/ bad value	An element value is not correct; e.g., wrong type, out of range, pattern mismatch.
unknown-element		400	<bad-element> : name of the unexpected element	An unexpected element is present.
unknown-namespace		400	<bad-element> : name of the element that contains the unexpected namespace <bad-namespace> : name of the unexpected namespace	An unexpected namespace is present.
malformed-message		400	None	A message could not be handled because it failed to be parsed correctly. For example, the message is not well-formed XML, or it uses an invalid character set. This error-tag is new in base:1.1 and MUST NOT be sent to old clients.
(request) too-big		413	None	The request or response (that would be generated) is too large for the implementation to handle.

access-denied		401	None	Access to the requested protocol operation or data model is denied because authorization failed.
operation-not-supported		405 or 501	None	Request could not be completed because the requested operation is not supported by this implementation.
operation-failed		412 or 500	None	Request could not be completed because the requested operation failed for some reason not covered by any other error condition.
partial-operation		500	<ok-element>: identifies an element in the data model for which the requested operation has been completed for that node and all its child nodes. This element can appear zero or more times in the <error-info> container. <err-element>: identifies an element in the data model for which the requested operation has failed for that node and all its child nodes. This element can appear zero or more times in the <error-info> container. <no op-element>: identifies an element in the data model for which the requested operation was not attempted for that node and all its child nodes. This element can appear zero or more times in the <error-info> container.	This error-tag is obsolete and SHOULD NOT be sent by servers conforming to this document. Some part of the requested operation failed or was not attempted for some reason. Full cleanup has not been performed (e.g., rollback not supported) by the server. The error-info container is used to identify which portions of the application data model content for which the requested operation has succeeded (<ok-element>), failed (<bad-element>), or not been attempted (<no op-element>).

Table 12: Responses for POST Operations

Error-tag	**TAPI *error-app-tag***	**HTTP Response status code**	**Error-info**	**Description**
		201		Post successfully created response
in-use		409	None	The request requires a resource that already is in use.
invalid-value		400, 404 or 406	None	The request specifies an unacceptable value for one or more parameters.

(response)too-big		400	<bad-attribute>: name of the missing attribute <bad-element>: name of the element that is supposed to contain the missing attribute	The request specifies an unacceptable value for one or more parameters. An expected attribute is missing. An unexpected namespace is present. A message could not be handled because it failed to be parsed correctly. For example, the message is not well-formed XML, or it uses an invalid character set.
missing-attribute		400	<bad-attribute>: name of the missing attribute <bad-element>: name of the element that is supposed to contain the missing attribute	And expected attribute is missing.
bad-attribute		400	<bad-attribute> : name of the attribute w/ bad value <bad-element> : name of the element that contains the attribute with the bad value	An attribute value is not correct; e.g., wrong type, out of range, pattern mismatch.
unknown-attribute		400	<bad-attribute> : name of the unexpected attribute <bad-element> : name of the element that contains the unexpected attribute	An unexpected attribute is present.
bad-element		400	<bad-element> : name of the element w/ bad value	An element value is not correct; e.g., wrong type, out of range, pattern mismatch.
unknown-element		400	<bad-element> : name of the unexpected element	An unexpected element is present.
unknown-namespace		400	<bad-element> : name of the element that contains the unexpected namespace <bad-namespace> : name of the unexpected namespace	An unexpected namespace is present.
malformed-message		400	None	A message could not be handled because it failed to be parsed correctly. For example, the message is not well-formed XML or it uses an invalid character set. This error-tag is new in: base:1.1 and MUST NOT be sent to old clients.
(request) too-big		413	None	The request or response (that would be generated) is too large for the implementation to handle.
access-denied		401	None	Access to the requested protocol operation or data model is denied because authorization failed.
lock-denied		409	<session-id>: session ID of session holding the requested lock, or zero to indicate a non-	Access to the requested lock is denied because the lock is currently held by another entity.

			NETCONF entity holds the lock	
resource-denied		409	None	Request could not be completed because of insufficient resources.
rollback-failed		500	None	Request to roll back some configuration change (via rollback-on-error or <discard-changes> operations) was not completed for some reason.
data-exists (post)		409	None	Request could not be completed because the relevant data model content already exists. For example, a "create" operation was attempted on data that already exists.
operation-not-supported		405 or 501	None	Request could not be completed because the requested operation is not supported by this implementation.
operation-failed		412 or 500	None	Request could not be completed because the requested operation failed for some reason not covered by any other error condition.
partial-operation		500	<ok-element>: identifies an element in the data model for which the requested operation has been completed for that node and all its child nodes. This element can appear zero or more times in the <error-info> container. <err-element>: identifies an element in the data model for which the requested operation has failed for that node and all its child nodes. This element can appear zero or more times in the <error-info> container. <no op-element>: identifies an element in the data model for which the requested operation was not attempted for that node and all its child nodes. This element can appear zero or more times in the <error-info> container.	This error-tag is obsolete, and SHOULD NOT be sent by servers conforming to this document. Some part of the requested operation failed or was not attempted for some reason. Full cleanup has not been performed (e.g., rollback not supported) by the server. The error-info container is used to identify which portions of the application data model content for which the requested operation has succeeded (<ok-element>), failed (<bad-element>), or not been attempted (<no op-element>).

Table 13: Responses for DELETE Operations

Error-tag	**TAPI *error-app-tag***	**HTTP Response**	**Error-info**	**Description**

		status code		
		204		No content – Patch successfully modified with body or successfully deleted
invalid-value		400, 404 or 406	None	The request specifies an unacceptable value for one or more parameters. An expected attribute is missing.
(response)too-big		400	<bad-attribute>: name of the missing attribute <bad-element>: name of the element that is supposed to contain the missing attribute	The request specifies an unacceptable value for one or more parameters. An expected attribute is missing. An unexpected namespace is present. A message could not be handled because it failed to be parsed correctly. For example, the message is not well-formed XML or it uses an invalid character set.
missing-attribute		400	<bad-attribute>: name of the missing attribute <bad-element>: name of the element that is supposed to contain the missing attribute	And expected attribute is missing.
bad-attribute		400	<bad-attribute> : name of the attribute w/ bad value <bad-element> : name of the element that contains the attribute with the bad value	An attribute value is not correct; e.g., wrong type, out of range, pattern mismatch.
unknown-attribute		400	<bad-attribute> : name of the unexpected attribute <bad-element> : name of the element that contains the unexpected attribute	An unexpected attribute is present.
bad-element		400	<bad-element> : name of the element w/ bad value	An element value is not correct; e.g., wrong type, out of range, pattern mismatch.
unknown-element		400	<bad-element> : name of the unexpected element	An unexpected element is present.
unknown-namespace		400	<bad-element> : name of the element that contains the unexpected namespace <bad-	An unexpected namespace is present.

			namespace> : name of the unexpected namespace	
malformed-message		400	None	A message could not be handled because it failed to be parsed correctly. For example, the message is not well-formed XML or it uses an invalid character set. This error-tag is new in: base:1.1 and MUST NOT be sent to old clients.
(request) too-big		413	None	The request or response (that would be generated) is too large for the implementation to handle.
access-denied		403	None	Access to the requested protocol operation or data model is denied because authorization failed.
rollback-failed		500	None	Request to roll back some configuration change (via rollback-on-error or <discard-changes> operations) was not completed for some reason.
operation-not-supported		405 or 501	None	Request could not be completed because the requested operation is not supported by this implementation.
operation-failed		412 or 500	None	Request could not be completed because the requested operation failed for some reason not covered by any other error condition.
partial-operation		500	<ok-element>: identifies an element in the data model for which the requested operation has been completed for that node and all its child nodes. This element can	This error-tag is obsolete, and SHOULD NOT be sent by servers conforming to this document. Some part of the requested

			appear zero or more times in the <error-info> container. <err-element>: identifies an element in the data model for which the requested operation has failed for that node and all its child nodes. This element can appear zero or more times in the <error-info> container. <no op-element>: identifies an element in the data model for which the requested operation was not attempted for that node and all its child nodes. This element can appear zero or more times in the <error-info> container.	operation failed or was not attempted for some reason. Full cleanup has not been performed (e.g., rollback not supported) by the server. The error-info container is used to identify which portions of the application data model content for which the requested operation has succeeded (<ok-element>), failed (<bad-element>), or not been attempted (<no op-element>).

6 Use Cases

Initial Considerations:

- For the RIA Use Cases, there are tables listing the "relevant parameters", which specify parameters and whether they are *Mandatory* (M), *Optional* (O) or *Conditionally mandatory* (C). These tables also list additional constraints in the allowed values as well as practical considerations.

- Further versions of this RIA will better clarify semantics of Optional parameters that are listed and not detailed in a Use Case.

There are three possible approaches to gaining and maintaining alignment (and dealing with changes):

- **Polling mode** - based on periodic polling retrieval operations and after each service creation to reconcile the actual state of the network.
- **Event triggered mode (Notifications)** - based on an initial proactive synchronization done from the NBI client module using the retrieval operations and a connection-oriented notification subscription session based on the NBI Notification mechanism described in section 2.7.
- **Compacted Log Streaming mode** – As described in [TR-548]. When using the compacted log stream approach entities should conform to the "Relevant parameters" in the "object definition" tables in the corresponding use case below.

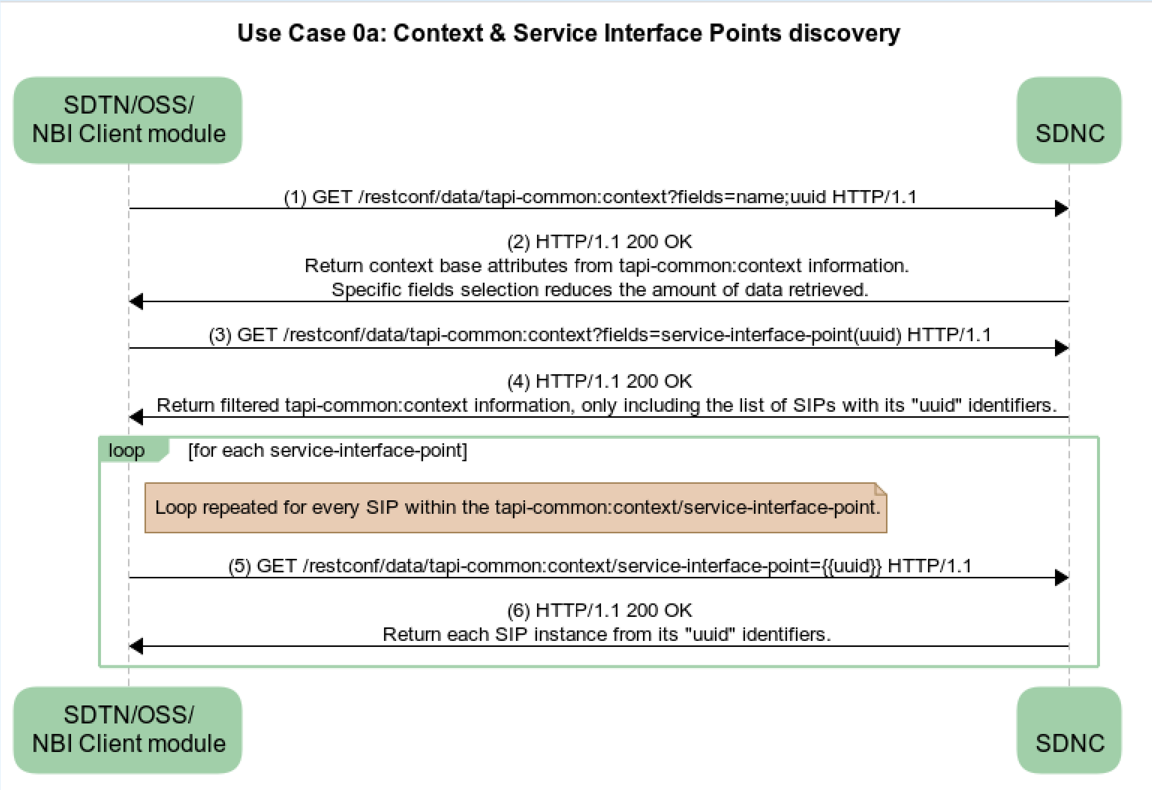
Implementations compliant with this specification MUST support the polling mechanism, MUST support the event triggered mode and MAY support compacted log mechanism.

6.1 Topology and services discovery

These use cases consist of retrieving information available from TAPI servers (SDN-C) including service-interface-points and topology. They are intended to be performed by any NBI client controller, module or application which intends to discover the logical representation of the network done by the SDN-C.

6.1.1 Use Case 0a: Context & Service Interface Points discovery

Number	**UC0a**
Name	**Context & Service Interface Points discovery**
Technologies involved	Optical
Process/Areas Involved	Planning and Operations
Brief description	The TAPI Context and Service Interface Points are the relevant network service information required before any connectivity-service creation operation. The discovery of this information is intended to be requested periodically and/or on-demand basis, proactively from the TAPI client role, to synchronize the context information.
Layers involved	DSR/DIGITAL_OTN/PHOTONIC_MEDIA
Type	Discovery
Description & Workflow	This use case consists of retrieving context and service-interface-point (SIP) information (Figure 6-1). If the first operation (1) is correctly supported by the NBI server, it MUST retrieve the context

	filtered by fields (name and uuid) (2). The response operation MUST provide the context with the parameters uuid and name as defined in Table 14 . The second operation (3) retrieves the list of service-interface-point (SIP) "uuid" (4), to recursively retrieve the full content of each SIP object in operation (5) which employs the "fields" query parameter to obtain only the desired filtered information. The response operation (6) MUST contain the parameters included in Table 15 which are marked as Mandatory (M) in the Support (Sup) column.  Figure 6-1 UC-0a: Context and Service Interface Point - Workflow.

6.1.1.1 Relevant parameters

Table 14: Context object definition

Context	**/tapi-common:context**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
uuid	As per RFC 4122.	RO	M	• Provided by *tapi-server*
name	List of {value-name, value} which MUST include: "value-name": "CONTEXT_NAME" "value": " [0-9a-zA-Z_]{64} "value-name": "VENDOR_NAME" "value": "[0-9a-zA-Z_]{64}"	RO	M	• Provided by *tapi-server* • CONTEXT_NAME is a user readable unstructured string tag to uniquely identify the tapi-server context. • VENDOR_NAME is a user readable unstructured string tag to uniquely identify the tapi-server owner or supplier.
service-interface-point	List of {service-interface-point}	RO	M	• Provided by *tapi-server*

				• Direct modification disallowed
profile	A common profile includes uuid and name. This RIA considers augmentations for { transmission-capability-profile tapi-oam:oam-profile, tapi-photonic-media:fiber-profile, tapi-photonic-media:transceiver-profile, tapi-photonic-media:amplification-profile tapi-photonic-media:connection-impairment-profile }	RO	C	• Provided by *tapi-server* • Profiles provide static, invariant data that groups and centralizes related information and that can be referred to by other TAPI objects, thus avoiding unnecessary duplication.
Transmission Profiles				
transmission-capability-profile	**potential-payload-structure** includes multiplexing-sequence number-of-cep-instances capacity (with value and unit)	RO	C	• Provided by *tapi-server* • See Table 21 and Table 22
OAM Profile				
tapi-oam:oam-profile	pm-data[local-id]	RO	C	• Provided by *tapi-server* • See Section 6.8 UC on OAM
Fiber Profile				
tapi-photonic-media:fiber-profile	Includes type-variety string loss-coef decimal64 fiber-pmd decimal64 effective-area decimal64	RO	C	• Provided by *tapi-server* • *Note: Implementations should refer to such profile from tapi-equipment:physical-span/abstract-strand and/or OTS_MEDIA CEPs.*
Transceiver Profiles				
tapi-photonic-media:transceiver-profile	**transceiver-standard-profile** with *application-code-rec* of type standard-application-code-rec (ITUT_G959_1, ITUT_698_1, ITUT_698_2, ITUT_G696_1, ITUT_G695) ... see yang file) *application-code* (string) **transceiver-organizational-profile** with *operational-mode* (string), *organization-identifier* (string), *common-organizational-explicit* **transceiver-explicit-profile** common-organizational-explicit common-explicit supported-standard-application-codes supported-organizational-modes	RO	C	• Provided by *tapi-server* • These containers are exclusive. Implementations are expected to have a single container in each profile. • *Note: Implementations should refer to such profile from a PHOTONIC_MEDIA NEP supporting OTSiMC CEP and/or OTSiMC CEPs to reflect current configuration and OTSiMC CSEP to reflect provisioning (CSEP profile list).* • See UC12d for additional comments.
With				
common-organizational-explicit	*Includes* frequency-range with upper-frequency and lower-frequency (in Hz) central-frequency-step (in Hz) tx-channel-power-min tx-channel-power-max rx-channel-power-min rx-channel-power-max rx-total-power-max	RO	C	• Provided by *tapi-server* • *See descriptions in the photonic-media yang file.*
common-explicit	*Includes* line-coding-bitrate max-polarization-mode-dispersion	RO	C	• Provided by *tapi-server* • *Mandatory for transceiver-explicit-profiles*

	max-chromatic-dispersion **chromatic-and-polarization-dispersion-penalty,** list, each entry including: chromatic-dispersion polarization-mode-dispersion penalty max-diff-group-delay **max-polarization-dependent-loss-penalty**, list, each entry with max-polarization-dependent-loss penalty standard-modulation-type min-osnr min-qfactor baud-rate roll-off min-carrier-spacing fec-type fec-code-rate fec-threshold other-properties array of value-names and values			• NOTE: the chromatic-and-polarization-penalty list allows mapping a given CD/PMD pair (sample) to a given penalty value. • NOTE: The *optional* max-polarization-dependent-loss-penalty is the penalty associated with the maximum acceptable accumulated polarization dependent loss. This list of pair pdl and penalty can be used to sample the function pdl = f(penalty)
supported-standard-application-codes	Optional profile-uuid (leafref to transceiver-standard-profile)	RO	C	• Provided by *tapi-server* • *This is used to refer to a supported standard application code which is supported by a given explicit profile*
supported-organizational-modes	Optional profile-uuid (leafref to transceiver-organizational-profile)	RO	C	• Provided by *tapi-server* • *This is used to refer to a supported organizational mode which is supported by a given explicit profile*
Amplification Profiles				
tapi-photonic-media: amplification-profile	*Includes* frequency-range with (in Hz) upper-frequency lower-frequency gain-range with min-gain max-gain noise-figure-range with min-noise-figure max-noise-figure extended-gain-range with min-gain max-gain max-power	RO	C	• Provided by *tapi-server* • *Note: Implementations should refer to such profile from OMS CEPs along with CEPs' amplification functions.* • *Note: In amplifiers with different NF and gain, the minimal NF is achieved when the EDFA operates at its maximal (and usually optimal, in terms of flatness) gain. The worst (maximal) NF applies when the EDFA operates at the minimal gain.*
Connection Profile				
tapi-photonic-media: connectivity-impairment-profile	*Includes* frequency-range with upper-frequency and lower-frequency in Hz	RO	C	• Provided by *tapi-server* • See UC 12d

	roadm-pmd roadm-cd roadm-pdl roadm-inband-crosstalk roadm-maxloss roadm-minloss roadm-typloss roadm-osnr roadm-noise-figure			
Context augments				
tapi-notification: notification-context	• List of {notif-subscription} • List of {notification} [RO] • List of {event-notification} [RO, new 2.4]	RW	C	• Provided by *tapi-server* • The notification context MAY be present in use cases related to notifications, depending on UC. • It is NOT REQUIRED to store the notifications / event-notifications in the context. • The list of subscriptions MUST be present IF the user has configured them.
tapi-topology: topology-context	• {network-topology-service} [RO] • List of {topology} [RO]	RO	M	• Provided by *tapi-server*
tapi-connectivity: connectivity-context	• List of {connectivity-service} • List of {connection} [RO]	RW	C	• Provided by *tapi-server* Note: see Section 2.4 regarding TAPI lists and presence containers.
tapi-path-computation: path-computation-context	• List of {path-comp-service} • List of {path} [RO]	RW	C	• Provided by *tapi-server* • *Depends on the Path Computation UC*
tapi-equipment: physical-context	• List of {device} [RO] • List of {physical-span} [RO]	RO	C	• Provided by *tapi-server*
tapi-streaming: stream-context	• List of {available-stream} • List of {supported-stream-type}	RW	C	• Provided by *tapi-server* • *Depends on the Streaming UC* • *See TR-548*
tapi-streaming: stream-admin-context	• List of {stream-monitor}	RW	C	• Provided by *tapi-server* • *Depends on the Streaming UC* • *See TR-548*
tapi-oam: oam-context	• List of {oam-service} • List of {oam-job} • List of {meg}	RW	C	• Provided by *tapi-server* • *Depends on the OAM UC*

Table 15: Service Interface Point (**SIP**) object definition

service-interface-point	**/tapi-common:context/service-interface-point**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
uuid	As per RFC 4122	**R**	M	• Provided by *tapi-server* • *NOTE: even if the Yang model allows R/W uuid, this RIA only considers SIPs with read-only uuid.*

name	The list of {value-name, value} MUST include: "value-name": "INVENTORY_ID", "value": " [0-9a-zA-Z_]{64}"	RW	M	• Initial value provided by *tapi-server* • INVENTORY_ID format is described in Section 4.2. • *NOTE: The Yang model species the list as being R/W. This RIA only considers read operations.*
direction	One of { “BIDIRECTIONAL”, “SOURCE”, “SINK” }	RO	M	• A SOURCE SIP acts as INPUT to the network domain for unidirectional CS. • A SINK SIP acts as OUTPUT from the network domain for unidirectional CS. • A BIDIRECTIONAL SIP acts as both SOURCE and SINK. • NOTE: This RIA only considers that BIDIRECTIONAL SIPs are used in BIDIRECTIONAL CS • NOTE: Unidirectional CS are defined between a SOURCE SIP and a SINK SIP.
layer-protocol-name	One of { "DSR", "DIGITAL_OTN", "PHOTONIC_MEDIA" } depending on the layer	RO	M	• Provided by *tapi-server* *NOTE: The case where a SIP could theoretically support more than one layer is left for further study. The model only supports one layer.*
administrative-state	One of {"UNLOCKED", "LOCKED"}	RW	M	• Initial value provided by *tapi-server* • Subsequent updates provided by *tapi-client or tapi-server* • *See dedicated use case UC0a.1*
operational-state	One of {"ENABLED", "DISABLED"}	RO	M	• Provided by *tapi-server* • *This attribute reflects operational state in terms of working / not working.*
lifecycle-state	One of { "PLANNED", "POTENTIAL_AVAILABLE", "POTENTIAL_BUSY", "INSTALLED", "PENDING_REMOVAL" }	RO	O	• Provided by *tapi-server*
profile	List of profile uuid refs	RO	C	• Provided by *tapi-server* • Profiles used to attach properties that are either applicable to bidirectional SIPs or are common to either Sink/Source directions (avoid duplication) or the direction can be inferred from the properties in the profile. • MUST appear if the SIP supports specific profiles.
sink-profile	List of profile uuid refs	RO	C	• Provided by *tapi-server* • Profiles that apply to the sink direction of the SIP. • MUST appear if the SIP supports specific sink profiles.
source-profile	List of profile uuid refs	RO	C	• Provided by *tapi-server* • Profiles that apply to the source direction of the SIP. • MUST appear if the SIP supports specific source profiles.
supported-cep-layer-protocol-qualifier-instances	List of immediately supported CEP Layer Protocol Qualifier, encoded as objects including: { *layer-protocol-qualifier*: The layer protocol qualifier and	RO	C	• Provided by *tapi-server* The potentially supported protocols and flows. In ITU-T terms, the potentially supported adaptation and termination functions. • All children identities defined for ["DIGITAL_SIGNAL_TYPE", "ODU_TYPE",

	number-of-cep-instances: The maximum number of supported CEP instances for this layer protocol qualifier }			"OTU_TYPE", "PHOTONIC_LAYER_QUALIFIER"] MUST be supported when applicable. *Note: This attribute is mandatory if there is no reference to a transmission capability profile (see UC0b, for the NEP). Otherwise, it MUST NOT be present.* *Note: The number of CEP instances for a given LPQ is optional.*
available-cep-layer-protocol-qualifier-instances	• List of available CEP Layer Protocol Qualifier (see also supported-cep-layer-protocol-qualifier-instances)	RO	C	• Provided by *tapi-server* • See also supported-cep-layer-protocol-qualifier-instances *Note: This attribute is mandatory if there is no available-payload-structure (see UC0b, for the NEP). Otherwise, it MUST NOT be present.* *Note: The number of CEP instances for a given LPQ is optional. In this case, this is used to convey information about exclusive LPQ (e.g., for dual purpose port).*
supported-payload-structure	• List of Payload Structure objects. Each single Payload Structure object contains a multiplexing-sequence, available number of CEP instances and available capacity.	RO	O	• Provided by *tapi-server* • For an explanation of the attributes see Table 22
available-payload-structure	• List of Payload Structure objects. Each single Payload Structure object contains a multiplexing-sequence, available number of CEP instances and available capacity.	RO	O	• Provided by *tapi-server* • For an explanation of the attributes see Table 22
total-potential-capacity/total-size	• "value": real, • "unit": *see tapi-common:capacity-unit*	RO	C	• Provided by *tapi-server* • NOTE: theoretical maximum bandwidth you can set up on the SIP. For example, 100 Gb/s. • NOTE: The use of capacity *objects*, values and units is technology-specific.
available-capacity/total-size	• "value": real, • "unit": *see tapi-common:capacity-unit*	RO	C	• Provided by *tapi-server* • NOTE: The use of capacity *objects*, values and units is technology-specific.

Table 16: Service Interface Point (**SIP**) augments

service-interface-point	**/tapi-common:context/service-interface-point**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
Photonic Media SIPs **/tapi-common:context/service-interface-point/tapi-photonic-media:photonic-media-service-interface-point-spec**				
spectrum-capability-pac	Includes the following lists: *supportable-spectrum* *available-spectrum* *occupied-spectrum*	RO	C	• Provided by *tapi-server* • NOTE: This block of parameters **MUST** augment SIPs of layer PHOTONIC_MEDIA exposing

	These are lists of spectrum bands, each band with upper-frequency lower-frequency frequency-constraint with adjustment-granularity grid-type.			MC/OTSiMC service provisioning capabilities.
power-management-capability-pac	See Section 3.2.6	RO	C	• Provided by *tapi-server*
total-power-threshold-pac	This is a list where each entry includes: spectrum with upper- and lower-frequency and total-power-upper-warn-threshold-default total-power-upper-warn-threshold-min total-power-upper-warn-threshold-max total-power-lower-warn-threshold-default total-power-lower-warn-threshold-min total-power-lower-warn-threshold-max as decimal64	RO	C	• Provided by *tapi-server* Note: this is to convey configurable power threshold crossing alerts where the user is able to provision a threshold value between the corresponding min and max (for both the lower and upper regions) assuming it is different from the default value.
When supporting the tapi-equipment model				
tapi-equipment: access-port-supports-sip	Includes access-port with device-uuid access-port-uuid	RO	C	• Provided by *tapi-server* • This MUST be present if an access port supports a SIP.

Comments on spectrum bands (supportable-spectrum, available-spectrum, occupied-spectrum)

Supportable Spectrum, Available Spectrum and Occupied Spectrum encode a list of *spectrum bands*, to denote, for example, which optical frequencies are in use. Each *spectrum band* includes its upper/lower-frequency bound (specified in Hz) as well as frequency constraints including adjustment-granularity and grid-type.

Notes:

- The upper and lower frequency values may not necessarily fit the ITU-T fixed and flexible DWDM grid constraints.
- The upper and lower frequency values may include spectrum portions which cannot be used to support services.

Such bands are used in both fixed grid and flexi-grid SIPs/NEPs. The adjustment-granularity, as per ITU-T G.694.1, is used to calculate nominal central frequencies. The grid-type specifies the reference set of frequencies used to denote allowed nominal central frequencies that may be used for defining applications. Both parameters may be used to constraint which channels / frequency-slots can be supported.

Notes:

- In fixed grid scenarios it is possible to encode multiple consecutive channels as either i) one band which aggregates such information or ii) exhaustively listing each channel separately. For example, the Available Spectrum list may include one *spectrum band* that encompasses 96 x 50 GHz channels in a fixed grid setting or, alternatively, may include 96 bands each corresponding to an individual 50 GHz channel.
- The combination of adjustment granularity and grid type informs about either ITU-T fixed or flexible grid capability. In fixed grids, the slot width is implicit (fixed grid in DWDM or CWDM).
    - e.g., if grid type = DWDM then the adjustment granularity informs about the fixed slot width.
    - e.g., if grid type = FLEX then the adjustment granularity informs about the minimum slot width (two times the adjustment granularity value).

6.1.2 Use Case 0b: Topology discovery

Number	**UC0b**
Name	**Topology discovery**
Technologies involved	Optical
Process/Areas Involved	Planning and Operations
Brief description	The TAPI Topology is the relevant network logical representation information required for inventory, traffic-engineering, or provisioning purposes. The discovery of this information is intended to be requested periodically and/or on-demand basis, proactively from the TAPI client role, to synchronize the context information.
Layers involved	DSR/ODU/PHOTONIC_MEDIA
Type	Discovery
Description & Workflow	The topology discover use case consists of the workflow and operations depicted in Figure 6-2. As stated in Section 4, this RIA does not specify uses for ***nw-topolopy-service.*** Therefore, a workflow based upon the topology-context is proposed. Following the message sequence in the figure: a) Sequence (1) & (2) retrieves the list of topology references (UUID) included in the ***tapi-topology:topology-context*** ○ Note that this RIA only details a single topology (see Section 4.1 and [TAPI-TOP-MODEL-REQ-1]) b) Sequence (3) & (4) retrieves the topology with a reference found in (a), where operation (3) is used to request a topology object instance by uuid filtered to provide the key parameters of the topology (4) including parameters as defined in Table 17 (i.e., uuid, name and layer-protocol-name). This sequence is repeated for each topology reference provided from (a) c) Sequence (5) & (6) retrieves the list of node references (UUIDs) for a topology found in (a). This sequence is repeated for each topology reference provided from (a) d) Sequence (7) & (8) retrieves the details of the node with a reference found in (c), where operation (7) is used to request a node by uuid with no filters so as to provide a full node subtree (8), including: ○ The parameters of the node as defined in Table 18 ○ The list of node-edge-points (owned-node-edge-point) of the node ○ The parameters for each node-edge-point as defined in Table 19 ○ The list of connection-end-points of a node-edge-point ○ The parameters for each connection-end-point as defined in the relevant parameters tables defined in UC1.0. This sequence is repeated for each node, from (c), for each topology, from (a)

	e) Sequence (9) & (10) retrieves the list of link references (UUIDs) for a topology found in (a). This sequence is repeated for each topology reference provided from (a) f) Sequence (11) & (12) retrieves the details of the link with a reference found in (e), where operation (11) is used to request a link by uuid with no filters so as to provide a link (12), including the parameters of the link defined in Table 25. This sequence is repeated for each link, from (c), for each topology, from (a). The details of the Topology object mandatory parameters included in Table 17 are provided via (b), (c) and (e) above. *Note: this UC reflects an agreement in terms of retrieved elements and subsequent GET operations. This use case does not exclude that an implementation MAY additionally provide a GET operation retrieving a whole topology object*.

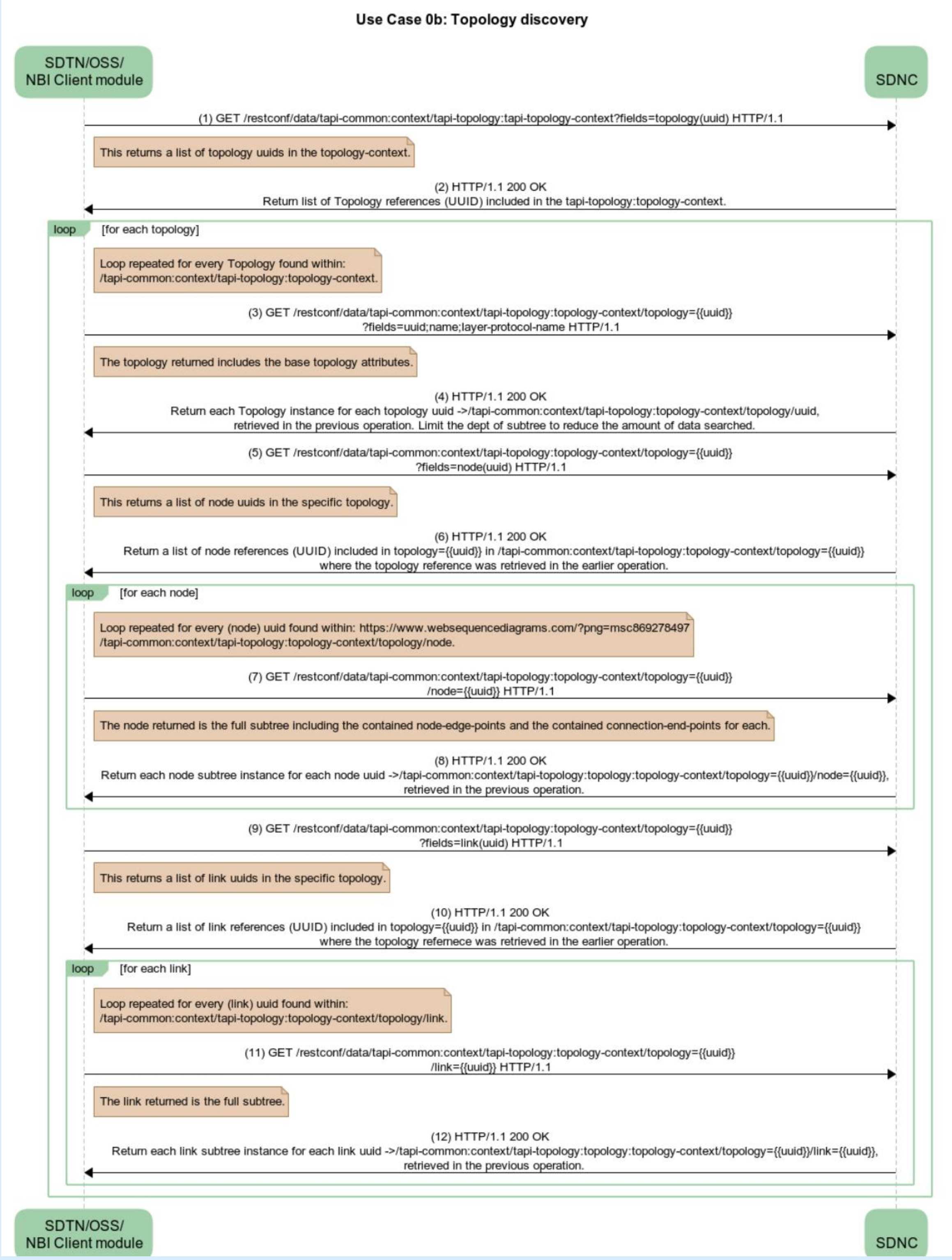


Figure 6-2 UC-0b: Topology discovery - Workflow.

6.1.2.1 Relevant parameters

These are the parameters for each object which is retrieved in the previously described RESTCONF operations.

Table 17: Topology object definition

topology	**/tapi-common:context/tapi-topology:topology-context/topology**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
uuid	As per RFC 4122	RO	M	• Provided by ***tapi-server***
name	MUST include "value-name": "TOPOLOGY_NAME" "value": " [0-9a-zA-Z_]{64}"	RO	M	• Provided by ***tapi-server*** • TOPOLOGY_NAME is a user readable unstructured string tag to uniquely identify the tapi-server topology. In case there are multiple topologies present, the T0 MUST be uniquely identified with a value prefixed with "T0_" (see Section 4)
layer-protocol-name	Leaf-List including the present Layer Protocol Names in the topology. They MUST be elements from { "DSR", "DIGITAL_OTN", "PHOTONIC_MEDIA"}	RO	M	• Provided by ***tapi-server***
link	List of {link}	RO	M	• Provided by ***tapi-server***
node	List of {node}	RO	M	• Provided by ***tapi-server***

Table 18: Node object definition

node	**/tapi-common:context/tapi-topology:topology-context/topology/node**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
uuid	As per RFC 4122	RO	M	• Provided by ***tapi-server***
name	List of {value-name: value} "value-name": "NW-NE-NAME" "value": " [0-9a-zA-Z_]{64}"	RO	M	• Provided by ***tapi-server*** • NW-NE-NAME is described in Section 4.2
profile	List of profile uuid refs	RO	C	• Provided by *tapi-server* • MUST appear if the Node supports specific profiles.
layer-protocol-name	List including elements from { "DSR", "DIGITAL_OTN", "PHOTONIC_MEDIA" }	RO	M	• Provided by ***tapi-server***
administrative-state	One of {"UNLOCKED", "LOCKED"}	RO	M	• Provided by ***tapi-server*** *NOTE: The RO needs to be considered that it is reflecting other mechanisms outside TAPI to change the administrative state.*
operational-state	One of {"ENABLED", "DISABLED"}	RO	M	• Provided by ***tapi-server***
lifecycle-state	One of { "PLANNED", "POTENTIAL_AVAILABLE", "POTENTIAL_BUSY", "INSTALLED", "PENDING_REMOVAL" }	RO	O	• Provided by ***tapi-server***
total-potential-capacity/total-size	• "value": real, • "unit": *see tapi-common:capacity-unit*	RO	O	• Provided by ***tapi-server*** • Unit depends on layer

available-capacity/total-size	• "value": real, • "unit": *see tapi-common:capacity-unit*	RO	O	• Provided by ***tapi-server*** • Unit depends on layer
cost-characteristic	List of {cost-name: cost-value} • "cost-name": "HOP_COUNT" "cost-value": "[0-9]{8}"	RO	O	• Provided by ***tapi-server***
latency-characteristic	List of { traffic-property-name: fixed-latency-characteristic } • "traffic-property-name": "FIXED_LATENCY" "fixed-latency-characteristic": "[0-9]{8}"	RO	O	• Provided by ***tapi-server***
risk-characteristic	List of {risk-characteristic-name and risk-identifier-list} • "risk-characteristic-name": ["SRNG"] "risk-identifier-list": List of string	RO	C	• Provided by *tapi-server* • *This RIA proposes at least one risk characteristic named "SRNG" along with a list of identifiers.* • *Used in UC3d* • *TBD in Path Computation Uses*
encap-topology	{"***topology-ref***"}	RO	O	• Provided by ***tapi-server*** • Needed if encapsulated-topology is supported
aggregated-node-edge-point	List of {"***node-edge-point-ref***"}	RO	O	• Provided by ***tapi-server*** • Needed if encapsulated-topology is supported
owned-node-edge-point	List of {***node-edge-point***}	RO	M	• Provided by ***tapi-server*** • See Table 19
node-rule-group	List of {***node-rule-group***}	RO	C	• Provided by ***tapi-server*** • See Table 23

Table 19: Node-edge-point (**NEP**) object definition

node-edge-point	**/tapi-common:context/tapi-topology:topology-context/topology/node/owned-node-edge-point**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
uuid	As per RFC 4122	RO	M	• Provided by *tapi-server*
name	MUST include "value-name": "INVENTORY_ID", "value": " [0-9a-zA-Z_]{64}"	RO	M	• Provided by *tapi-server* • INVENTORY_ID format is described in Section 4.2
layer-protocol-name	One of {"DSR", "DIGITAL_OTN", "PHOTONIC_MEDIA"}	RO	M	• Provided by *tapi-server*
Supported CEP instances				
supported-cep-layer-protocol-qualifier-instances	List of immediately supported CEP Layer Protocol Qualifier, encoded as objects including: { *layer-protocol-qualifier*: The layer protocol qualifier and *number-of-cep-instances*: The maximum number of supported CEP instances for this layer protocol qualifier }	RO	C	• Provided by *tapi-server* The potentially supported protocols and flows. In ITU-T terms, the potentially supported adaptation and termination functions. • All children identities defined for ["DIGITAL_SIGNAL_TYPE", "ODU_TYPE", "OTU_TYPE", "PHOTONIC_LAYER_QUALIFIER"] MUST be supported when applicable. *Note: This attribute is mandatory if there is no reference to a transmission*

				capability profile (see next). Otherwise, it MUST NOT be present. *Note: The number of CEP instances for a given LPQ is optional.*
available-cep-layer-protocol-qualifier-instances	List of available CEP Layer Protocol Qualifier (see also supported-cep-layer-protocol-qualifier-instances), including: *layer-protocol-qualifier*: The layer protocol qualifier and *number-of-cep-instances*: The number of available supported	RO	C	• Provided by *tapi-server* • See also supported-cep-layer-protocol-qualifier-instances *Note: This attribute is mandatory if there is no available-payload-structure (see next). Otherwise, it MUST NOT be present.* *Note: The number of CEP instances for a given LPQ is optional. In this case, this is used to convey information about exclusive LPQ (e.g., for dual purpose port).*
Supported payload structures.				
supported-payload-structure	List of Payload Structure objects. Each single Payload Structure object contains a multiplexing-sequence, available number of CEP instances and available capacity.	RO	O	• Provided by *tapi-server* • For an explanation of the attributes see Table 22
available-payload-structure	List of Payload Structure objects. Each single Payload Structure object contains a multiplexing-sequence, available number of CEP instances and available capacity.	RO	O	• Provided by *tapi-server* • For an explanation of the attributes see Table 22
profile	List of profile uuid refs	RO	C	• Provided by *tapi-server* • Profiles used to attach properties that are either applicable to bidirectional NEPs or are common to either Sink/Source directions (avoid duplication) or the direction can be inferred from the properties in the profile. • MUST appear if the NEP supports specific profiles.
sink-profile	List of profile uuid refs	RO	C	• Provided by *tapi-server* • Profiles that apply to the sink direction of the NEP. • MUST appear if the NEP supports specific sink profiles.
source-profile	List of profile uuid refs	RO	C	• Provided by *tapi-server* • Profiles that apply to the source direction of the NEP. • MUST appear if the NEP supports specific source profiles.
administrative-state	One of {"UNLOCKED", "LOCKED"}	RO	M	• Provided by *tapi-server* *NOTE: The RO needs to be considered that it is reflecting other mechanisms outside TAPI to change the administrative state.*
operational-state	One of {"ENABLED", "DISABLED"}	RO	M	• Provided by *tapi-server*
lifecycle-state	One of { "PLANNED", "POTENTIAL_AVAILABLE", "POTENTIAL_BUSY", "INSTALLED", "PENDING_REMOVAL"	RO	O	• Provided by *tapi-server*

	}			
direction	One of { "BIDIRECTIONAL", "SOURCE", "SINK" }	RO	M	• A SOURCE NEP is transmitting the signal to the attached link. The flow is down the layer stack from the client side to the server side. • A SINK NEP is receiving the signal from the attached link. The flow is up the layer stack from the server side to the client side. • A BIDIRECTIONAL NEP acts as both SOURCE and SINK.
link-port-role	One of { "SYMMETRIC", "ROOT", "LEAF", "TRUNK" or "UNKNOWN" }	RO	M	• Provided by *tapi-server* • *NOTE: This RIA only considers SYMMETRIC roles*
total-potential-capacity/total-size	• "value": real, • "unit": *see tapi-common:capacity-unit*	RO	C	• Provided by *tapi-server* • *Conditioned to the Layer and Qualifier* • *MUST be used in DSR NEP to reflect the nominal maximum capacity.*
available-capacity/total-size	• "value": real, • "unit": *see tapi-common:capacity-unit*	RO	C	• Provided by *tapi-server* • *Conditioned to the Layer and Qualifier*
aggregated-node-edge-point	List of { *node-edge-point-ref* }	RO	O	• Provided by *tapi-server*
mapped-service-interface-point	List of objects including **{** service-interface-point-uuid, leafref to ***/tapi-common:context/service-interface-point/uuid*** }	RO	C	• Provided by *tapi-server* If the NEP supports a SIP, the SIP uuid MUST be listed.
inter-domain-plug-id-pac	Includes { plug-id-inter-domain-local-id, plug-id-inter-domain-remote-id }	RO	C	• Provided by *tapi-server* • See UC 0.d
cep-list/connection-end-point	List of { *connection-end-point* }	RO	M	• Provided by *tapi-server*
node-rule-group	List of { *node-rule-groups* } that refer to this NEP.	RO	C	• Provided by *tapi-server*
tapi-oam:mep-mip-list	Contains the list of associated MIP and MEP instances. (see UC17)	RO	C	• Provided by *tapi-server* MUST be present if the NEP supports OAM functions.

Table 20: Node-edge-point (**NEP**) object definition augments

node-edge-point	**/tapi-common:context/tapi-topology:topology-context/topology/node/owned-node-edge-point**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
Photonic Media NEPs ***/tapi-common:context/tapi-topology:topology-context/topology/node/owned-node-edge-point/tapi-photonic-media:photonic-media-node-edge-point-spec***				
spectrum-capability-pac	See SIP description	RO	C	• Provided by *tapi-server*

				• NOTE: This block of parameters **MUST** augment NEPs of layer PHOTONIC_MEDIA exposing MC/OTSiMC service provisioning capabilities.
power-management-capability-pac	See SIP description	RO	C	• Provided by *tapi-server*
total-power-threshold-pac	See SIP description	RO	C	• Provided by *tapi-server*
When supporting the tapi-equipment model				
tapi-equipment:access-port-supports-nep	Includes access-port with device-uuid and access-port-uuid	RO	C	• Provided by *tapi-server*

NEPs can refer to Transmission Capability profiles, which augment a common profile as follows:

Table 21: NEP Transmission Capability Profiles

profile	**/tapi-common:context/profile/transmission-capability-profile**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
potential-payload-structure	Includes a list of *Payload Structure* objects. Each single Payload Structure object contains a multiplexing-sequence, max number of CEP instances and maximum capacity.	RO	M	• Provided by *tapi-server* • See next table

Table 22: NEP Transmission Capability Profile Payload Structure

	/tapi-common:context/profile/transmission-capability-profile/supported-payload-structure			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
multiplexing-sequence	List (>0) of layer protocol qualifier reflecting one supported multiplexing sequence. For example, ODU0; ODU1; ODU2; ODU4 The first entry indicates the upper most client (non-terminated) CEP, the rest of entries indicate the server terminated CEPs (forming the mux path).	RO	M	• Provided by *tapi-server*
number-of-cep-instances	uint64, number of uppermost client CEPs (non-terminated). This relates to the first entry of the mux sequence list.	RO	M	• Provided by *tapi-server*
capacity	The capacity of the multiplexing sequence (with value and unit).	RO	C	• Provided by *tapi-server* • This attribute is to be used when the layer protocol qualifier does not allow to infer a capacity value (for example, in case of ODUFlex) • It is currently unused in PHOTONIC_MEDIA NEPs.

Please find next some examples of *Transmission Capability Profiles Payload Structures* (in the examples, a colon separates the MUX sequence, max number of CEP instances and capacity attributes, while the semicolon separates the layer protocol qualifiers within the multiplexing sequence. In the case the capacity can be inferred or does not apply, the attribute is not present).

Note: the current encoding of a multiplexing sequence as a list of layer protocol qualifiers may not allow the encoding of complex constraints such as an ODU4 that can support either ODU3 or ODU2 but not a mix of ODU3/ODU2. In other words, a transmission profile with two mux sequences {ODU2;ODU4 and ODU3;ODU4} is to be understood as a ODU4 supporting such mix.

- For an ODU NEP (100G rate)
- ODU0; ODU1; ODU2; ODU4 : *80* :
- ODU0; ODU1; ODU2; ODU3; ODU4 : *64* :
- ODUflex; ODU2; ODU3; ODU4 : *64* [64/ts] : 10G *(each ODUflex CEP can have a max capacity of 10G)*
- ODUflex; ODU2; ODU4: *80* [80/ts] : 10G

- For a DIGITAL_OTN NEP (B100G rate) for any value of n.
- ODU1; ODU2; ODU3; ODUCn : 40 [mult. by n] :
- ODU2; ODU4; ODUCn: 10 [mult. by n] :
- ODU2; ODU3: ODU4; ODUCn: 8 [mult. by n] :
- ODU3; ODU4; ODUCn: 2 [mult. by n] :

Note that the max number of CEP instances defines the actual ODUCn value (e.g., n=1, 2, 4, 8…) since the protocol layer qualifier is unique (ODU_TYPE_ODU_CN)

  - ODU1; ODU2; ODU3; ODUCn : **80** : <empty> → this reflects an ODUC2
  - ODU2; ODU3: ODU4; ODUCn : **64** : <empty> → this reflects an ODUC8

- For a PHOTONIC_MEDIA NEP (ROADM)
- OTSiMC; MC; OMS; OTS_MEDIA : 80 : <empty>
- MC; OMS; OTS_MEDIA : 200 : <empty>
- OTSiMC; OMS; OTS_MEDIA : 80 : <empty>

- For a PHOTONIC_MEDIA NEP (B100G rate)
- ODU2e;ODU4;OTU4;OTSiMC: 20 : (in case the payload structure is defined in a NEP directly supporting OTSiMC CEPs, with 20 max ODU2e CEP instances)
- ODUFlex; ODUCn; OTSiMC : **N** : 200G (in case the payload structure is defined in a NEP directly supporting OTSiMC CEPs, where **N** is the max number of ODUFlex instances)
- ODUFlex; ODUCn; OTSiMC; OTS_MEDIA : **N** : 200G (in case the payload structure is defined in a NEP directly supporting the OTS_MEDIA CEP)

- For a DSR NEP (10G rate)
- 10_GigE_LAN: 1 (For example, terminal client port supporting 1 CEP at 10 Gb/s)
- 10_GigE_WAN: 1
- FC-1200: 1

- For a DSR NEP *dual mode* (10G or 100G rate) supports two modes:
- 10_GigE_WAN: 1
- 100_GigE: 1

6.1.2.2 Criteria to add NEP Transmission Capability Profile with Payload Structures

It is expected that a NEP refers to a Transmission Capability Profile in the following cases:

- The NEP is supporting a SIP.

- The NEP is the lowest NEP present in the topology (e.g., DSR or OTS_MEDIA)

- The NEP is the lowest NEP in its Layer Protocol Name (DSR, DIGITAL_OTN, PHOTONIC_MEDIA)

- All NEPs (highly redundant)

Table 23: Node-rule-group object definition

node-rule-group	**/tapi-common:context/tapi-topology:topology-context/topology/node/node-rule-group**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
uuid	As per RFC 4122	RO	M	• Provided by *tapi-server*
name	MUST include "value-name": "NRG_NAME" "value": " [0-9a-zA-Z_]{64}"	RO	M	• Provided by *tapi-server*
node-edge-point	List of {***node-edge-point-ref}***	RO	M	• Provided by *tapi-server*
rule	List of {**rule**}	RO	M	• Provided by *tapi-server* • See Table 24

Table 24: Rule object definition

rule	**/tapi-common:context/tapi-topology:topology-context/topology/node/node-rule-group/rule**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
local-id	string	RO	M	• Provided by *tapi-server*
name	List of {value-name, value} • "value-name": "RULE_NAME" "value": " [0-9a-zA-Z_]{64}"	RO	M	• Provided by *tapi-server*
rule-type	"FORWARDING" or “IMPAIRMENT”	RO	M	• Provided by *tapi-server*
forwarding-rule	One of ["MAY_FORWARD_ACROSS_GROUP", "MUST_FORWARD_ACROSS_GROUP", "CANNOT_FORWARD_ACROSS_GROUP", "NO_STATEMENT_ON_FORWARDING"]	RO	M	• Provided by *tapi-server*

Table 25: Link object definition

link	**/tapi-common:context/tapi-topology:topology-context/topology/link**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
uuid	As per RFC 4122	RO	M	• Provided by *tapi-server*
name	MUST include "value-name": "LINK_NAME" "value": " [0-9a-zA-Z_]{64}"	RO	M	• Provided by *tapi-server*
layer-protocol-name	List of elements from {"DSR", "DIGITAL_OTN", "PHOTONIC_MEDIA"}	RO	M	• Provided by *tapi-server* Minimum list size is 1. Unless specified otherwise this RIA assumes that a given link has only ONE layer protocol name.
administrative-state	One of {"UNLOCKED", "LOCKED"}	RO	M	• Provided by *tapi-server* • *NOTE: The RO needs to be considered that it is reflecting other mechanisms outside TAPI to change the administrative state.*
operational-state	One of {"ENABLED", "DISABLED"}	RO	M	• Provided by *tapi-server*
lifecycle-state	One of { "PLANNED", "POTENTIAL_AVAILABLE", "POTENTIAL_BUSY", "INSTALLED", "PENDING_REMOVAL"}	RO	O	• Provided by *tapi-server*
direction	One of { "BIDIRECTIONAL", "UNIDIRECTIONAL"}	RO	M	• Provided by *tapi-server*
total-potential-capacity/total-size	• "value": real, • "unit": *see tapi-common:capacity-unit*	RO	O	• Provided by *tapi-server* • *If this attribute is present, it MUST be considered for the purposes of path computation and path feasibility analysis.*
available-capacity/total-size	• "value": real, • "unit": *see tapi-common:capacity-unit*	RO	O	• Provided by *tapi-server* • *If this attribute is present, it MUST be considered for the purposes of path computation and path feasibility analysis.*
resilience-type	Includes restoration-policy protection-type	RO	C	• Provided by *tapi-server* • Depends on the use case. It is mandatory for specific resilience use cases.
cost-characteristic	List of Objects including { cost-name: cost-value: cost-algorithm: } • "cost-name": "HOP_COUNT" "cost-value": "[0-9]{8}"	RO	O	• Provided by *tapi-server* • Characterize the link e.g., in path computation use cases. • *TBD in Path Computation Uses*
latency-characteristic	List of { traffic-property-name: fixed-latency-characteristic } • "traffic-property-name": "FIXED_LATENCY" "fixed-latency-characteristic": "[0-9]{8}"	RO	O	• Provided by *tapi-server* • *TBD in Path Computation Uses*

risk-characteristic	List of {risk-characteristic-name and risk-identifier-list} • "risk-characteristic-name": ["SRLG"] "risk-identifier-list": List of string	RO	C	• Provided by *tapi-server* • *This RIA proposes at least one risk characteristic named "SRLG" along with a list of identifiers.* • *Used in UC3d* • *TBD in Path Computation Uses*
node-edge-point	List of {"*node-edge-point-ref*"}	RO	M	• Provided by *tapi-server*
tapi-equipment:supporting-physical-span/physical-span/physical-span-uuid	LeafRef to the Physical Span UUID	RO	O	• Provided by *tapi-server* • This attribute should be used for PHOTONIC_MEDIA links between NEPs supporting OTS_MEDIA CEPs. • *Several links may be supported by the same physical span*

6.1.2.3 Expected results

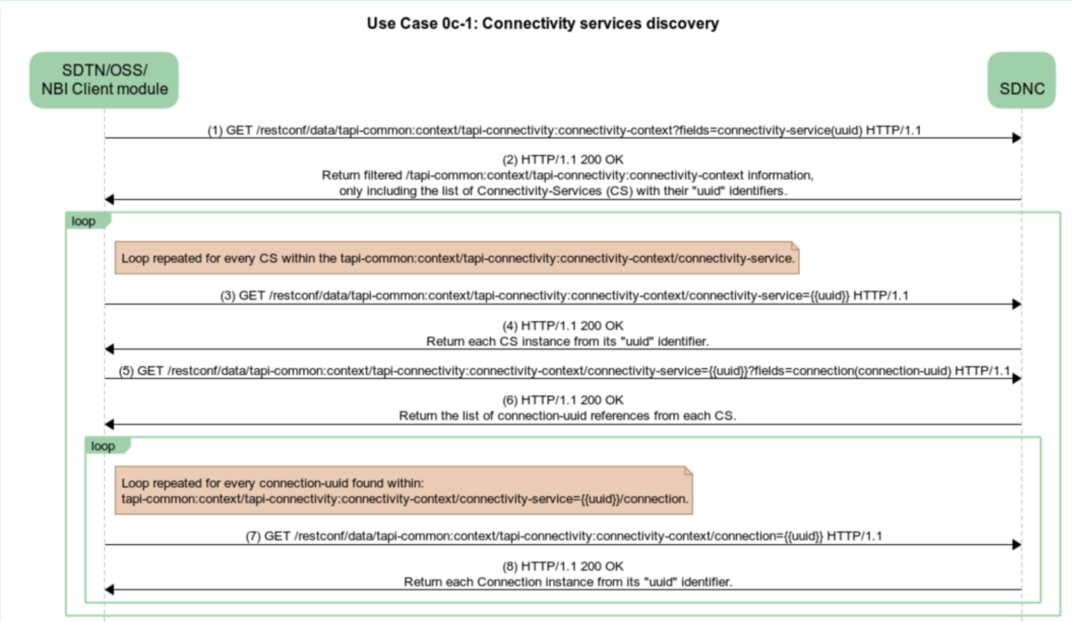
See Section 5 for the examples of detailed TAPI-Topology modelling expected at "Time 0" (i.e., after the commissioning stage of the network devices into the SDN-C, but before any service is configured).

6.1.3 Use Case 0c: Connectivity Service and Connection discovery

Number	**UC0c**
Name	Connectivity Service and Connection discovery
Technologies involved	Optical
Process/Areas Involved	Planning and Operations
Brief description	The TAPI Connectivity Service and/or Connection is a relevant network service information required for the operation. The discovery of this information is intended to be requested periodically and/or on-demand basis, proactively from the TAPI client role, in order to synchronize the connectivity information.
Layers involved	DSR, DIGITAL_OTN, PHOTONIC_MEDIA
Type	Planning
Description & Workflow	The Use Case 0c: Connectivity Service and Connection discovery consists of the retrieval of a connectivity-service and/or connections at the DSR/DIGITAL_OTN/PHOTONIC_MEDIA layers. Considering the retrieval of connections, two sub-cases MUST be supported: The first one, the connections retrieved by the uuid are obtained from a connectivity service list of "connections" (SC 0c-1). In the second one, the client retrieves all the connection uuids from the context, thus including all connections that are not referred to by any connectivity service. The client may later correlate connection uuids to referring connectivity services (SC 0c-2), if any. **SC 0c-1:** The NBI Client first retrieves the connectivity-context trimmed by the *?fields=connectivity-service* filter to retrieve all connectivity-services deployed in the TAPI Server (2). Then, iteratively the information of each Connectivity-Service (3) is requested, and also its list of Connection references (5). For all Connection reference a Connection retrieval operation is performed to get the Connection object details (7). Note that this UC also covers the direct retrieval of connections where the uuid is known directly (step 7).

The NBI server MUST return a valid object, if previous operations (4)(6)(8) succeed, which are compliant with the definition of the objects included as defined in UC1.0.

SC 0c-2: Here, the initial connectivity-service retrieval (steps (1) - (4)) is as above. Then, instead of using the connections referenced from the connectivity-service the client requests the list of connections in the context step (5) & (6) then loops through the list of connection uuids retrieved (step (7) & (8)). This allows the client to retrieve all connections including those not related to connectivity-services.



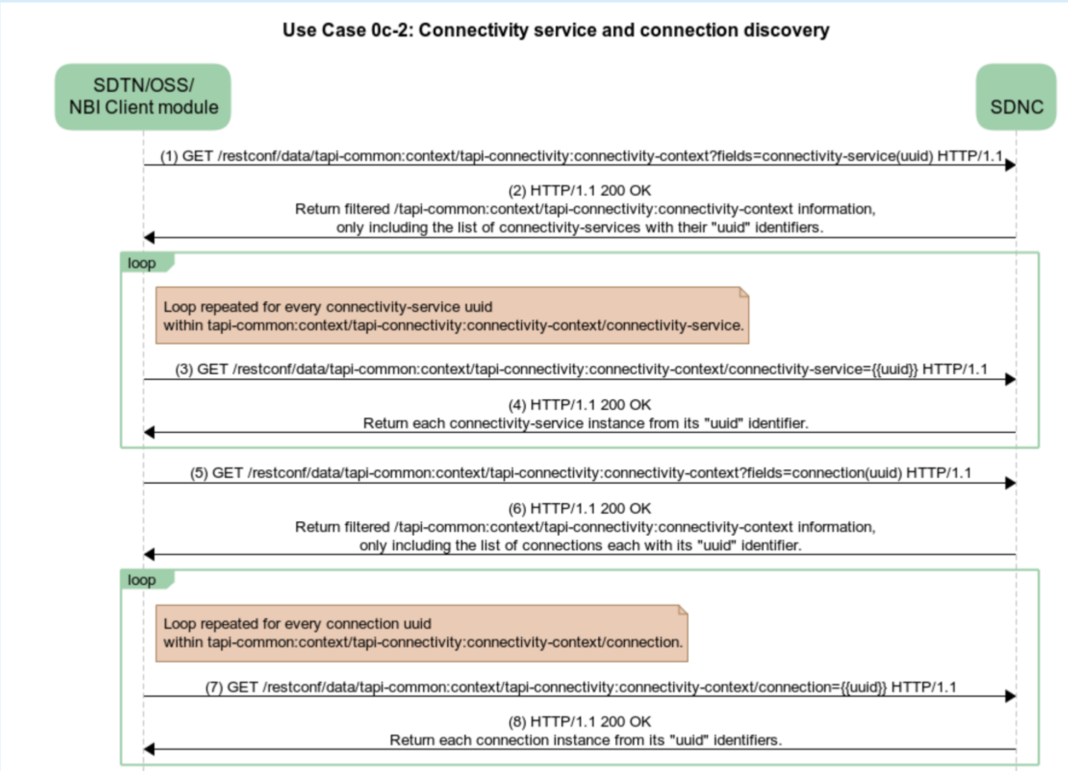


	Figure 6-3 UC-0c: Connectivity Service - Workflows UC 0c-1 (top) and UC 0c-2 (bottom)

6.1.3.1 Relevant parameters

For the details about the parameters for each object retrieved, please refer to the UC1.0, which lists the required parameters for generic unconstrained service provisioning, in which the Connectivity Services, Connections and CEP objects shall be understood as "provided by server" after the successful completion of the HTTP workflows shown above.

6.1.4 Use Case 0c.1: Mapping Connections to Physical Route

Number	**UC 0c.1**
Name	Mapping Connections to Physical Route.
Technologies involved	Optical
Process/Areas Involved	Planning and Operations
Brief description	**Disclaimer: This use case is in a draft state, the final definition will be completed based on the feedback provided by the industry upon this release of the reference specification.** The purpose of the physical route augmentation is to extend connections of the TAPI connectivity model to expose the supporting Equipment (e.g., OLP, Multiplexers, Combiners/Splitters, WSS). A TAPI Physical Route represents a list of Physical Route Elements, and each element involves an access port and its corresponding connector-pin. In other words, a Physical Route is an *ordered* list of references to connection-pins. The order the pins (thus the access ports) are traversed is the order in which they appear in the list (the list is a read-only data node and it is implicitly *ordered-by* system). A Physical Route **only** augments a Top Connection (tapi-connection) and exposes physical adjacencies providing additional information to detail how a connection is supported in terms of equipment to assist in tasks of inventory, fault management and planning activities. This UC covers the retrieval from a TAPI client of the physical route supporting a given connection. This UC only considers MC, OMS and OTS Top Connections.
Layers involved	PHOTONIC_MEDIA (MC, OMS, OTS_MEDIA qualifiers)
Type	Discovery
Description & Workflow	To illustrate the retrieval of a physical route, consider the scenario 6 (Fig 5-6 Scenario 6). The figure below depicts a possible hardware (tapi-equipment) arrangement inside the first ROADM Network Element (tapi-device).

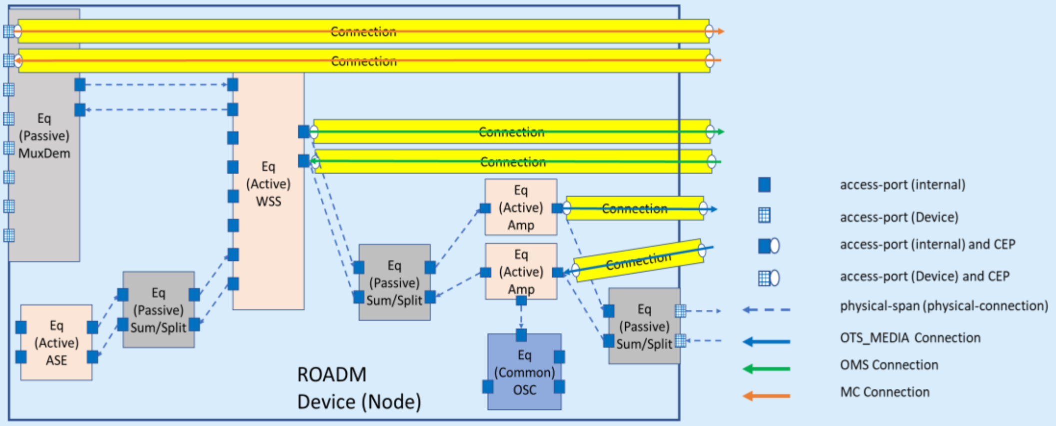


Figure 6-4: TOP Connection and Equipment within a ROADM Device

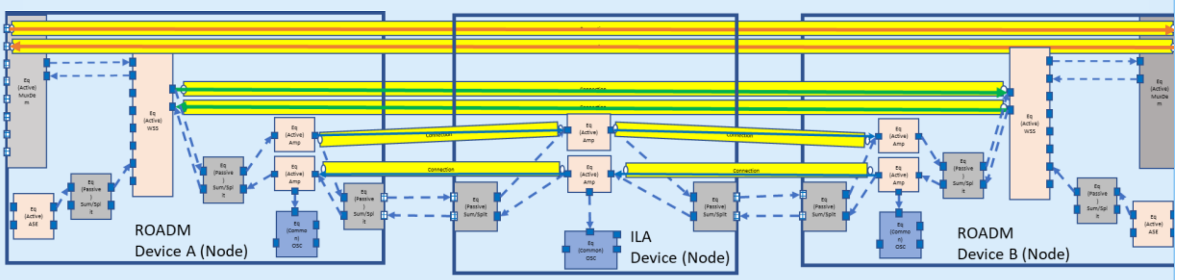


Figure 6-5: TOP Connections across ILA and ROADM devices.

For example, in the figure above, an OTS TOP Connection (blue) starts in the amplifier of the leftmost node, crosses the Passive Sum/Split, through another Sum/Split intermediate node, and ends in its amplifier. The Physical route would thus contain 6 Access Ports and the used Connector Pins.

The TAPI Physical Route is intended to augment OTS, OMS and MC connections only. Note that a Top Connection MAY be supported by more than one Physical Route (e.g., for resiliency purposes). TAPI server SHALL support the individual retrieval of Physical Route through a GET operation as described in step (1) in the figure below.

NOTE: As an augment of a connection object, the response to a client GET operation on the connection resource (i.e., via its uuid as in UC.0c) will contain the physical route. Thus, this UC focuses on retrieving only the physical-route of a connection given its uuid.

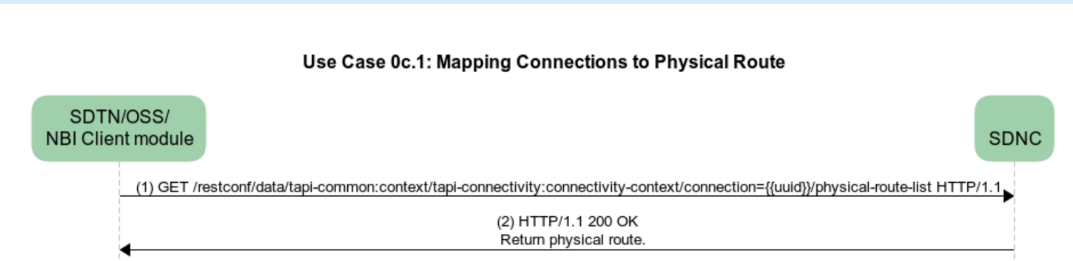


Figure 6-6: UC0c1 workflow

6.1.4.1 Relevant parameters

Table 26: physical-route-list (container) object definition

physical-route-list	**/tapi-common:context/tapi-connectivity:connection/tapi-equipment:physical-route-list**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
physical-route	List of Physical Routes	RO	M	• Provided by ***tapi-server*** It is mandatory for MC. OMS and OTS top connections.

Table 27: physical-route object definition

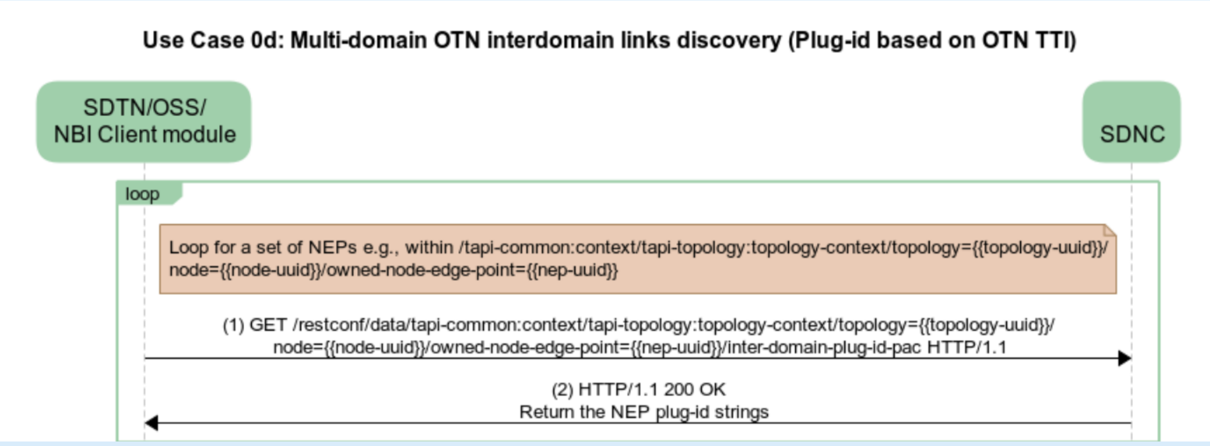
physical-route	**/tapi-common:context/tapi-connectivity:connection/tapi-equipment:physical-route-list/physical-route**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
local-id	String	RO	M	• Provided by ***tapi-server***
name	List of (value-name, value) pairs	RO	O	• Provided by ***tapi-server***
physical-route-state	Identities inheriting from PHYSICAL_ROUTE_STATE, such as CURRENT, NOT_CURRENT or UNKNOWN	RO	M	• Provided by ***tapi-server***
physical-route-element	List of {physical-route-element}	RO	M	• Provided by ***tapi-server***

Table 28: Physical Route Element object definition

used-physical-span				
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
access-port-in-route	Contains: - device-uuid - access-port-uuid	RO	C	• Provided by ***tapi-server*** • A Physical Route Element may include either: an access port, a list of connector pins or both. In case the access port is missing, the list of connector pins MUST be provided. The rationale is that it must be possible in any case to identify at least one equipment.
connector-pin-in-route	List of connector pins involved in the connection. Each entry contains: - device-uuid - equipment-uuid - connector-identification (string), - pin-identification (string) - pin-and-role (list of pin-role, pin-name and location-in-connector)	RO	C	• Provided by ***tapi-server*** • If this attribute is not present, it may mean that either all the connector pins are involved in the connection or no information on the used pins is provided (thus the physical route is a list of access-ports only). In such case the access-port-in-route MUST be present.

6.1.5 Use Case 0d: Multi-domain OTN interdomain links discovery (Plug-id based on OTN TTI).

Number	**UC 0d**

Name	Multi-domain OTN interdomain links discovery (Plug-id based on OTN TTI).
Technologies involved	Optical
Process/Areas Involved	Planning and Operations
Brief description	The objective of this use case is to define the mechanism and data structure to support the automatic discovery of OTN interdomain links between E-NNI interfaces of different network providers.This proposed mechanism allows TAPI client applications to compose a multi-domain topology among several vendors The main requirement for the TAPI Server entities (e.g SDN domain controllers) is to provide unique(s) tag which identify the E-NNI interface in both ends.The mechanism proposed in this use case is **the inter-domain-plug-id concept.**
Layers involved	DSR, DIGITAL_OTN, PHOTONIC_MEDIA
Type	Planning
Description & Workflow	The Use Case 0d: Multi-domain OTN interdomain links discovery consists of the retrieval of the inter-domain-plug-id related attribute(s) from the owned-node-edge-points objects. Please refer to the workflow included in the UC 0b Topology discovery in Section **0**. The GET operations defined in case 0b already includes the discovery of the owned-node-edge-points that include this parameter. Additionally, the TAPI server SHALL support the individual retrieval of this attribute through a GET operation as described in (1). This allows the TAPI client to retrieve the information of the inter-domain-plug-id value of each NEP individually.  Figure 6-7: UC0d workflow

6.1.5.1 Plug ID Concept

The multi-domain network composition has been traditionally performed by network operations teams manually or based on static inventory information. The Plug-ID definition in this document attempts to state a common way of correlate topology end-points of different TAPI topologies stored in different contexts. The main requirement for the TAPI Server entities (e.g., SDN domain controllers) is to provide unique tag(s) which identify uniquely the E-NNI interface in both ends. The ***tapi-topology:owned-node-edge-point*** object structure includes:


```
      +--ro topology* [uuid]
         +--ro node* [uuid]
         |  +--ro owned-node-edge-point* [uuid]
         |  |  +--ro layer-protocol-name?
         |  |  +--ro base-layer-protocol-qualifier?
         ...
         |  |  +--ro inter-domain-plug-id-pac
         |  |  |  +--ro plug-id-inter-domain-local-id?    string
         |  |  |  +--ro plug-id-inter-domain-remote-id?   string
```

The TAPI Server entity (SDN Domain Controller) must be able to automatically generate a unique pair of ids **plug-id-inter-domain-local-id** and **plug-id-inter-domain-remote-id** for the node edge point.

6.1.5.1.1 Plug ID Concept in OTN

For the case of OTN, the proposed use case consists of an autonomous and standard generation of the tags representing E-NNI/UNI interfaces connected to external network domains, based on the exchanged information across inter-domain interfaces through the OTUk, ODUk overhead TTI SAPI and DAPI identifiers.

The mechanism MAY be based on the information obtained from the OTN protocol stack, e.g., by the OTUk and ODUk frame Section Monitoring (SM) Trail Trace Identifier (TTI) (**Section 15.2/G.709/Y.1331)**. Each OTUk link end is characterized by an Access Point Identifiers (APIs) so:

- The access point identifier consists of a three-character international segment and a twelve-character national segment coded according to [ITU-T T.50]. The international segment field provides a three-character ISO 3166 geographic/political country code (G/PCC). The country code shall be based on the three-character uppercase alphabetic ISO 3166 country code. The national segment field consists of two subfields: the ITU carrier code (ICC) followed by a unique access point code (UAPC). The ITU carrier code is assigned to a network operator/service provider and shall consist of 1-6 left-justified characters, alphabetic, or leading alphabetic with trailing numeric [e.g., "USATELCORuapc"]
- each access point identifier must be appropriately unique, the access point identifier should not change while the access point remains in existence. For example, the access point identifier should be able to identify the country and network operator which is responsible for routing to and from the access point.

6.1.5.1.2 Management Considerations

G.874.1 (01/2002) Optical transport network (OTN): Protocol-neutral management information model for the network element view lists the following objects:

- **TxTI**: string[64 bytes]: The Trail Trace Identifier (TTI) information, *provisioned by the managing system* at the termination source, to be placed in the TTI overhead position of the source of a trail for transmission.
- The **Expected Destination Access Point Identifier (ExDAPI)**, *provisioned by the managing system*, to be compared with the TTI accepted at the overhead position of the sink for the purpose of checking the integrity of connectivity.
- The **Expected Source Access Point Identifier (ExSAPI),** *provisioned by the managing system*, to be compared with the TTI accepted at the overhead position of the sink for the purpose of checking the integrity of connectivity.
- **AcTI**: string[64 bytes] The Trail Trace Identifier (TTI) information recovered (Accepted) from the TTI overhead position at the sink of a trail.

Implementations following this RIA SHOULD set the following values:

Local-id: Source Access Point Identifier (SAPI) in TxTI

Remote-id: Expected Source Access Point Identifier (ExSAPI)

NOTE: This UC assumes that the TxTI and ExSAPI/ExDAPI have been provided, for example, using UC17b.1.

6.1.5.2 Relevant parameters

node-edge-point	**/tapi-common:context/tapi-topology:topologycontext/topology/node/owned-node-edge-point**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
inter-domain-plug-id-pac	includes{ plug-id-inter-domain-local-id: string plug-id-inter-domain-remote-id: string }	RO	M	• Provided by *tapi-server*. Example values for illustration purposes.

6.2 E2E Service Provisioning

6.2.1 Introduction, Definitions and Considerations

This RIA considers these main types of constraints potentially added during the provisioning of a Connectivity Service:

1. Constraints regarding the external viewpoint:
    a. CSEP constraints including parameters which apply only to the *functional boundary* of the service, like mapping type, time slots of channelized ENNI interfaces, OAM on the entire connectivity service for QoS / SLA / SLS.
    b. Coroute inclusion / diversity exclusion with respect to available connectivity services.

2. Constraints regarding the internal viewpoint:
    a. Include / exclude available Nodes, NEPs, and Links in the supporting connections.
    b. Cost parameters to be used during path computation.
    c. CSEP constraints regarding the immediate server layer, such as the bandwidth portion, e.g., time slots or spectrum. This solution is a subset of d.ii), applicable for simpler layering scenarios (i.e., only one server trail, as shown in specific use cases).
    d. Include / exclude available connections at a server layer of the connectivity service. *Note: At the server layer, the generic specification of resources (e.g., bandwidth portion, time slots or spectrum) for non-trivial cases is left for future consideration*. [explicit resource control]
    e. Include / exclude available connections at the same layer of the connectivity service (*for future consideration*), e.g., supporting the stitching of existing Connections or the usage of *orphan* Connections.

[**unconstrained**] the term **unconstrained** (UC-1X) indicates that the TAPI-Client is not introducing any of the aforementioned constraints in the service request. The provisioning relies on the capabilities of the TAPI-Server to select the network resources employed to provide the desired service characteristics.

[**server restrictions, CSEP bottom-up**] In a "bottom-up" approach, two different connectivity services (client and server, for example OTSiMC and MC) are established sequentially: first the server layer and then the client layer.

[*deprecated*] The TAPI-Client may restrict the client CSEP to use the server CSEP, *referring to* the server CSEP by its uuid (*server-connectivity-service-end-point refers to an existing CSEP)*. The following yang tree clarifies the use of server-connectivity-service-end-point. This option is deprecated since it does not allow to specify more than one server CS.

tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service={uuid}/end-point={lid}/server-connectivity-service-end-point

```
module: tapi-connectivity
  augment /tapi-common:context:
    +--rw connectivity-context
       +--rw connectivity-service* [uuid]
       |  +--rw end-point* [local-id]
       |  |  +--rw server-connectivity-service-end-point
       |  |  |  +--rw connectivity-service-uuid?              -> .../connectivity-service/uuid
       |  |  |  +--rw connectivity-service-end-point-local-id? -> .../......-service/end-point/local-id
```

[*preferred*] The client connectivity service is constrained to use one or more server CS (any server layer protocol name and qualifier), *referring to* the server connectivity service by its uuid. The following yang tree clarifies its use.

tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service={uuid}/connectivity-service/connectivity-service-uuid [list]

```
module: tapi-connectivity
  augment /tapi-common:context:
    +--rw connectivity-context
       +--rw connectivity-service* [uuid]
       |  +--rw connectivity-service* [connectivity-service-uuid]
       |  |  +--rw connectivity-service-uuid    -> ...uuid
```


[**server restrictions, top-down**] In addition to the bottom-up approach, some service provisioning Use Cases (such as UC1c, UC1g and UC2b) request a Connectivity Service adding *server layer protocol restrictions* in a "top-down" approach, thus enabling the creation of the supporting connections in a single operation. For example, such constraints MAY specify constraints of the client relative position within any server (the time slot of a ODU2 within any ODU4) or MAY specify constraints that affect the properties of the server layer (such as the absolute frequency slot of an MC). By convention, then:

- Such use cases rely on the usage of CSEP *layer protocol constraints* where appropriate to convey restrictions that apply at a given layer.
- The server MAY instantiate as many top level and supporting connections as needed.
- The server MAY create connectivity services that relate to the server restrictions. For example, the creation of an OTSiMC connectivity service MAY/MAY NOT cause the instantiation of an MC connectivity service by the server. See UC10 for the guidelines referring to connectivity service deletion [**server-allocated connectivity services**].

For use cases UC-3X:

i) Since there currently is no mechanism to indicate whether a set of constraints MUST or SHOULD be applied, by default these constraints are considered loose (i.e., best effort, a controller SHOULD not trigger a failure in case the path computation cannot find a suitable route), unless specified otherwise in a particular UC.

ii) In case the constraints are applied to a service with restoration capabilities, any reroute action SHOULD account for any constraints policy defined if possible but, as a general rule, the restoration MUST always take place even if the specified constraints enter in conflict with the new route.

Previous versions of this RIA named some of the provisioning use cases as *unconstrained.* It is now considered that the specification of the relevant parameters for such use cases (such the definition of the mapping type) corresponds to the specification of constraints (albeit simple ones). The use cases rely now on the usage of e.g., */tapi-common:context/tapi-connectivity:connectivity-context/ connectivity-service/end-point/…*

- *layer-protocol-constraint/tapi-digital-otn:odu-connectivity-service-end-point-spec/odu-csep-ttp-pac/configured-mapping-type,*
- *layer-protocol-constraint/tapi-photonic-media:otsia-csep-ttp-pac/tapi-photonic-media:number-of-otsi*

6.2.2 Network Scenarios for Provisioning Use Cases

This specification includes the following types of connectivity services:

1. DSR CS (UNI to UNI)
2. DSR Asymmetric CS (UNI to OTN ENNI)
3. ODUk Infrastructure Trail CS (INNI to INNI)
4. ODUk Handoff/Semi-terminated Trail CS (INNI to ENNI)
5. Transponder to Transponder CS (INNI to INNI):
   a. ODUk Serial Compound Link Connection CS
   b. ODUCn Trail CS
   c. ODUk Trail CS
6. OTSiMC CS (INNI to INNI or UNI to UNI in disaggregated scenario)
7. MC CS (INNI to INNI or UNI to UNI in disaggregated scenario)

For future consideration: *Mountain/Internally Symmetric* connectivity services.

There are three base scenarios for *transponder-to-transponder* connectivity set up:

- ODUk Serial Compound Link Connection Connectivity Service
- ODUCn Trail Connectivity Service
- ODUk Trail Connectivity Service

The transponder-to-transponder connectivity is the base for all DSR and OTN client connectivity use cases.

There are three base scenarios for *ROADM-to-ROADM* connectivity set up:

- MC Add/Drop Connectivity Service
- MC Degree Connectivity Service
- OTSiMC Connectivity Service without server MC

The ROADM-to-ROADM MC connectivity is the base for all the OTSiMC client connectivity use cases.

In the figures of this section the

- CSEPs are decorated by
  - green boxes representing connectivity configuration items,
  - red boxes representing OAM configuration items.
- CEPs are decorated by
  - green boxes representing connectivity state items,
  - by red boxes representing OAM state items.
- OSEPs (OAM Service End Points) are decorated by
  - red boxes representing OAM configuration items. Note that items shown in Italic font (such as the PhoOamMepServicePoint object) are considered experimental and for further study.

NOTES:

- Currently, this RIA only considers *OTSiMC services between ROADM add/drop ports* (OTSiMC services between transceivers line port are left for a further version of this RIA, along with the usage of clients other than ODU/OTU). Note that, when provisioning higher layer services in transponder-to-transponder (e.g., DSR, ODU) OTSiMC connections also appear in transponders.
- In the figures of this section the configuration parameters are shown for one CSEP of the connectivity service. In all scenarios but the asymmetric ones the CSEP configuration is assumed to be the same at both ends of the connectivity service.
- In the following scenarios, only a subset of the possible options regarding UNI and ENNI modeling are shown (e.g., no DSR cross-connection or explicit DSR). It is understood that all the previously detailed options may be used. Please see Section 5.2.3 for further details on other possible options (e.g., simplified DSR UNI).
- The presented scenarios focus on the transmission and layering parameters (in terms of protocol layer constraints) and do not systematically include additional (e.g., topological or connectivity) constraints that may also be applicable. For example, it is assumed that if a scenario relies on "*reusing an already existing connectivity service*", this implies that the client is adding the appropriate *connectivity-constraint/coroute-inclusion* parameter referring to the existing connectivity service by its uuid.
- **For simplicity, in the following scenarios OTU/OTU-Cn top-connection is not represented.** Implementations shall follow the guidelines regarding top-connections as per Section 5.1.2
- The items in *italic* are for further study, e.g., photonic OAM.
- OtsiConfig is an abbreviation of OtsiConfigPac.
- OtsiMcConfig is an abbreviation for OtsiMcBandwidthConfigPac, OtsiMcSpectrumConfigPac, OtsiMcFrequencyConfigPac.
- In the right part of the figures, the model of resulting states, the relationships between objects are simplified.
- On transponders and ROADM a/d ports, the OTS_MEDIA may be replaced by OS_MEDIA LPQ.

- OtuCsepTtpPac, OduCsepCtpPac, OduCsepTtpPac may be [omitted] in case the currently defined attributes are not applicable to the specific scenario, for example:
  - In case of ODUk supported by OTUk, the OduCsepCtpPac is not applicable.
  - In case of ODUk supporting an ODUj, the OduCsepTtpPac is not applicable.

6.2.2.1 ODUk Serial Compound Link Connection Connectivity Service

Figure 6-8 shows the configuration parameters for the provisioning of the *transponder-to-transponder* connectivity based on an ODUk container, the ODUk *Serial Compound Link Connection* (SCLC) Connectivity Service.

The result includes the OTSiMC connection plus the ODUk *unterminated* Connection. OTUk connection is considered optional.

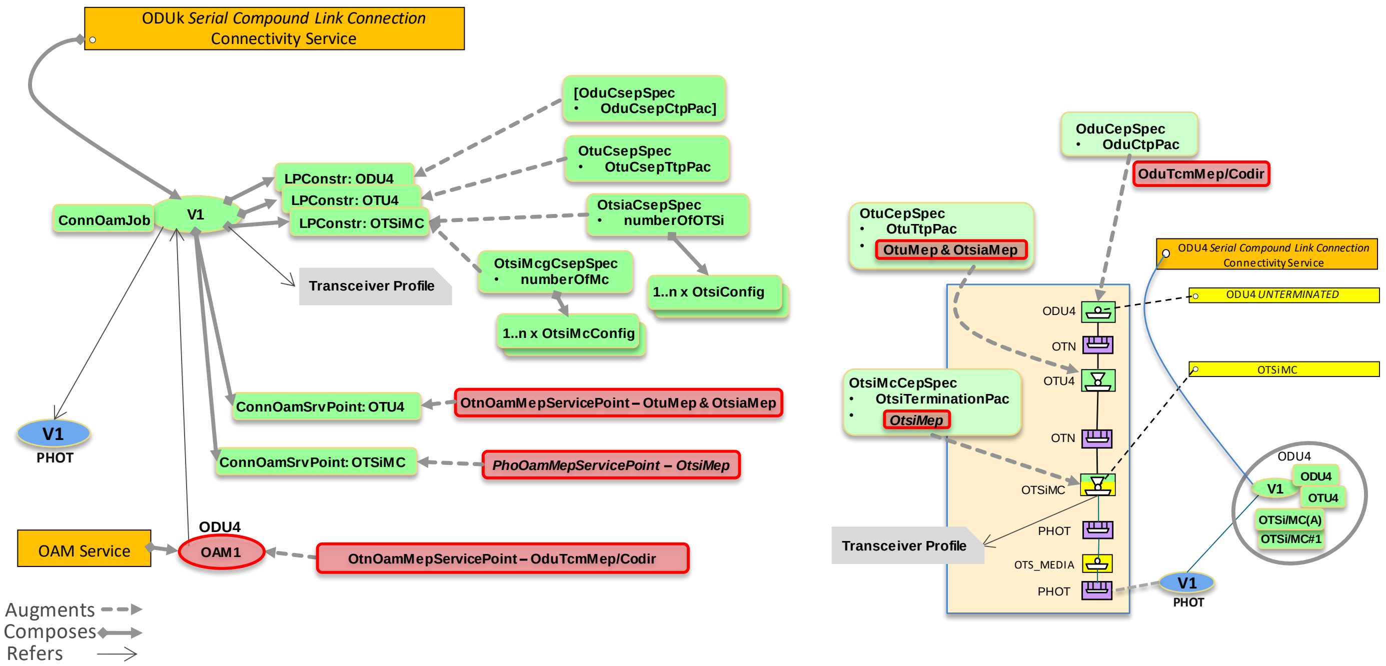


Figure 6-8 ODUk Serial Compound Link Connection Connectivity Service

Figure 6-9 shows the configuration parameters for the provisioning of the DSR connectivity service on an existing *transponder-to-transponder* connectivity service (ODUk *Serial Compound Link Connection* CS).

This scenario considers the DSR rate equal to the ODUk *Serial Compound Link Connection* rate, in other words the DSR payload is transported directly by a ODUk Infrastructure Trail.

The result includes the DSR connection plus the ODUk *terminated* connection.

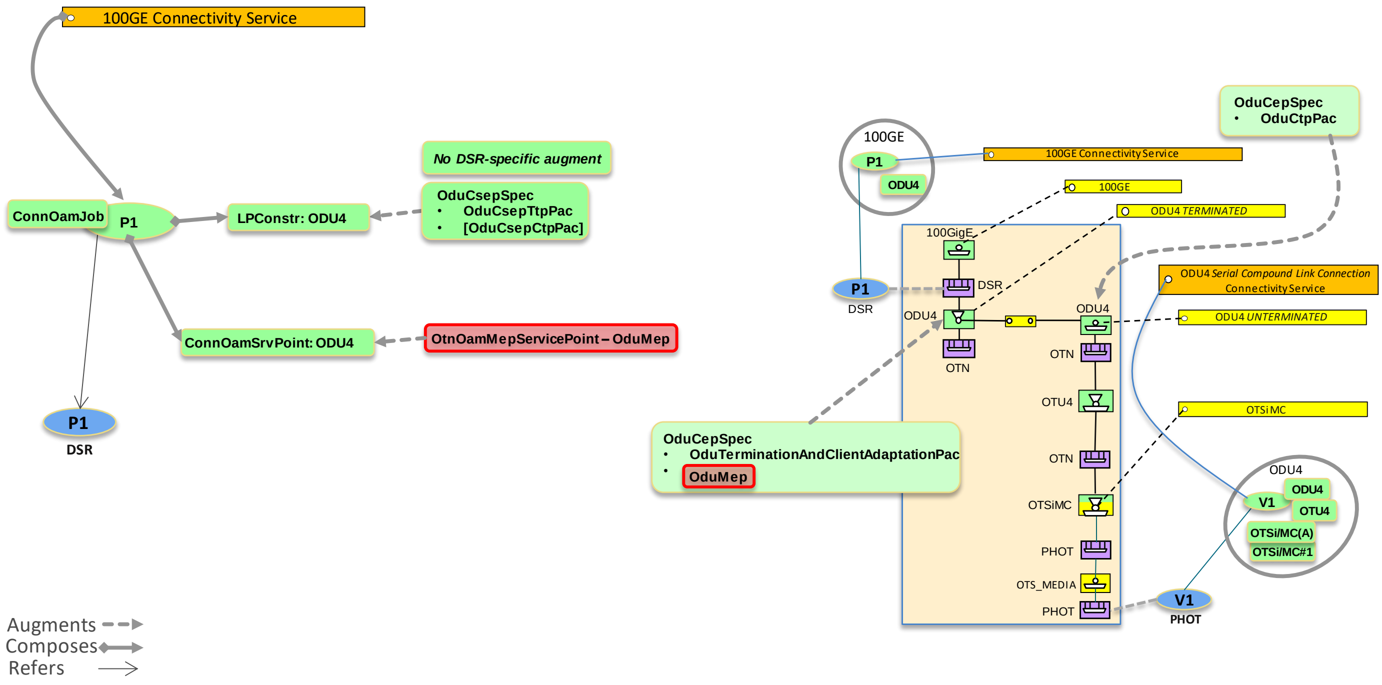


Figure 6-9 DSR/ODUk Connectivity Service on ODUk SCLC CS

Figure 6-10 shows the configuration parameters for the provisioning of the DSR connectivity service on an existing *transponder-to-transponder* connectivity service (ODUk *Serial Compound Link Connection* CS).

This scenario foresees OTN multiplexing, i.e., the DSR payload is transported by an ODU lower order container (ODUj) which is multiplexed into a higher order ODU container (ODUk Infrastructure Trail), which in turn is supported by the *transponder-to-transponder* connectivity.

It is assumed that the server ODUk Infrastructure Trail is either automatically created or reused if already existing. Note that ODUk parameters MAY be specified together with ODUj parameters to drive the creation of the server ODUk Infrastructure Trail.

The result includes the DSR connection plus the ODUj and ODUk *terminated* connections.

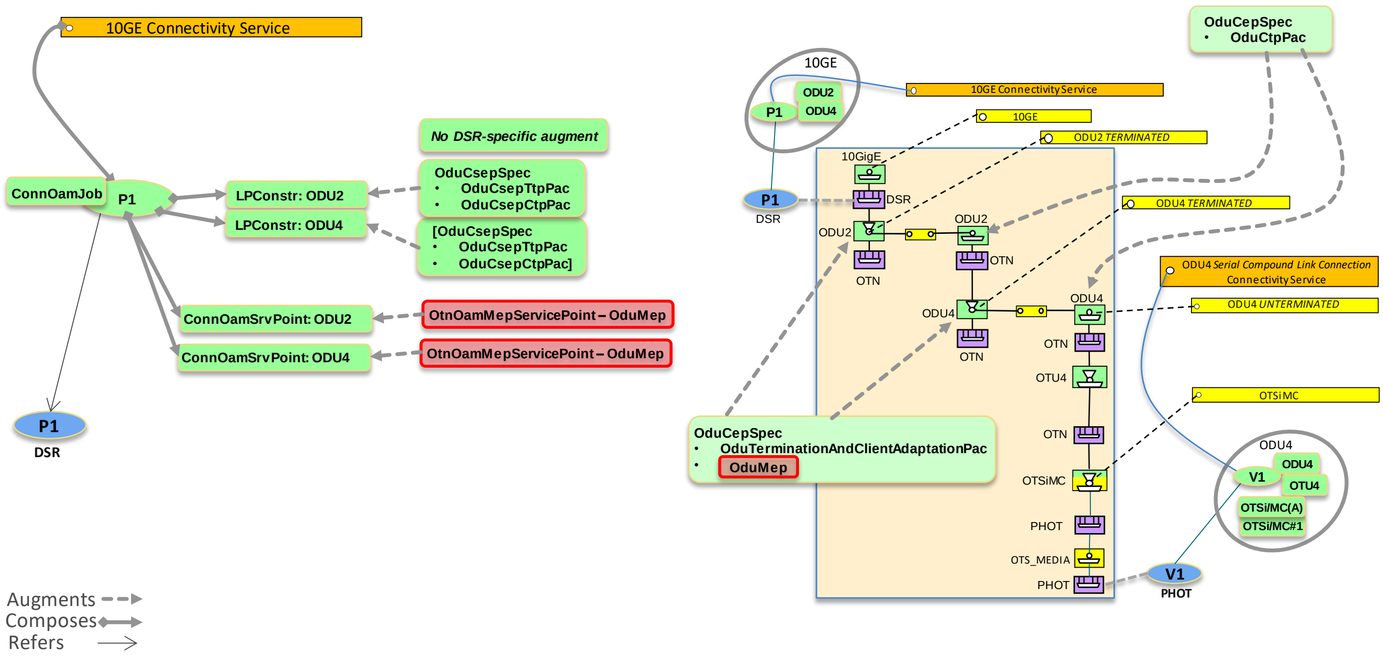


Figure 6-10 DSR/ODUj CS on ODUk SCLC CS, ODUk Terminated Connection automatically created or reused

Figure 6-11 shows a similar scenario with respect to Figure 6-10, with OTU3/ODU3 LPQ.

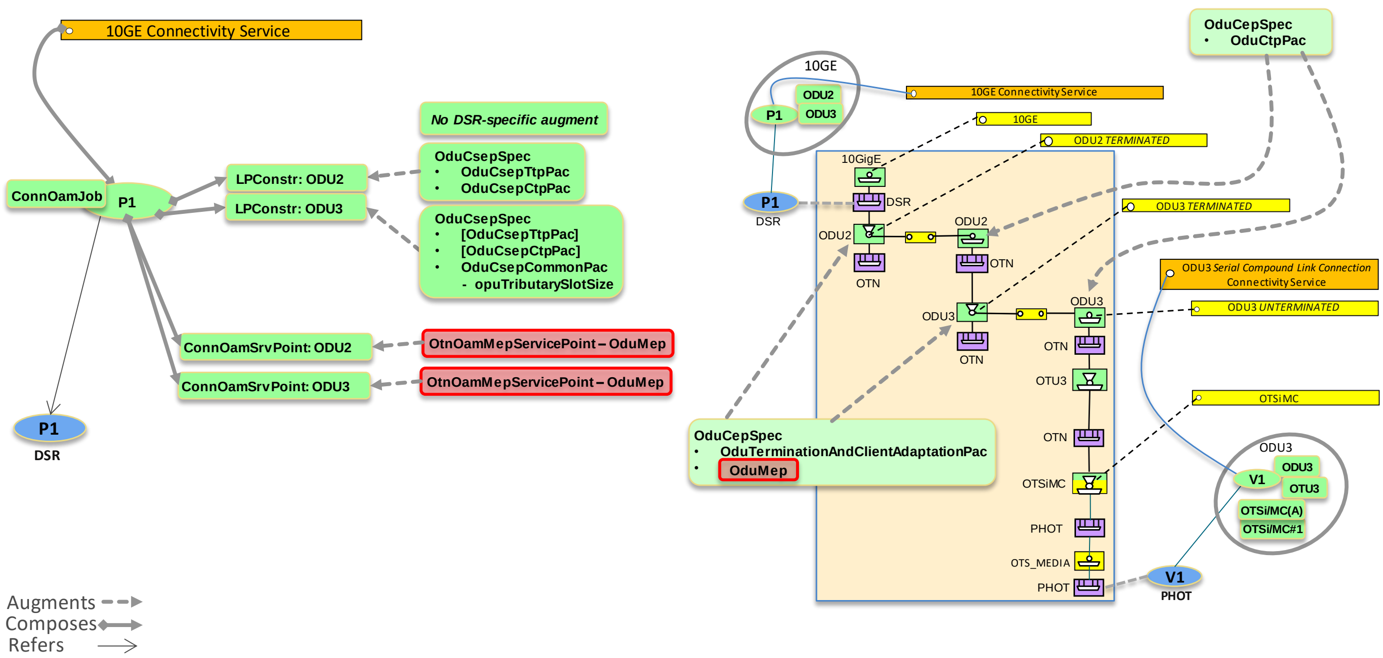


Figure 6-11 DSR/ODU2 CS on ODU3 SCLC CS, ODU3 Terminated Connection automatically created or reused

Figure 6-12 shows a similar scenario with respect to Figure 6-10, with no flexibility at ODU2 layer.

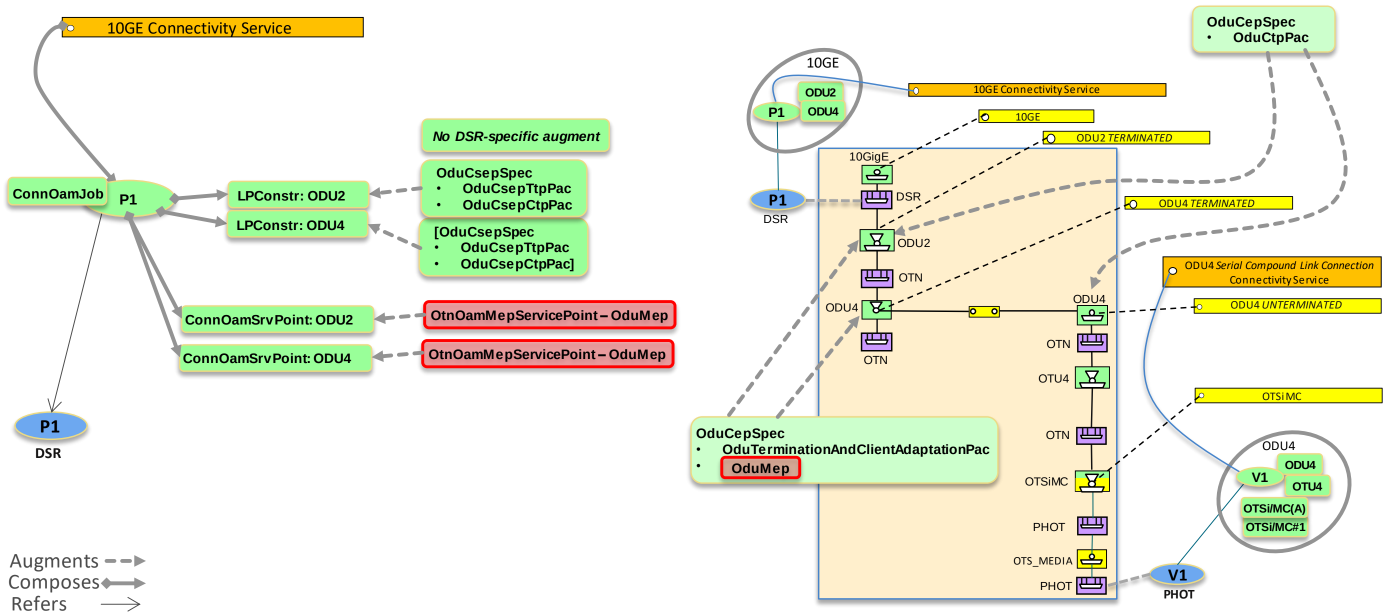


Figure 6-12 DSR/ODUj CS on ODUk SCLC CS, ODUk Term. Conn. autom. created or reused, no ODUj flexibility

Figure 6-13 shows a similar scenario with respect to Figure 6-10, with the server controller creating also the ODUk Infrastructure Trail *connectivity service*.

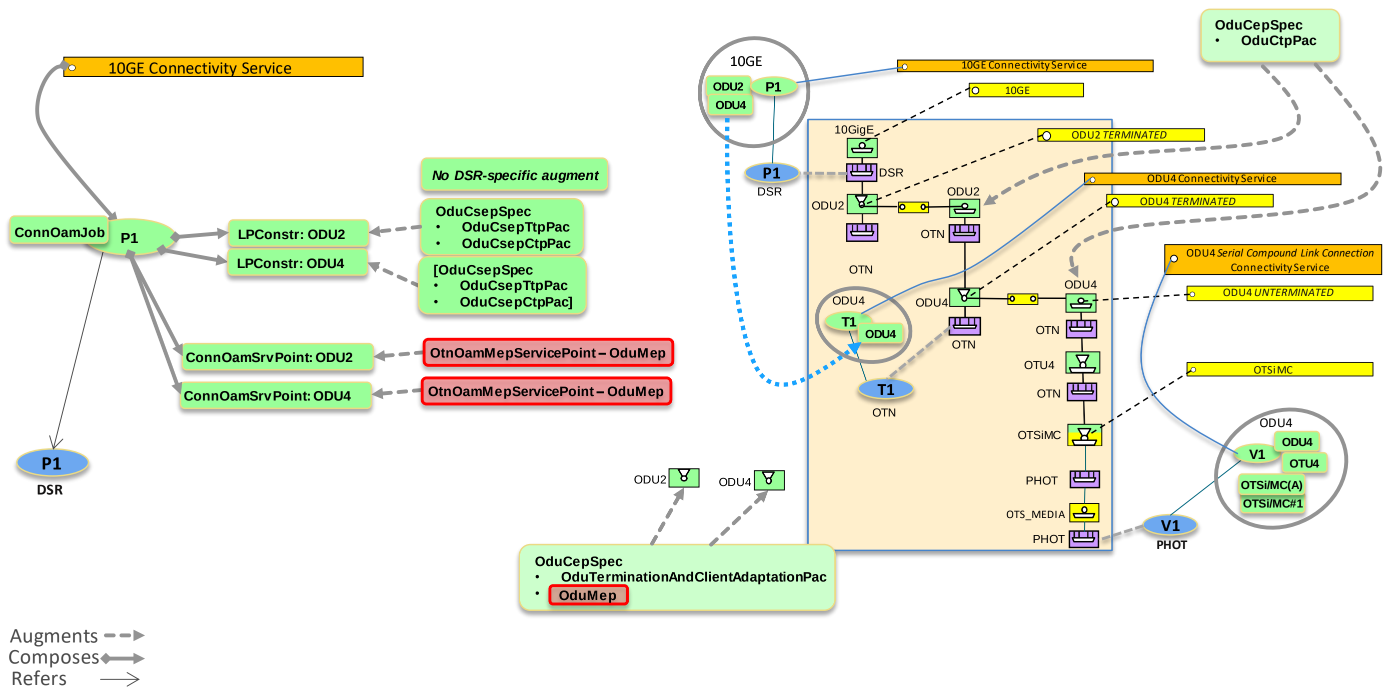


Figure 6-13 DSR/ODUj Connectivity Service on ODUk SCLC CS, auto creation of ODUk CS

Figure 6-14 shows the configuration parameters for the provisioning of the ODUk Infrastructure Trail connectivity service on an existing *transponder-to-transponder* connectivity service (ODUk *Serial Compound Link Connection* CS).

The result includes the ODUk *terminated* connection. Note that this scenario relies on the existence of a “floating” OTN NEP and associated SIP (T1), which is present in the topology, previous to the establishment of the terminated ODUk Infrastructure Trail CS, and which indicates the related capability.

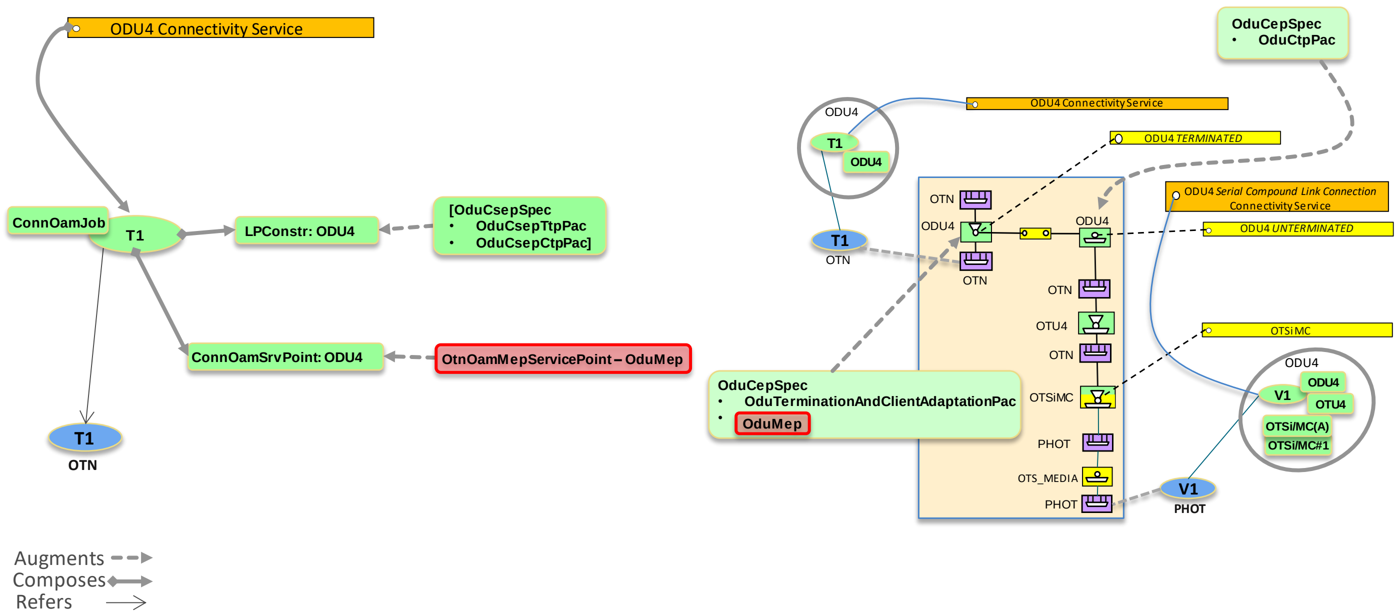


Figure 6-14 Infrastructure or Handoff ODUk Connectivity Service on ODUk SCLC CS

Figure 6-15 shows the configuration parameters for the provisioning of the DSR connectivity service on an existing ODUk Infrastructure Trail connectivity service. This builds on top of the Figure 6-14 and illustrates that it is only needed to specify the ODUj parameters.

The result includes the DSR connection plus the ODUj *terminated* connection.

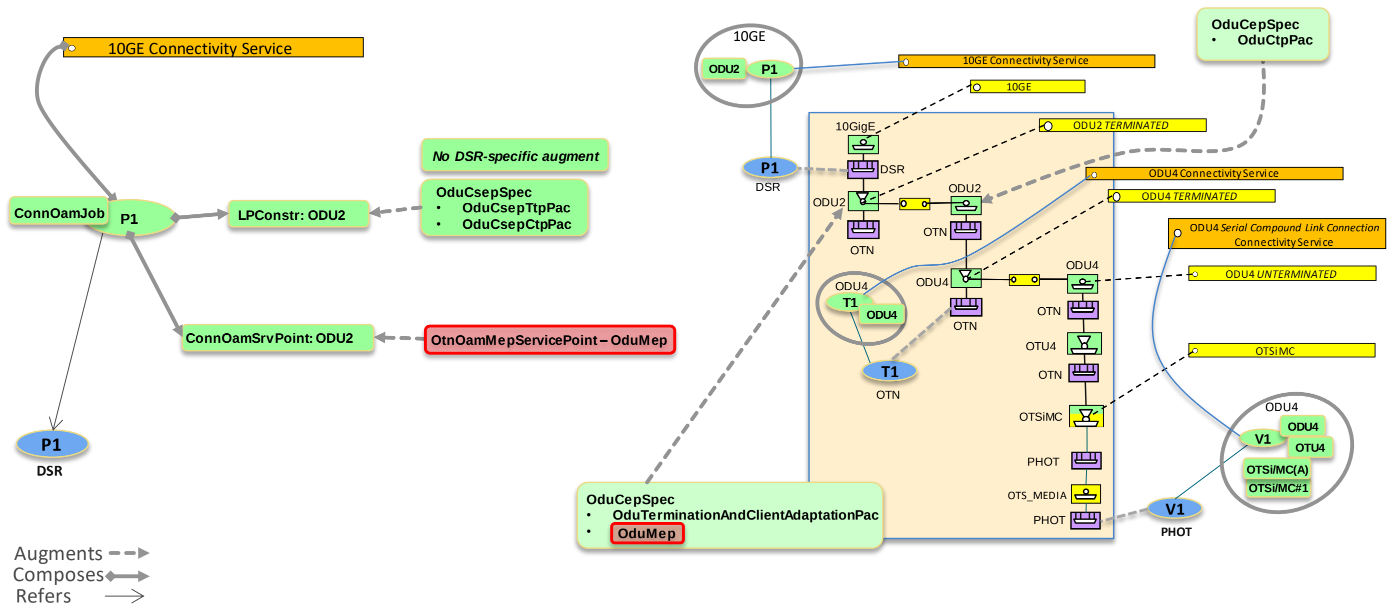


Figure 6-15 DSR/ODUj Connectivity Service on ODUk CS on ODUk SCLC CS

6.2.2.2 ODUk Serial Compound Link Connection CS – Transit Scenarios

In “transit scenarios” the Connectivity Service (and its CSEPs) could be specified at any client layer protocol name/qualifier (e.g., DSR or ODU2), as this is the *intent* specification. In other words, the CS represents the intent for a connection between SIPs, the CSEPs the intent for the amount and type of bandwidth on these SIPs. The only relationship between actual-local SIP/NEP capabilities and CS/CSEPs layer protocol name/qualifier is the known rule of technology stack (e.g., a 10G DSR can be potentially supported by an ODU4 container, the reverse case not). The server controller will allocate the appropriate resources at same and/or server layers.

By convention in this RIA, in transit scenarios the "unterminated" Top Connection(s) shall be represented only if there is at least one monitoring point in the transit managed domain (e.g., regeneration, Figure 5-16).

For the transit scenarios, please consider:

- If the unterminated top-level connection(s) are represented (such as a 10GE DSR or a ODU2 top-connection) they end at the outermost transit layer CEPs (e.g., ODU4 CEPs). Such ENNI CEPs are intended as the points in the topology where the Connection is received from/delivered to the external domain(s).

6.2.2.3 ODUk Serial Compound Link Connection CS – Asymmetric Scenarios

For the asymmetric scenarios, please consider:

- In asymmetric scenarios, the semi-terminated top-level connection(s) (such as the 10GE DSR or the ODU2 top-connection in the Figure 6-16) end at the outermost server layer CEP (e.g., the ODU4 CEP in the Figure 6-16). Such ENNI CEP is intended as the point in the topology where the Connection is delivered to the external domain. The DSR and ODU2 top-connections "*will continue*" in the next domain.

Figure 6-16 shows the DSR Asymmetric connectivity service where the handoff at ENNI is modeled by a (edge) node with only high order ODUk switching, while the lower order ODU switching is represented by other internal nodes (asymmetric scenario 1).

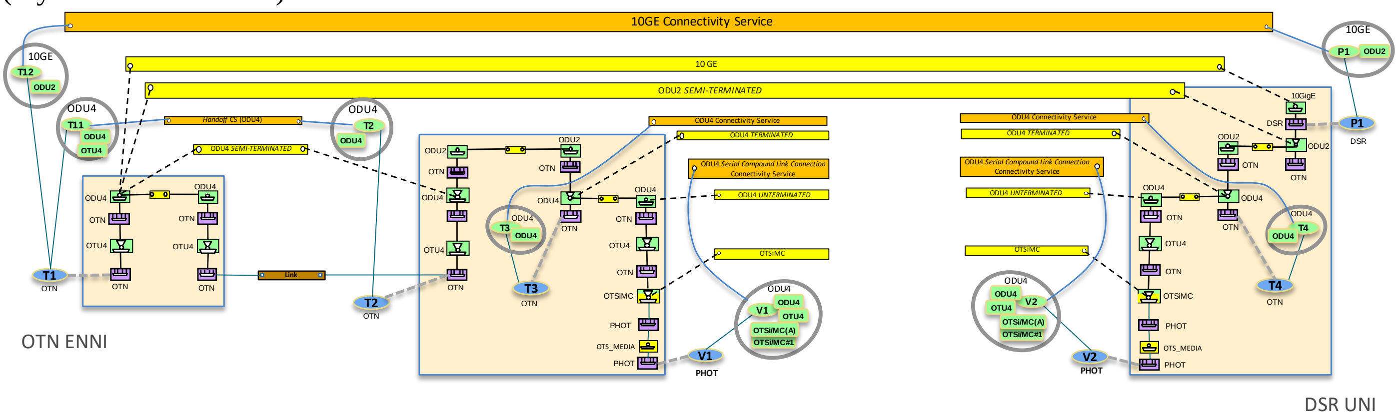


Figure 6-16 Asymmetric Scenario 1: Handoff at ODU4 Layer, no ODU2 layer on ENNI

Figure 6-17 shows a variation of scenario 1:

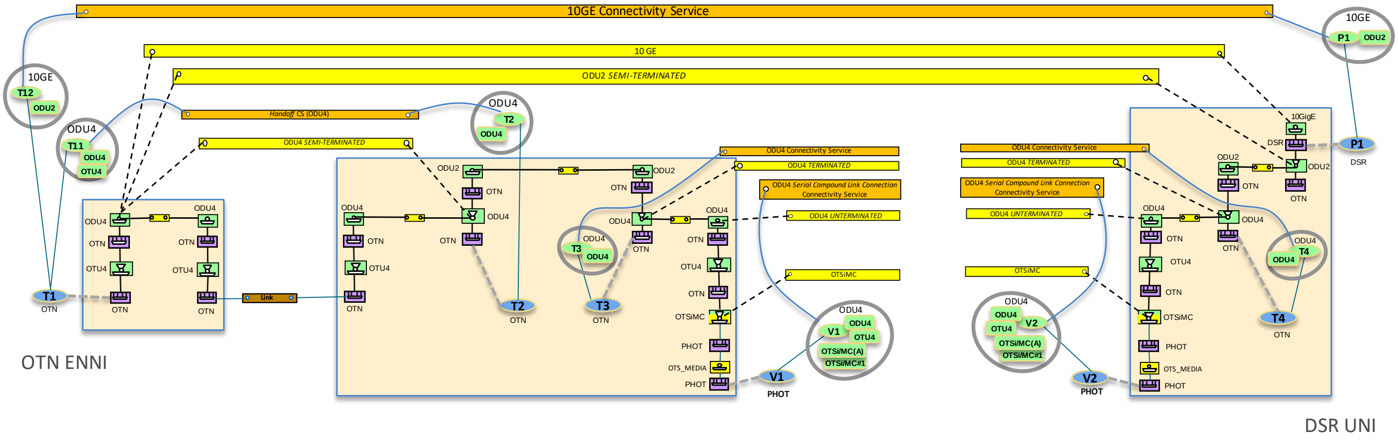


Figure 6-17 Asymmetric Scenario 1: Handoff at ODU4 Layer, ODU2 layer switching on Edge Node, *variation*

Figure 6-18 shows the DSR Asymmetric connectivity service where the handoff at ENNI is modeled by a (edge) node with both higher and lower order ODU switching, and with higher order ODU handoff at ENNI NEP (asymmetric scenario 2). Note the presence of the ODU4 semi-terminated top-level connection (despite it only spans a single node). In other scenarios such connection may span multiple nodes (e.g., in Figure 6-17).

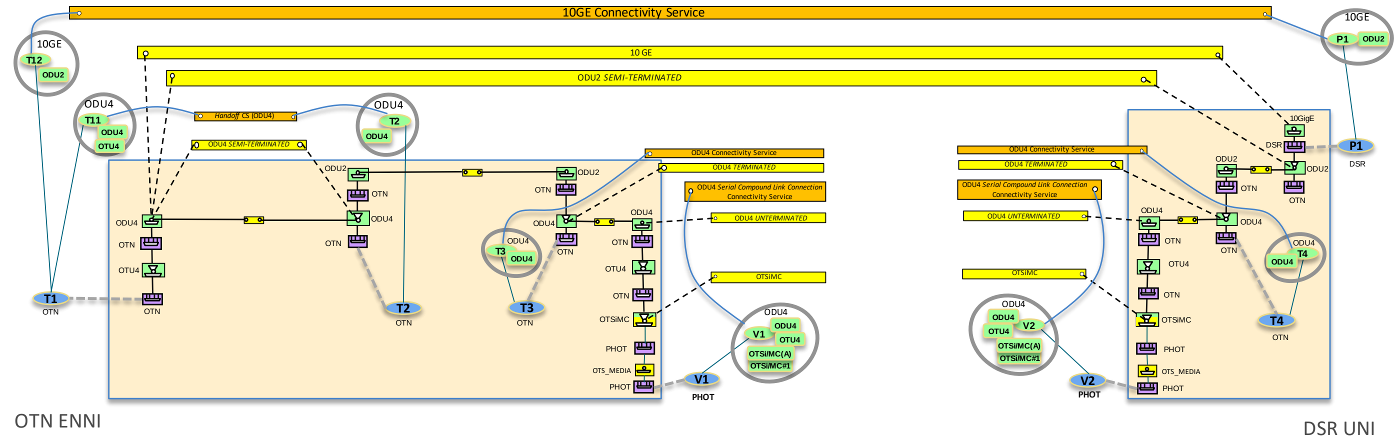


Figure 6-18 Asymmetric Scenario 2: Handoff at ODU4 Layer, ODU2 layer switching on Edge Node

Figure 6-19 shows the DSR Asymmetric connectivity service where the handoff at ENNI is modeled by a (edge) node with lower order ODU handoff at ENNI NEP (asymmetric scenario 3).

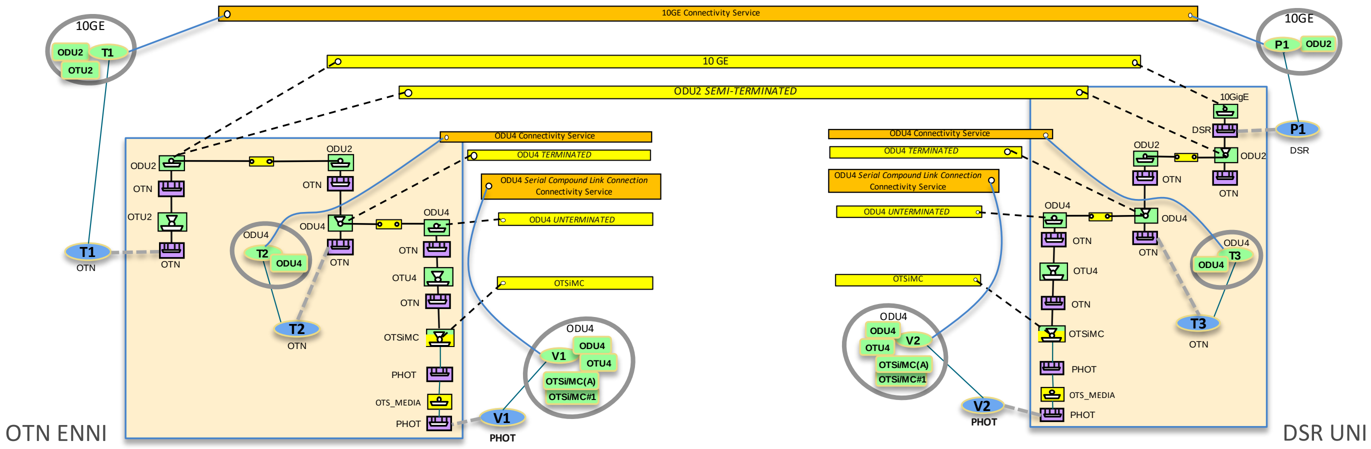


Figure 6-19 Asymmetric Scenario 3: Handoff at ODU2 Layer

Figure 6-20 shows the DSR Asymmetric connectivity service where the handoff at ENNI is modeled by a (edge) node with both higher and lower order ODU switching, with both higher and lower order ODU handoffs at ENNI NEP (asymmetric scenario 4).

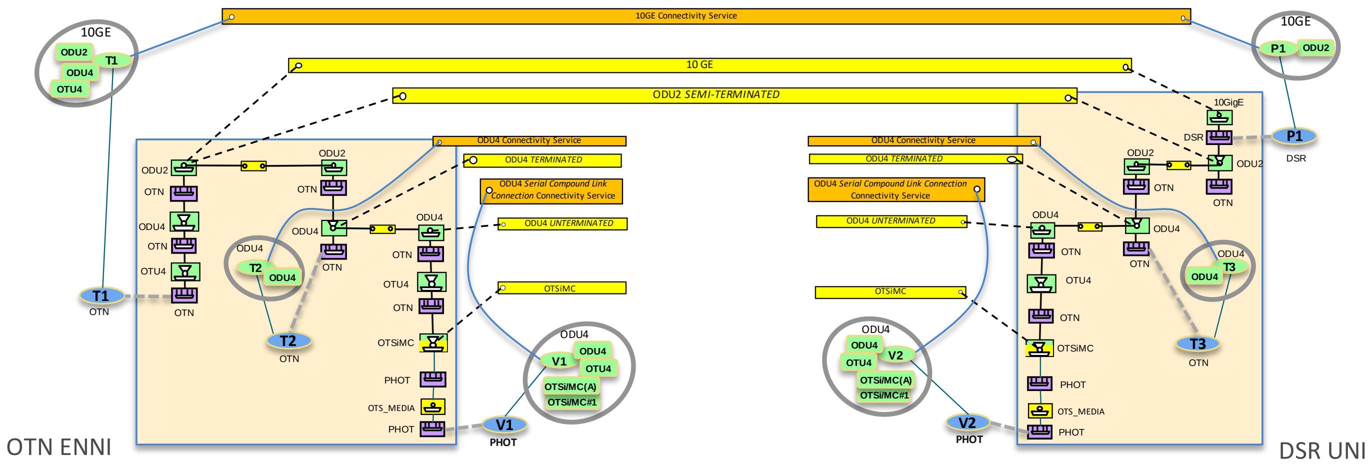


Figure 6-20 Asymmetric Scenario 4: Handoff at ODU4 Layer, ODU2 layer on ENNI

In the following, we introduce the provisioning aspects and configuration parameters for different asymmetric scenarios presented above.

Figure 6-21 shows the configuration parameters for the provisioning of the ENNI CSEP of the ODUk Handoff/Semi-terminated Trail connectivity service in the *asymmetric scenario 1*.

The result includes the ODUk *semi-terminated* connection.

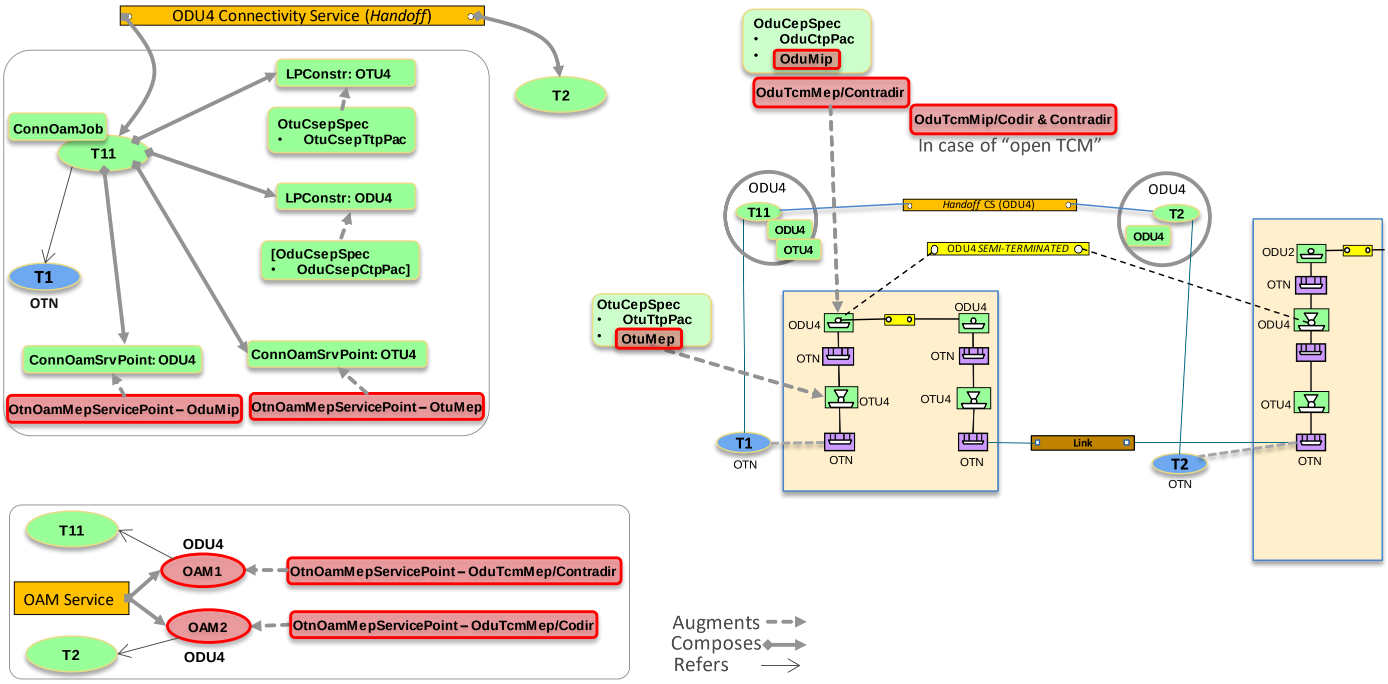


Figure 6-21 Asymmetric scenario 1: ODUk Handoff CS (OTN ENNI) – Part 1

Figure 6-22 shows the configuration parameters for the provisioning of the INNI CSEP of the ODUk Handoff/Semi-terminated Trail connectivity service in the asymmetric scenario 1.

The result includes the ODUk *semi-terminated* connection.

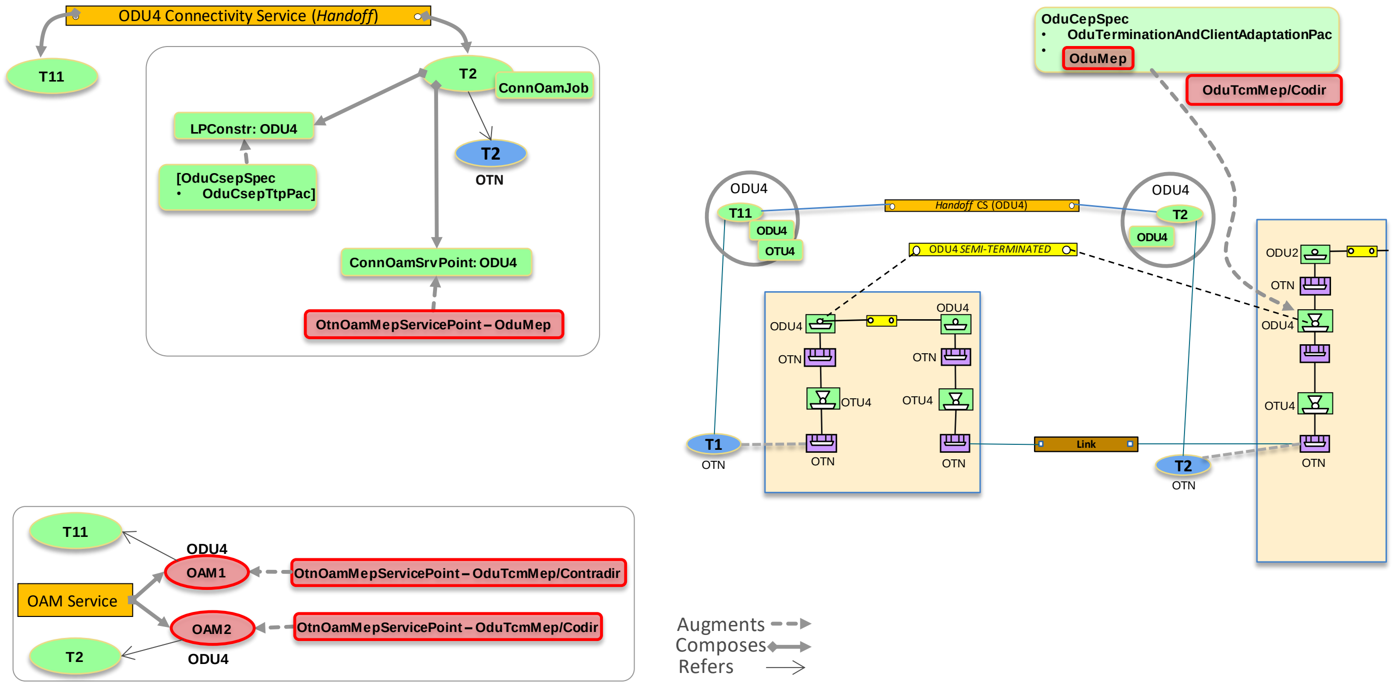


Figure 6-22 Asymmetric scenario 1: ODUk Handoff CS (OTN ENNI) – Part 2

Figure 6-23 shows the configuration parameters for the provisioning of the DSR Asymmetric connectivity service on an existing ODUk Handoff/Semi-terminated Trail connectivity service (hence ENNI side) in the asymmetric scenario 1.

The result includes the DSR connection plus the ODUj *semi-terminated* connection.

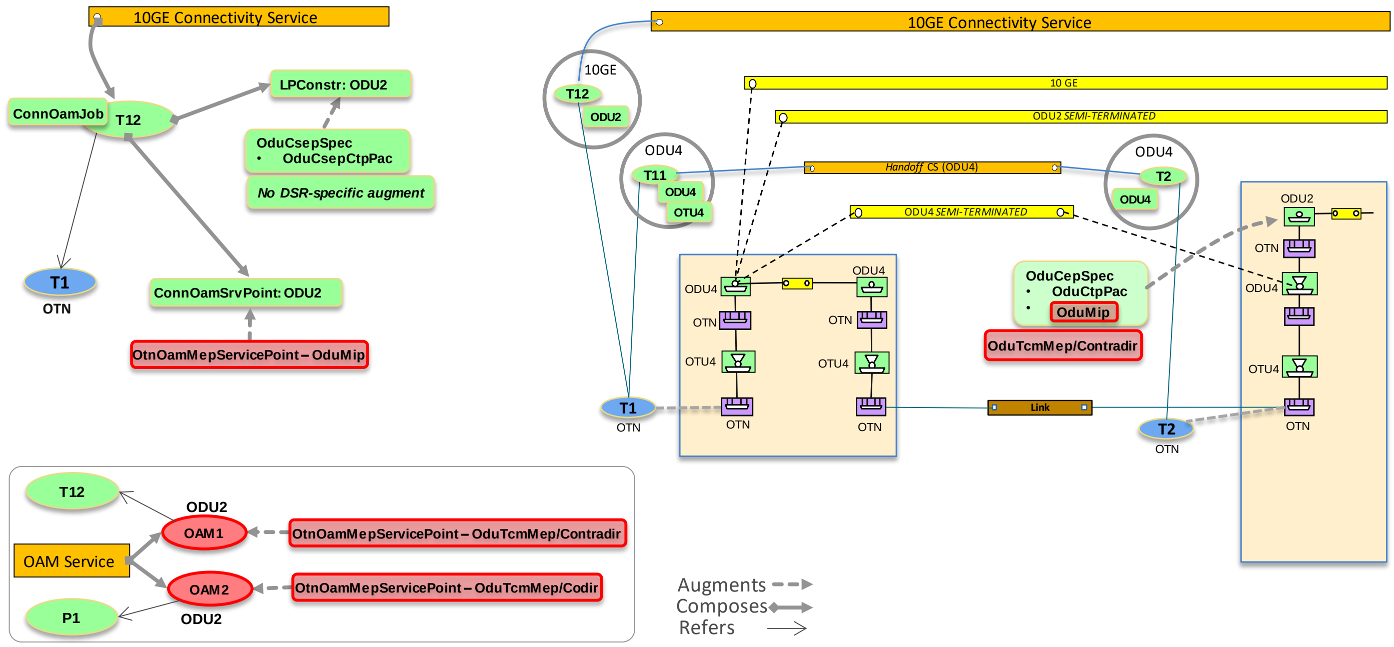


Figure 6-23 Asymmetric scenario 1: DSR/ODUj CS (OTN ENNI)

Figure 6-24 shows the configuration parameters for the provisioning of the DSR Asymmetric connectivity service on the UNI side in the asymmetric scenario 1. Note that, compared to Figure 6-15, tandem connection monitoring is added as it may be required in asymmetric scenarios.

The result includes the DSR connection plus the ODUj *semi-terminated* connection.

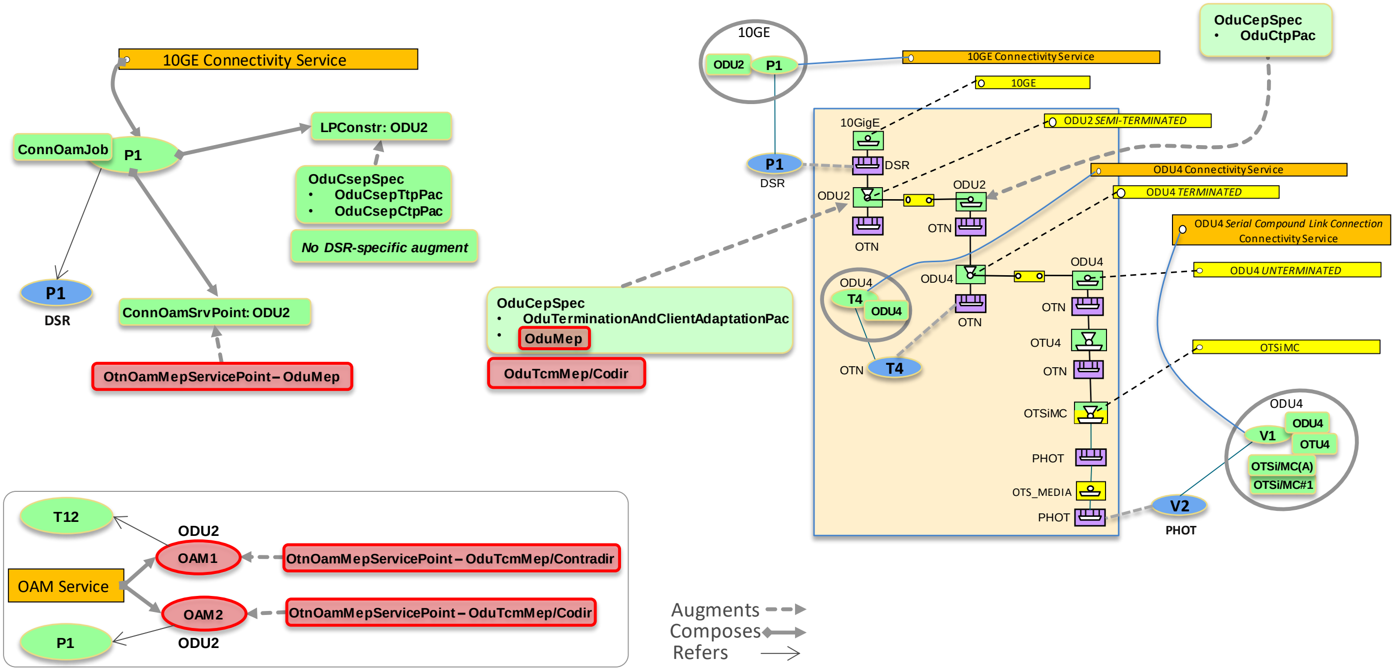


Figure 6-24 Asymmetric scenario 1: DSR/ODUj CS (DSR UNI)

Figure 6-25 shows the configuration parameters for the provisioning of the ENNI CSEP of the ODUk Handoff/Semi-terminated Trail connectivity service in the asymmetric scenario 2.

The result includes the ODUk *semi-terminated* connection.

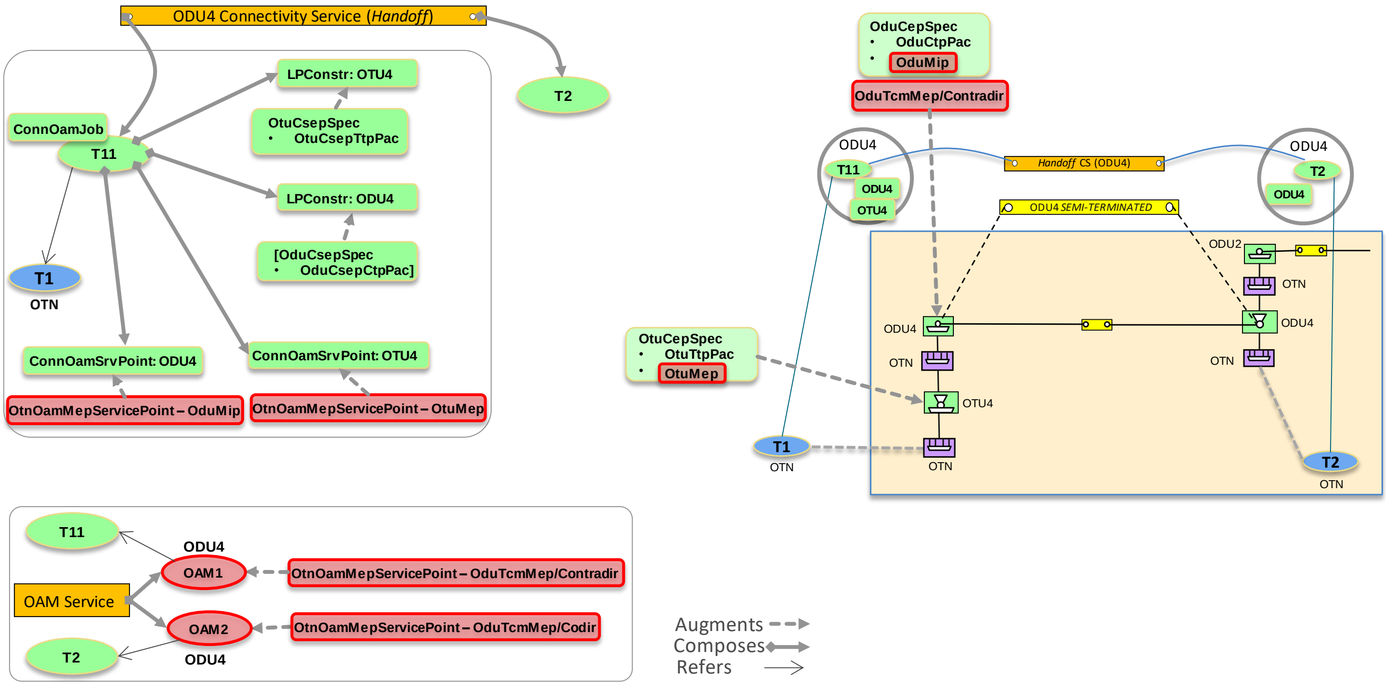


Figure 6-25 Asymmetric scenario 2: ODUk Handoff CS (OTN ENNI) - Part 1

Figure 6-26 shows the configuration parameters for the provisioning of the INNI CSEP of the ODUk Handoff/Semi-terminated Trail connectivity service in the asymmetric scenario 2.

The result includes the ODUk *semi-terminated* connection.

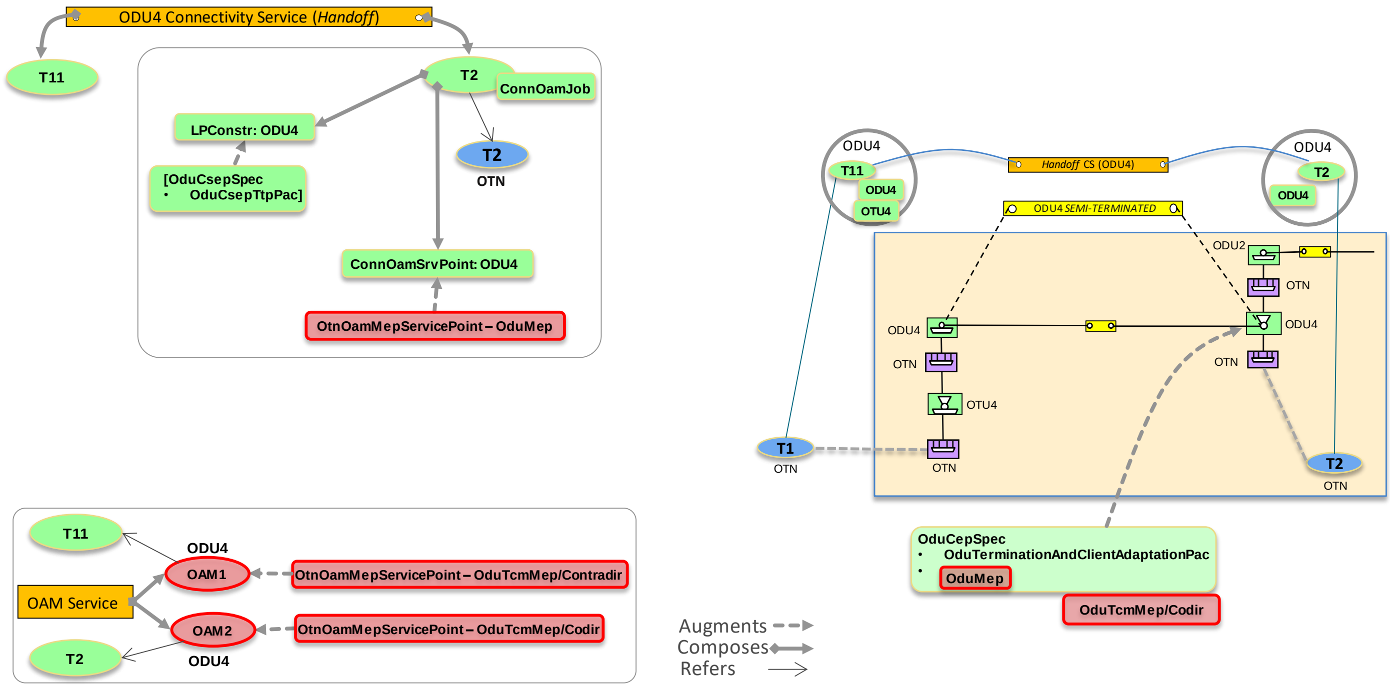


Figure 6-26 Asymmetric scenario 2: ODUk Handoff CS (OTN ENNI) - Part 2

Figure 6-27 shows the configuration parameters for the provisioning of the DSR Asymmetric connectivity service on an existing ODUk Handoff/Semi-terminated Trail connectivity service (hence ENNI side) in the asymmetric scenario 2.

The result includes the DSR connection plus the ODUj *semi-terminated* connection.

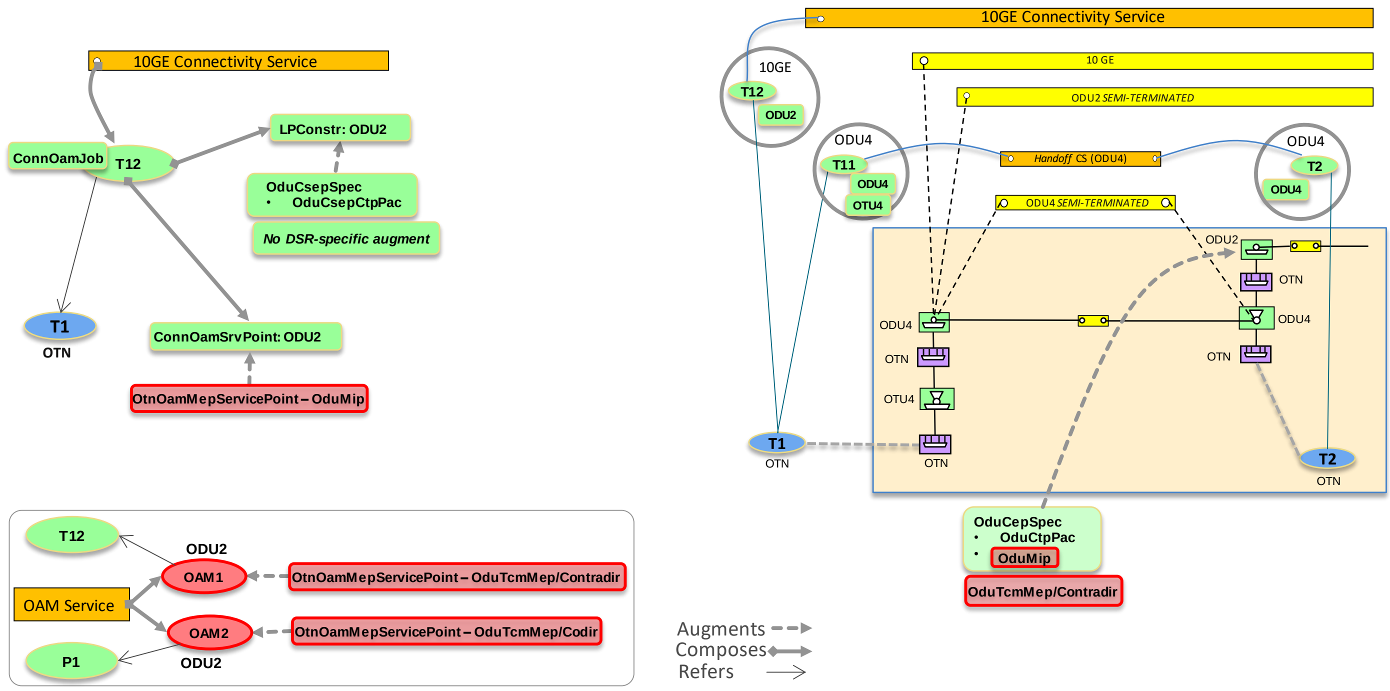


Figure 6-27 Asymmetric scenario 2: DSR/ODUj CS (OTN ENNI)

Figure 6-28 and Figure 6-29 shows the configuration parameters for the provisioning of the DSR Asymmetric connectivity service with the server controller creating also the ODUk Handoff/Semi-terminated Trail connectivity service (hence ENNI side) in the asymmetric scenario 2.

Note that ODUk parameters MAY be specified together with ODUj parameters to drive the creation of the server ODUk Handoff/Semi-terminated Trail.

Also note that the:

- TCM on ODUk TTP and
- TCM contra-directional on ODUk CTP

can be provisioned separately and the OduMep on ODUk TTP can be activated automatically by server controller.

The result includes the DSR connection plus the ODUj and ODUk *semi-terminated* connections.

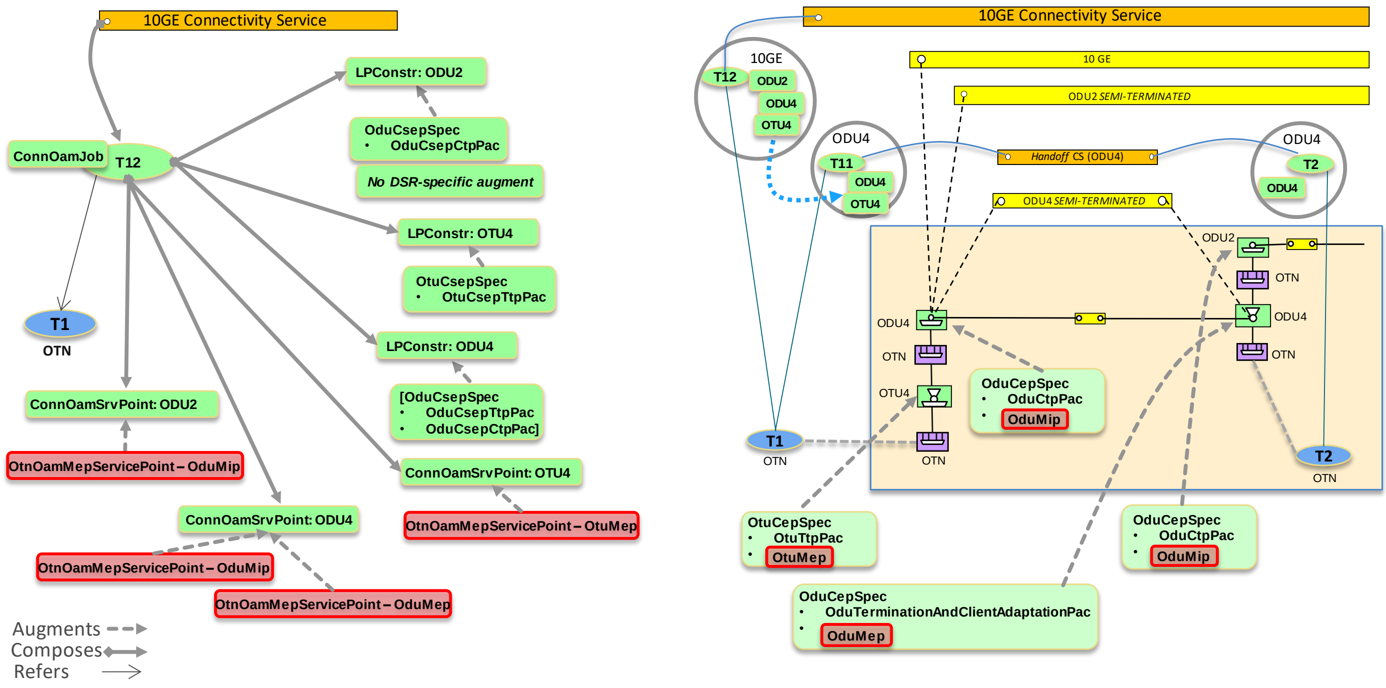


Figure 6-28 Asymmetric scenario 2: DSR/ODUj CS (OTN ENNI) - Auto creation of ODUk Handoff CS – Part 1

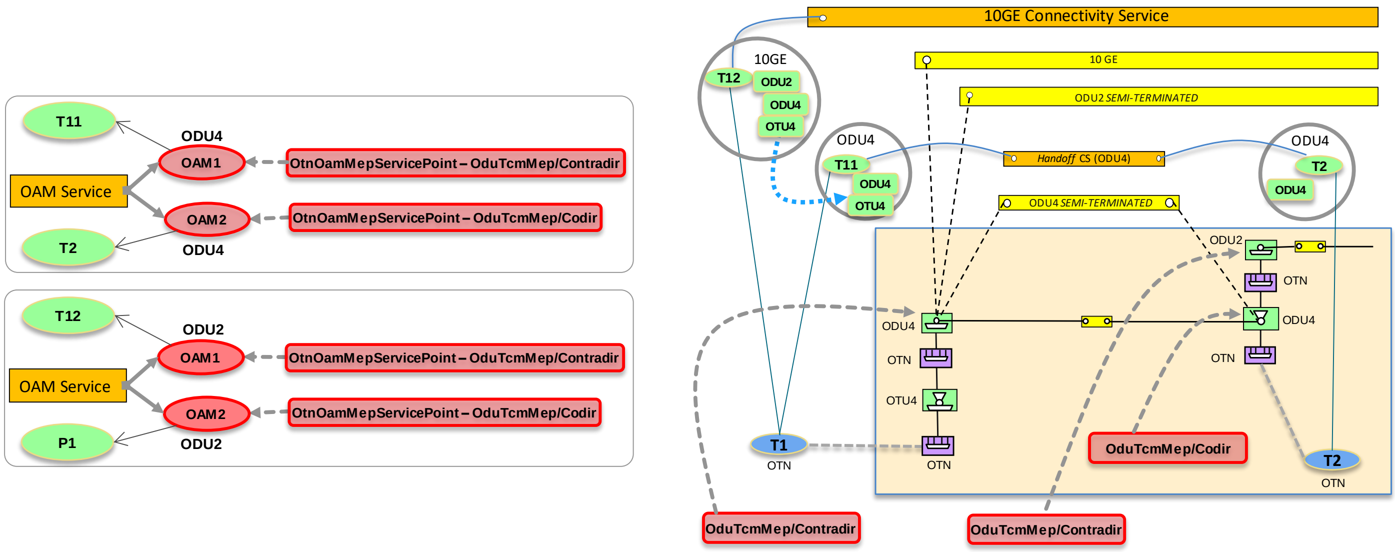


Figure 6-29 Asymmetric scenario 2: DSR/ODUj CS (OTN ENNI) - Auto creation of ODUk Handoff CS – Part 2

Figure 6-30 shows the configuration parameters for the provisioning of the DSR Asymmetric connectivity service in the asymmetric scenario 3.

The result includes the DSR connection plus the ODUj *semi-terminated* connection.

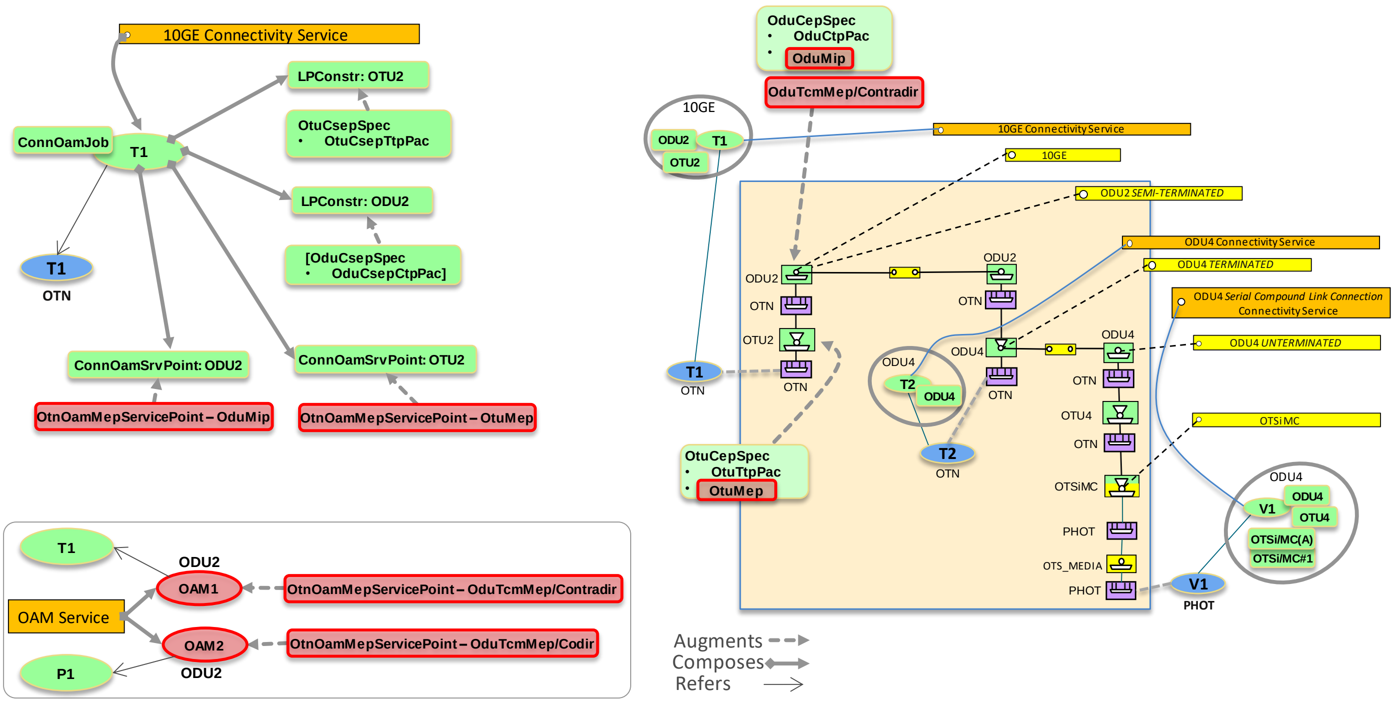


Figure 6-30 Asymmetric scenario 3: DSR/ODUj CS (OTN ENNI)

Figure 6-31 shows the configuration parameters for the provisioning of the DSR Asymmetric connectivity service including the parameters of the server ODUk Handoff/Semi-terminated Trail (hence ENNI side) in the asymmetric scenario 4. This scenario is similar to the one of Figure 6-28 but the ODUk Handoff/Semi-terminated Trail CS and connection is not represented at the management interface.

The result includes the DSR connection plus the ODUj *semi-terminated* connection.

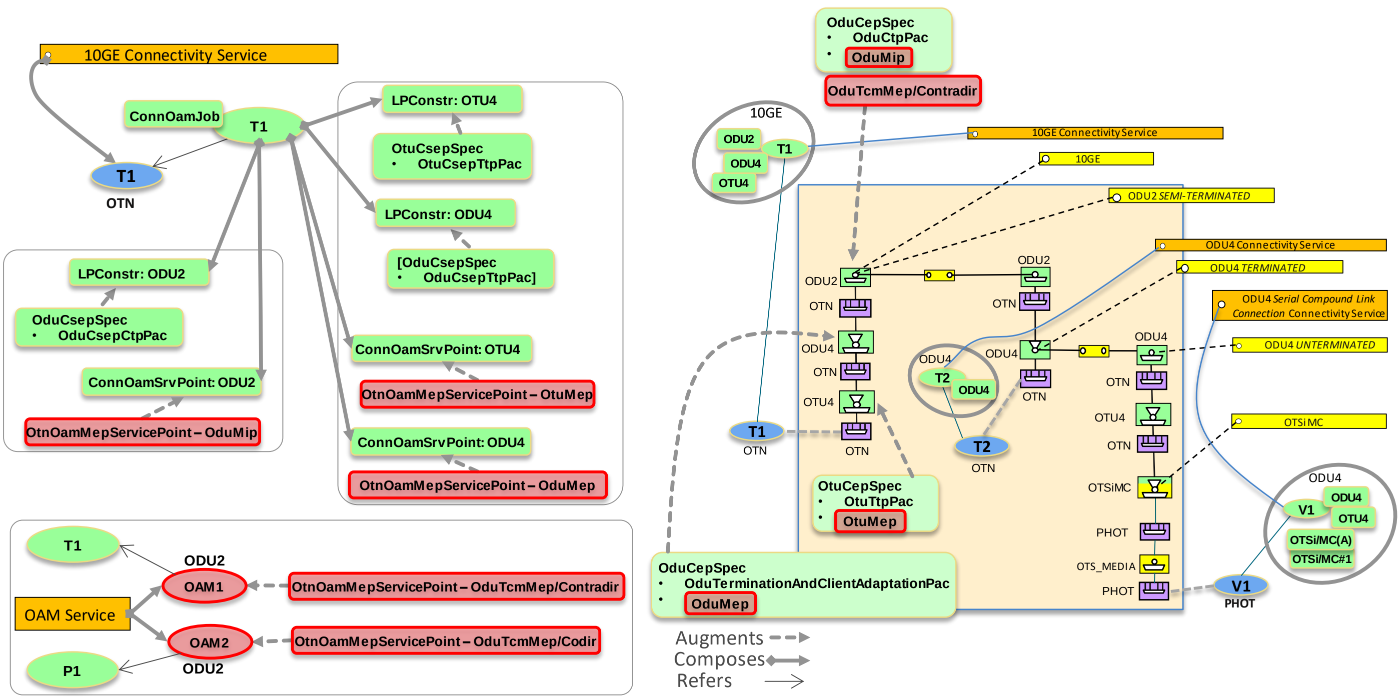


Figure 6-31 Asymmetric scenario 4: DSR/ODUj CS (OTN ENNI)

Figure 6-32 scenario is similar to the one of Figure 6-28 but applied to asymmetric scenario 4.

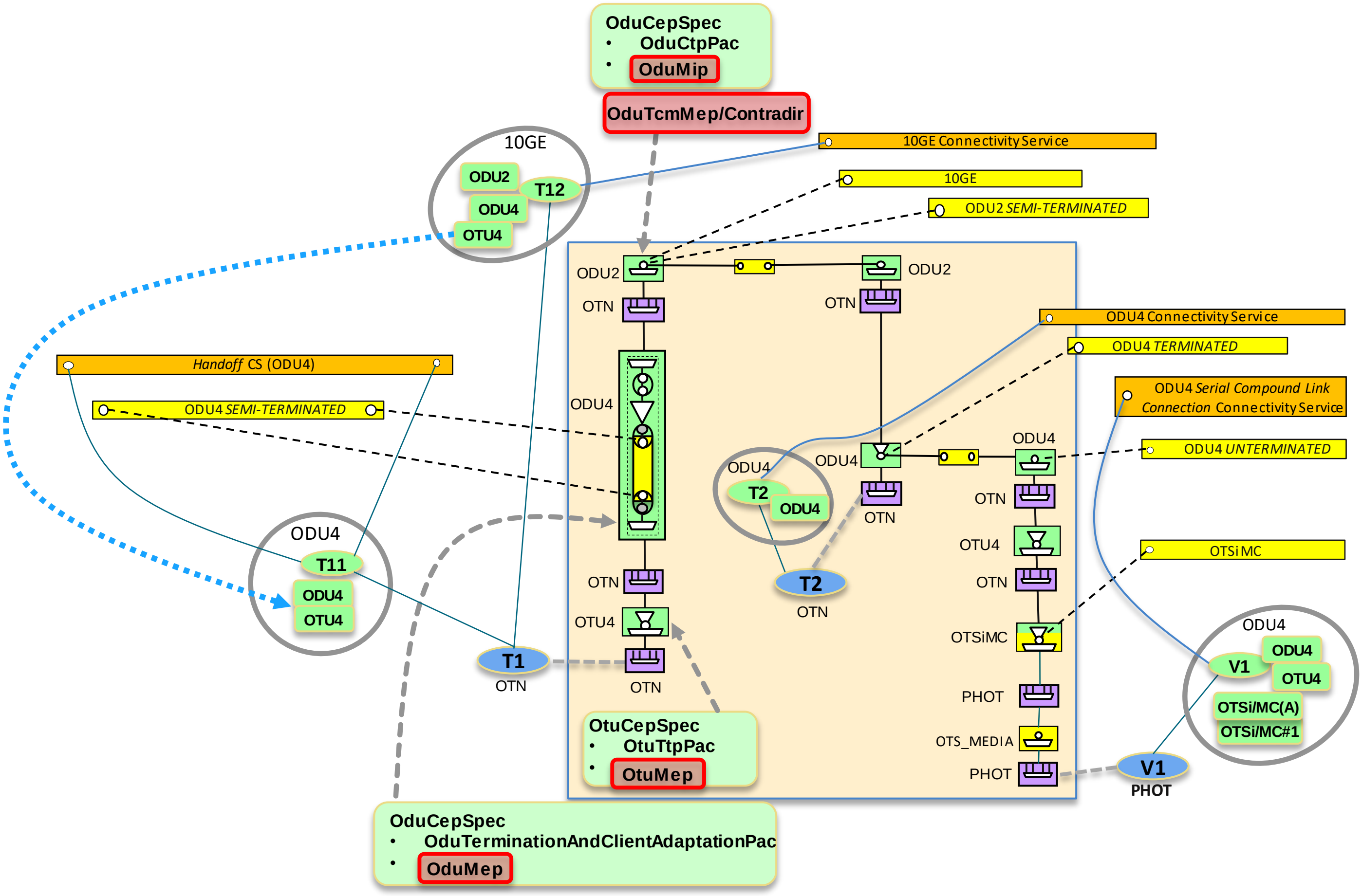


Figure 6-32 Asymmetric scenario 4: DSR/ODUj CS (OTN ENNI) - Explicit ODU4 Handoff CS and Connection

6.2.2.4 ODUCn Trail Connectivity Service

Figure 6-33 shows the configuration parameters for the provisioning of the *transponder-to-transponder* connectivity based on ODUCn container, the ODUCn *Trail* Connectivity Service.

The result includes the OTSiMC connection(s) plus the ODUCn Connection.

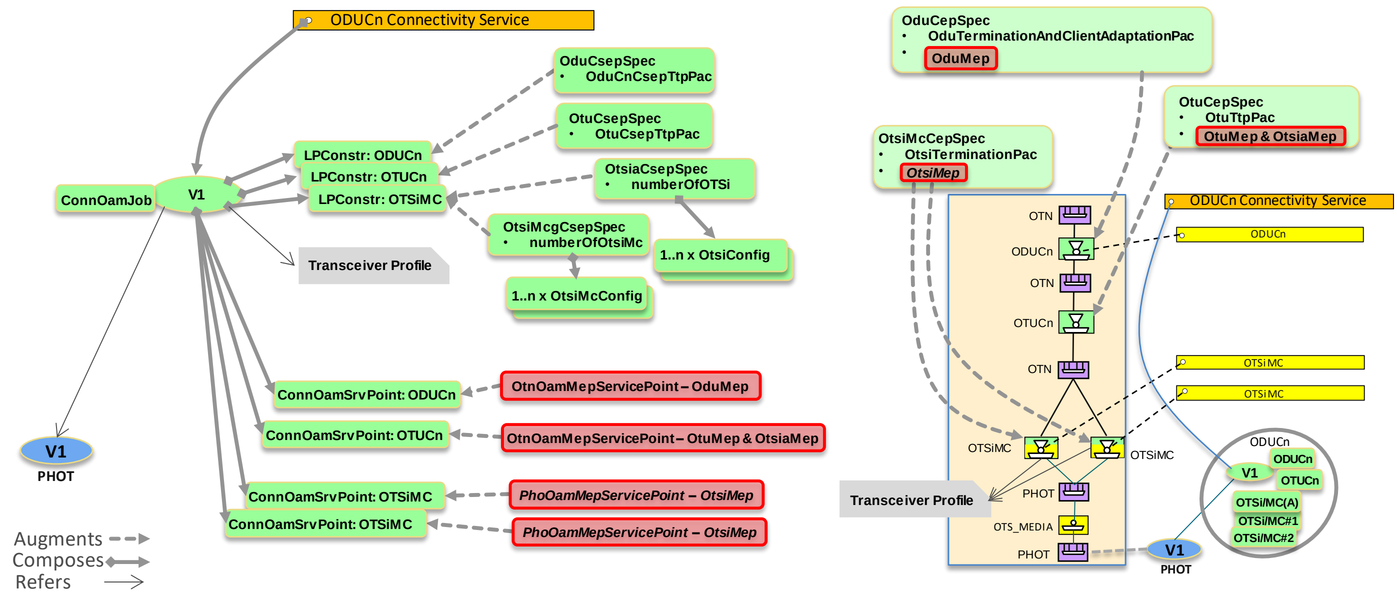


Figure 6-33 ODUCn Connectivity Service

Figure 6-34 shows the configuration parameters for the provisioning of the DSR connectivity service on an existing *transponder-to-transponder* connectivity service (ODUCn *Trail* CS).

In this scenario the DSR payload is transported by an ODU Flex container over the ODUCn Trail.

The result includes the DSR connection plus the ODU Flex *terminated* connection.

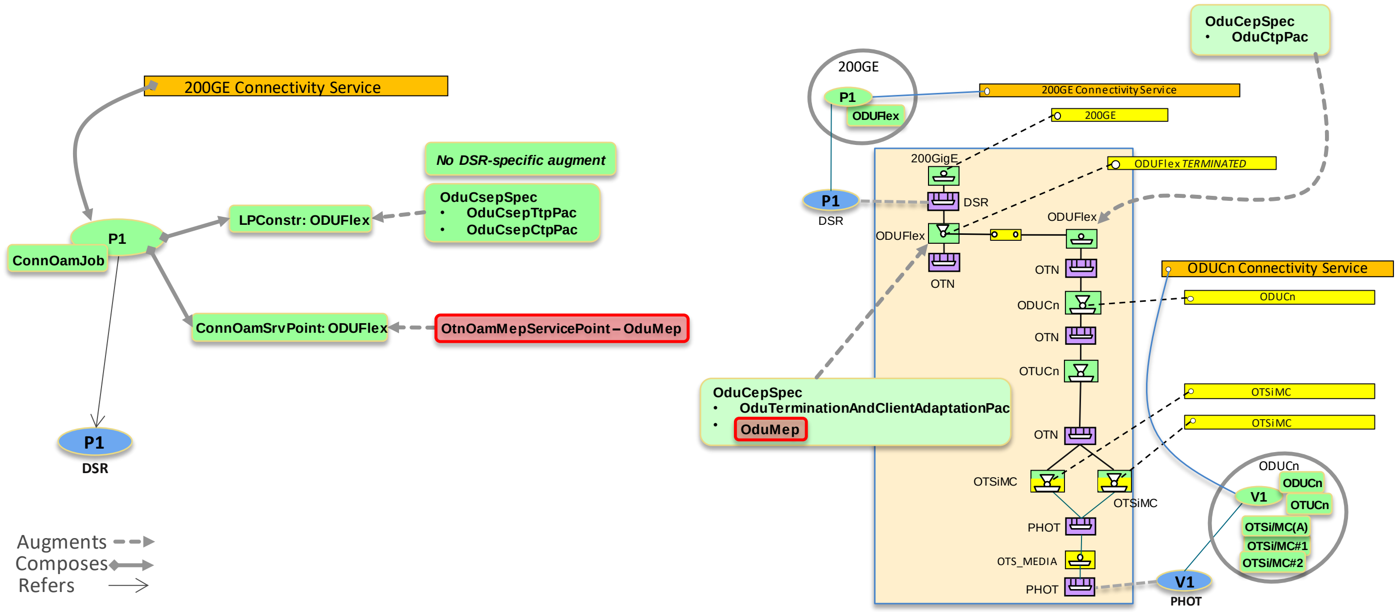


Figure 6-34 DSR/ODUFlex Connectivity Service on ODUCn CS

Figure 6-35 shows the configuration parameters for the provisioning of the DSR connectivity service on an existing *transponder-to-transponder* connectivity service (ODUCn *Trail* CS).

This scenario foresees OTN multiplexing, i.e. the DSR payload is transported by an ODU lower order container (ODUj) which is multiplexed into a higher order ODU container (ODUk Infrastructure Trail), which in turn is supported by the *transponder-to-transponder* connectivity.

It is assumed that the server ODUk Infrastructure Trail is either automatically created or reused if already existing. Note that ODUk parameters MAY be specified together with ODUj parameters to drive the creation of the server ODUk Infrastructure Trail.

The result includes the DSR connection plus the ODUj and ODUk *terminated* connections.

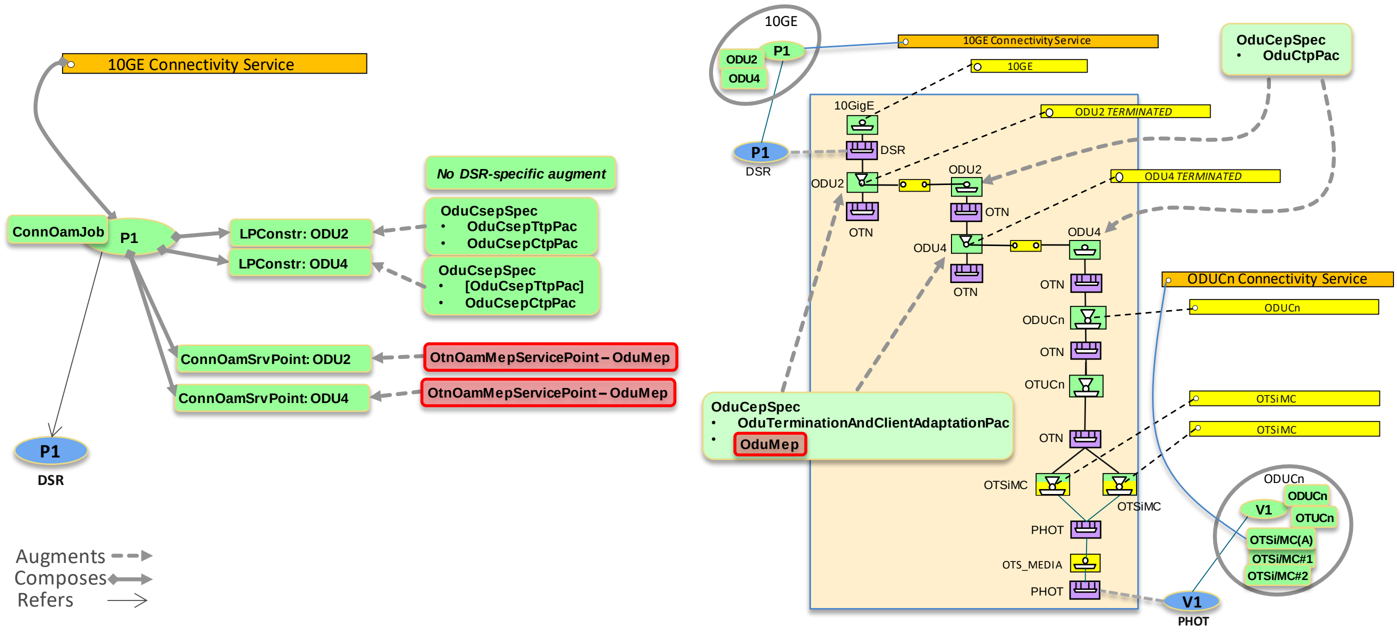


Figure 6-35 DSR/ODUj CS on ODUk on ODUCn CS - ODUk Terminated Connection automatically created or reused

Figure 6-36 shows a similar scenario with respect to Figure 6-35, with the server controller creating also the ODUk Infrastructure Trail *connectivity service*.

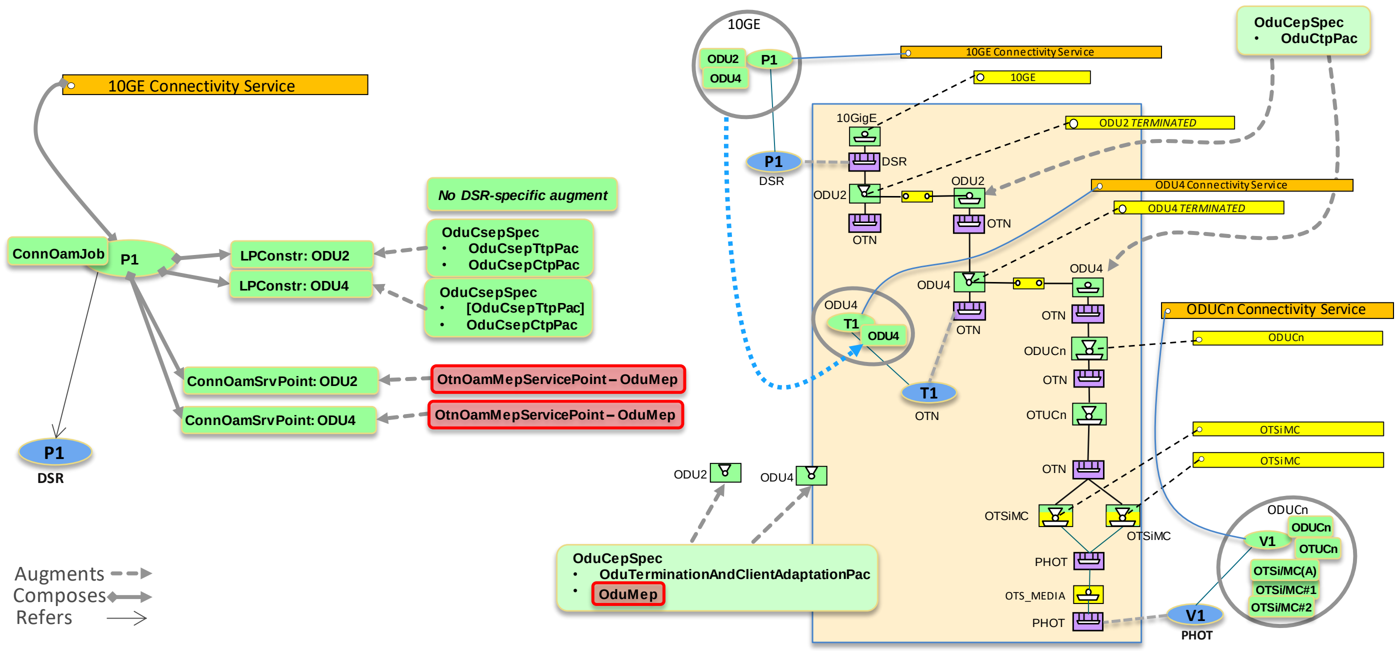


Figure 6-36 DSR/ODUj CS on ODUk CS on ODUCn CS - Auto creation of ODUk CS

Figure 6-37 shows the configuration parameters for the provisioning of the ODUk Infrastructure Trail connectivity service on an existing *transponder-to-transponder* connectivity service (ODUCn *Trail* CS).

The result includes the ODUk *terminated* connection. Note that this scenario relies on the existence of a “floating” OTN NEP and associated SIP (T1), which is present in the topology, previous to the establishment of the terminated ODUk Infrastructure Trail CS, and which indicates the related capability.

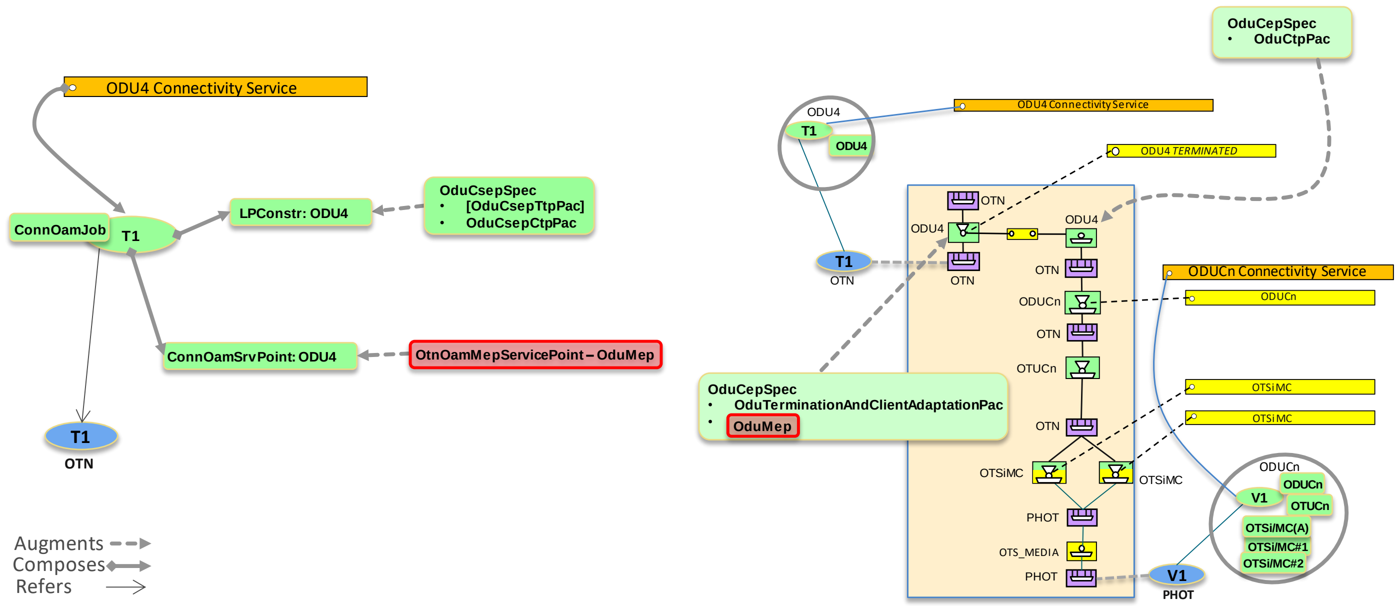


Figure 6-37 Infrastructure or Handoff ODUk CS on ODUCn CS

Figure 6-38 shows the configuration parameters for the provisioning of the DSR connectivity service on an existing ODUk Infrastructure Trail connectivity service. This builds on top of the Figure 6-37 and illustrates that it is only needed to specify the ODUj parameters.

The result includes the DSR connection plus the ODUj *terminated* connection.

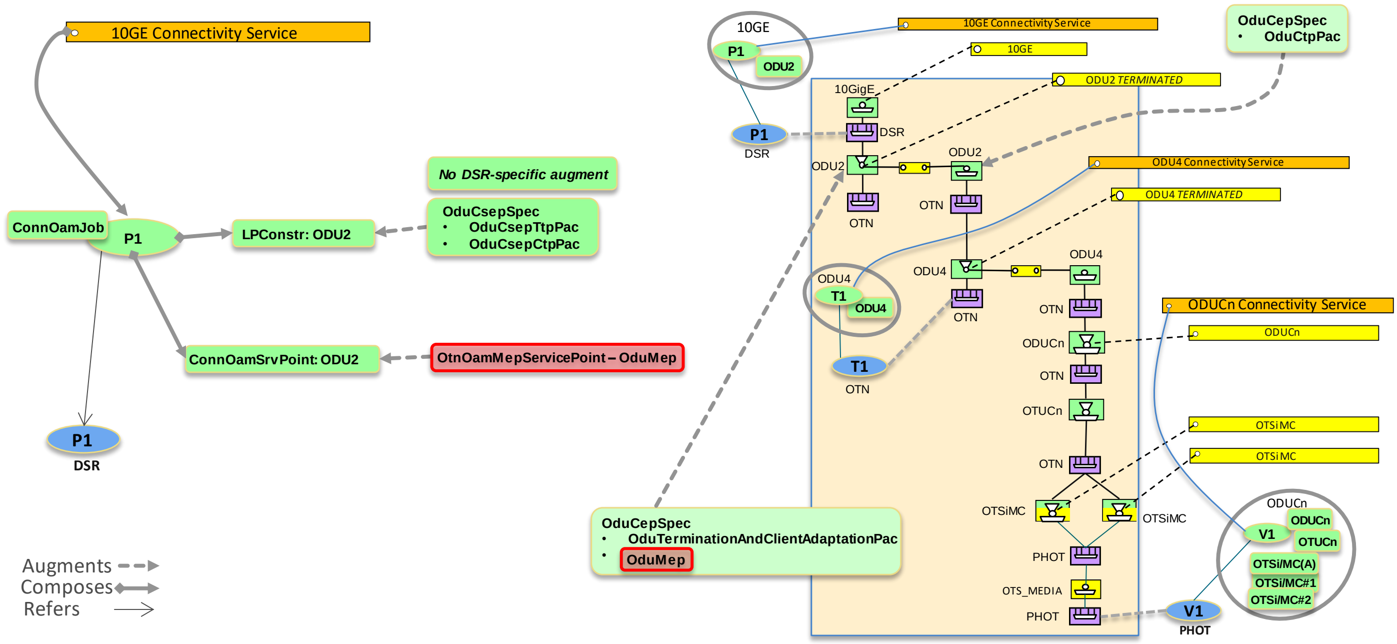


Figure 6-38 DSR/ODUj CS on ODUk CS on ODUCn CS

Regarding the asymmetric scenarios, see

- Figure 6-16 Asymmetric Scenario 1: Handoff at ODU4 Layer, no ODU2 layer on ENNI
- Figure 6-17 Asymmetric Scenario 1: Handoff at ODU4 Layer, ODU2 layer switching on Edge Node, *variation*
- Figure 6-18 Asymmetric Scenario 2: Handoff at ODU4 Layer, ODU2 layer switching on Edge Node
- Figure 6-19 Asymmetric Scenario 3: Handoff at ODU2 Layer
- Figure 6-20 Asymmetric Scenario 4: Handoff at ODU4 Layer, ODU2 layer on ENNI

replacing the ODUk *Serial Compound Link Connection* CS with the ODUCn *Trail* CS.

6.2.2.5 ODUk Trail Connectivity Service

Figure 6-39 shows the configuration parameters for the provisioning of the *transponder-to-transponder* connectivity based on a *terminated* ODUk container, the ODUk *Trail* Connectivity Service.

The result includes the OTSiMC connection plus the ODUk *terminated* Connection. OTUk connection is considered optional.

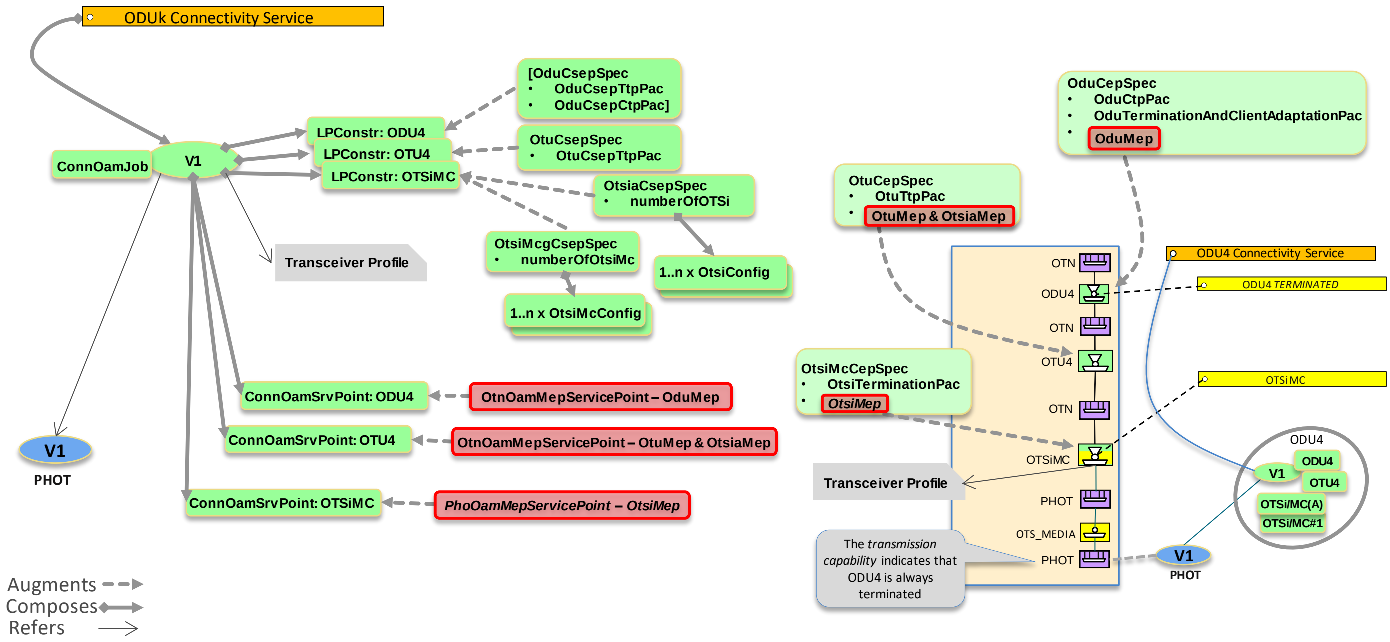


Figure 6-39 ODUk Trail Connectivity Service

Figure 6-40 shows the configuration parameters for the provisioning of the DSR connectivity service on an existing *transponder-to-transponder* connectivity service (ODUk *Trail* CS).

This scenario foresees OTN multiplexing, i.e., the DSR payload is transported by an ODU lower order container (ODUj) which is multiplexed into the *transponder-to-transponder* ODUk *Trail*.

The result includes the DSR connection plus the ODUj *terminated* connection.

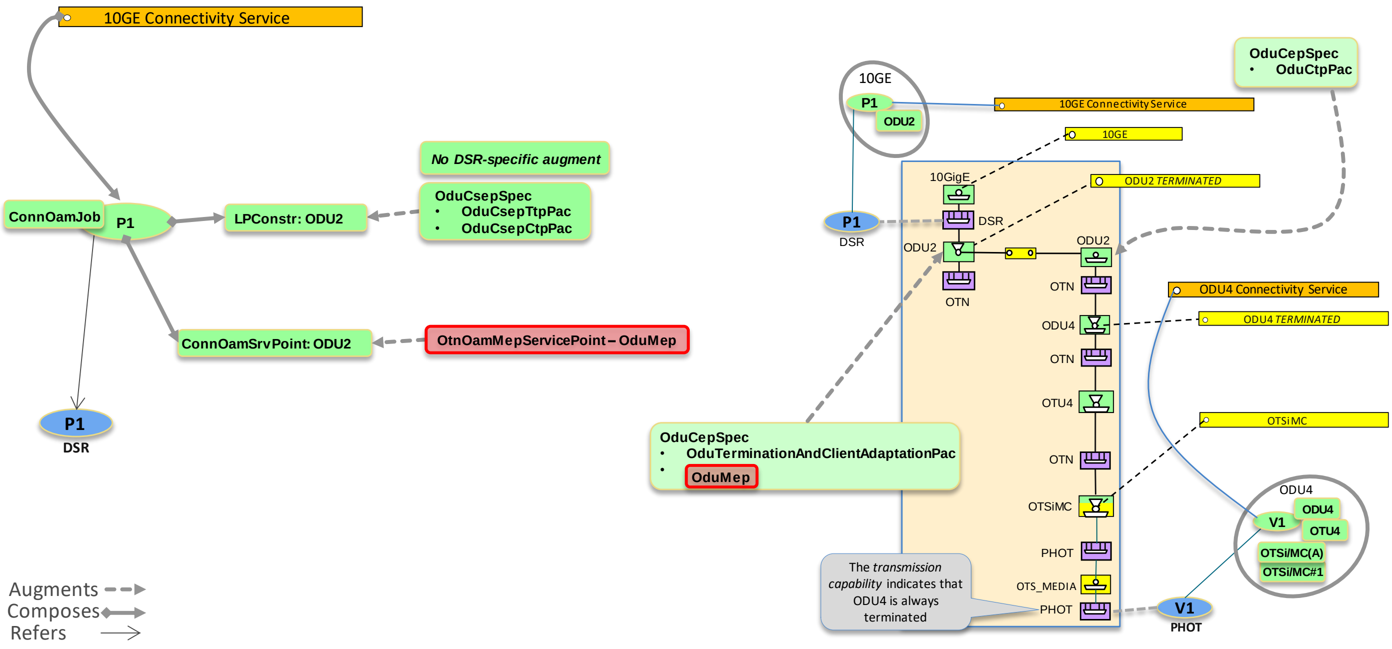


Figure 6-40 DSR/ODUj CS on ODUk CS

Figure 6-41 shows the configuration parameters for the provisioning of the DSR connectivity service on an existing *transponder-to-transponder* connectivity service (ODUk *Trail* CS).

This scenario foresees OTN multiplexing, i.e. the DSR payload is transported by an ODU lower order container (ODUj) which is multiplexed into the *transponder-to-transponder* ODUk *Trail*. With respect to Figure 6-40, this scenario foresees the flexibility at DSR layer rather than at ODUj layer.

The result includes the DSR connection plus the ODUj *terminated* connection.

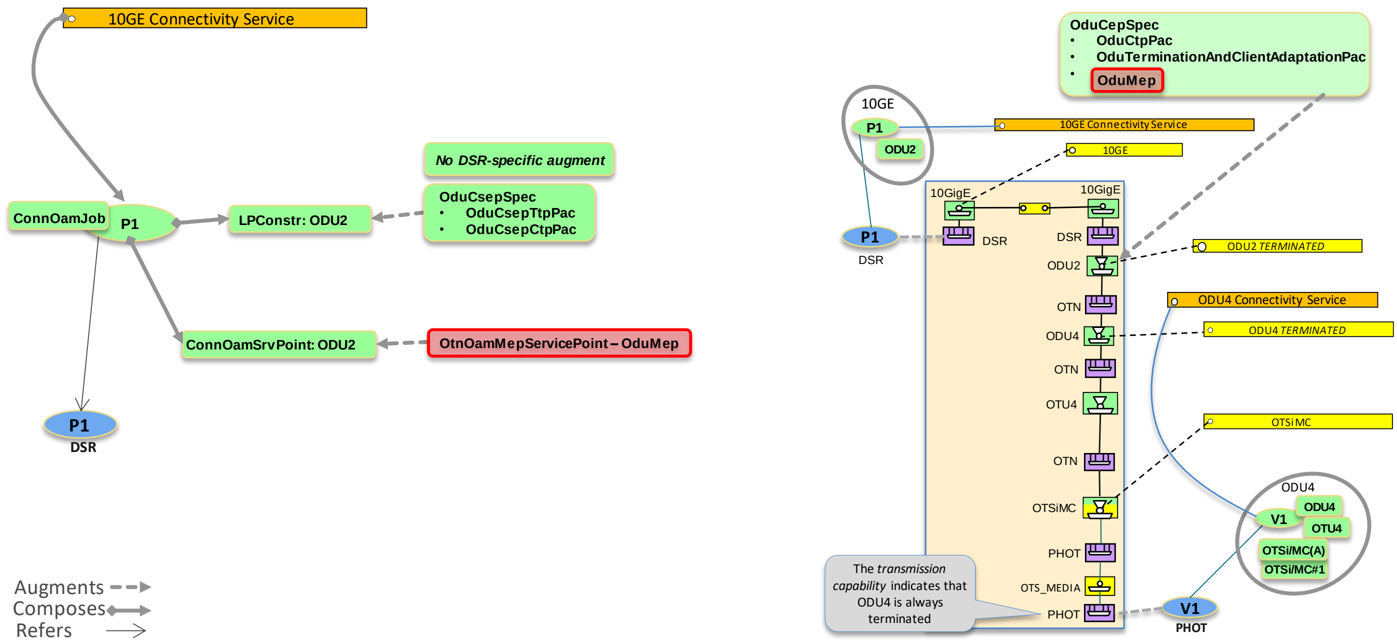


Figure 6-41 DSR CS on ODUj on ODUk CS (DSR flexibility)

6.2.2.6 MC Connectivity Service originating and/or terminating at Add/Drop port

Figure 6-42 shows the configuration parameters for the provisioning of the *ROADM-to-ROADM* connectivity based on the MC Connectivity Service, with the SIP on the Add/Drop side of the ROADM.

The result includes the MC top-level connection.

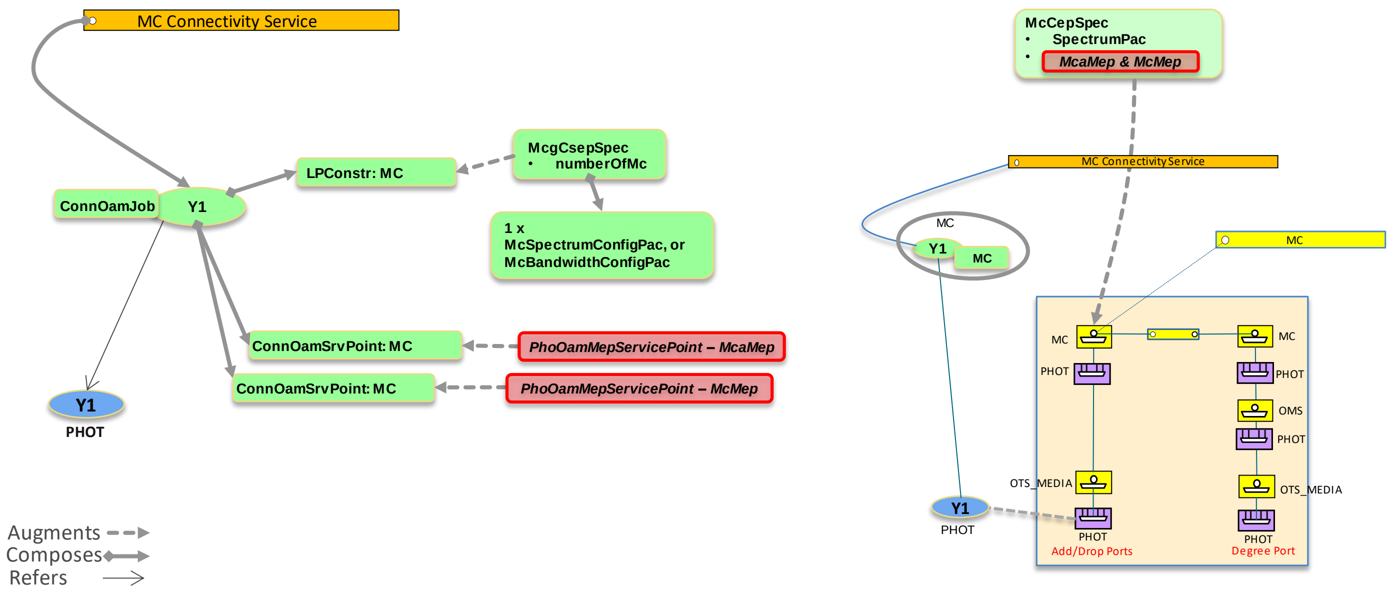


Figure 6-42 MC Connectivity Service at Add/Drop side

Figure 6-43 shows the configuration parameters for the provisioning of the *ROADM-to-ROADM* connectivity based on the MCG Connectivity Service, with the SIP on the Add/Drop side of the ROADM. This is modeled by node with MC cross-connections.

The result includes the MC connections which support the MCG.

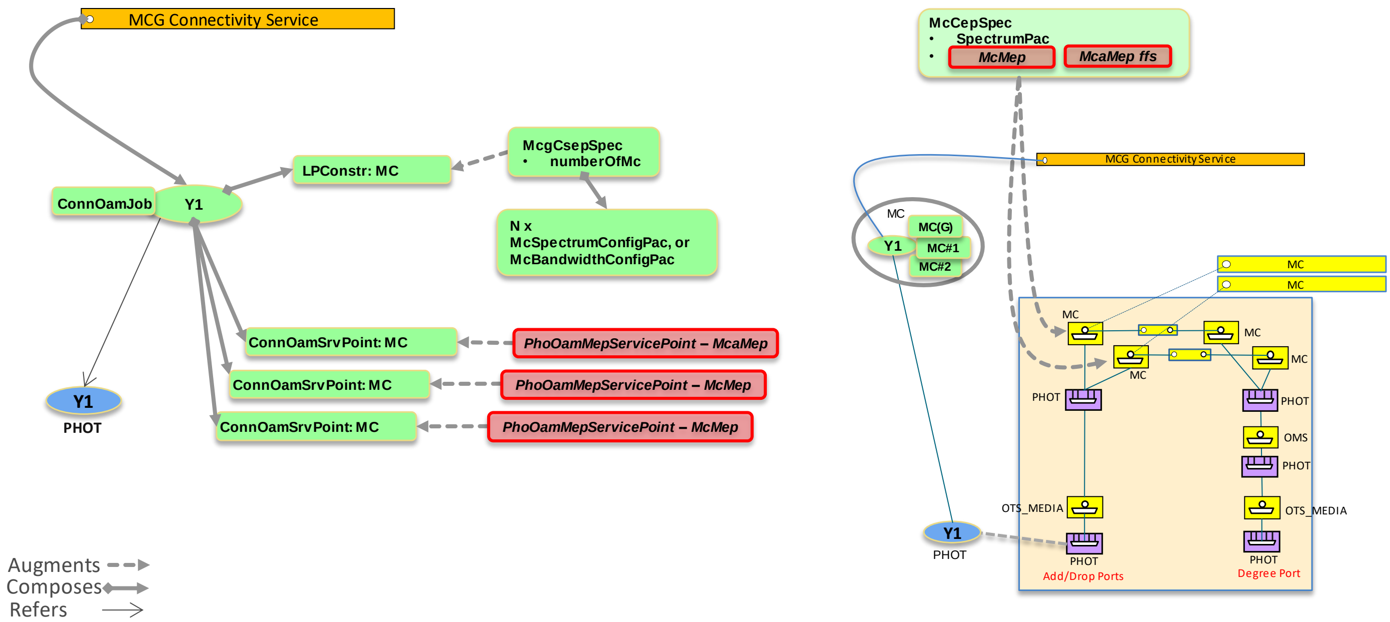


Figure 6-43 MCG Connectivity Service at Add/Drop side

Figure 6-44 shows the configuration parameters for the provisioning of the OTSiMC(G) connectivity service.

It is assumed that the supporting MC connection is either automatically created or reused if already existing. Note that MC parameters MAY be specified together with OTSiMC parameters to drive the creation of the server MC connection.

Note that each OTSiMC of the OTSiMCG MUST be a spectrum portion of the MC.

The result includes the OTSiMC connection(s) which support the OTSiMC(G) plus the MC connection.

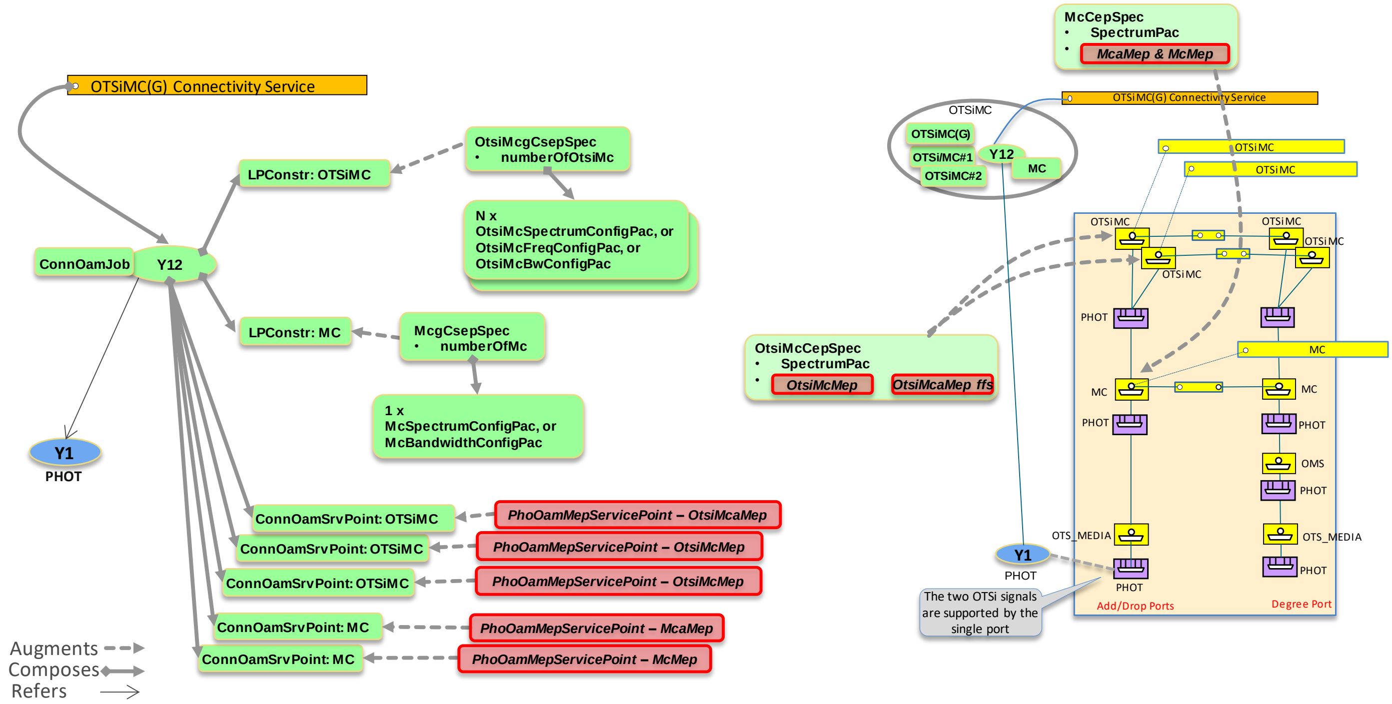


Figure 6-44 OTSiMCG CS on MC at Add/Drop side, MC Connection automatically created or reused

Figure 6-45 shows a similar scenario with respect to Figure 6-44, with the server controller also creating the MC connectivity service.

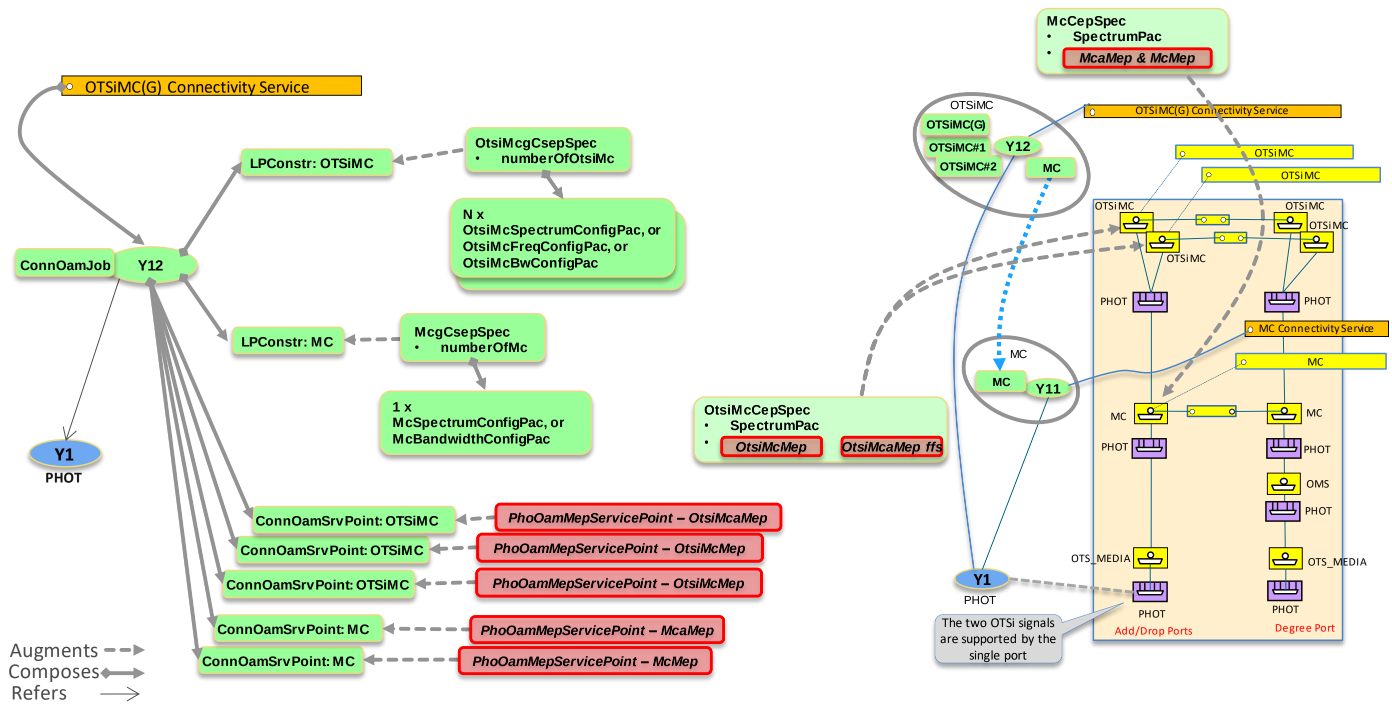


Figure 6-45 OTSiMCG CS on MC CS at Add/Drop side, auto creation of MC CS

Figure 6-46 shows the configuration parameters for the provisioning of the OTSiMC(G) connectivity service on an existing MC Connectivity Service.

The result includes the OTSiMC connection(s) which support the OTSiMC(G).

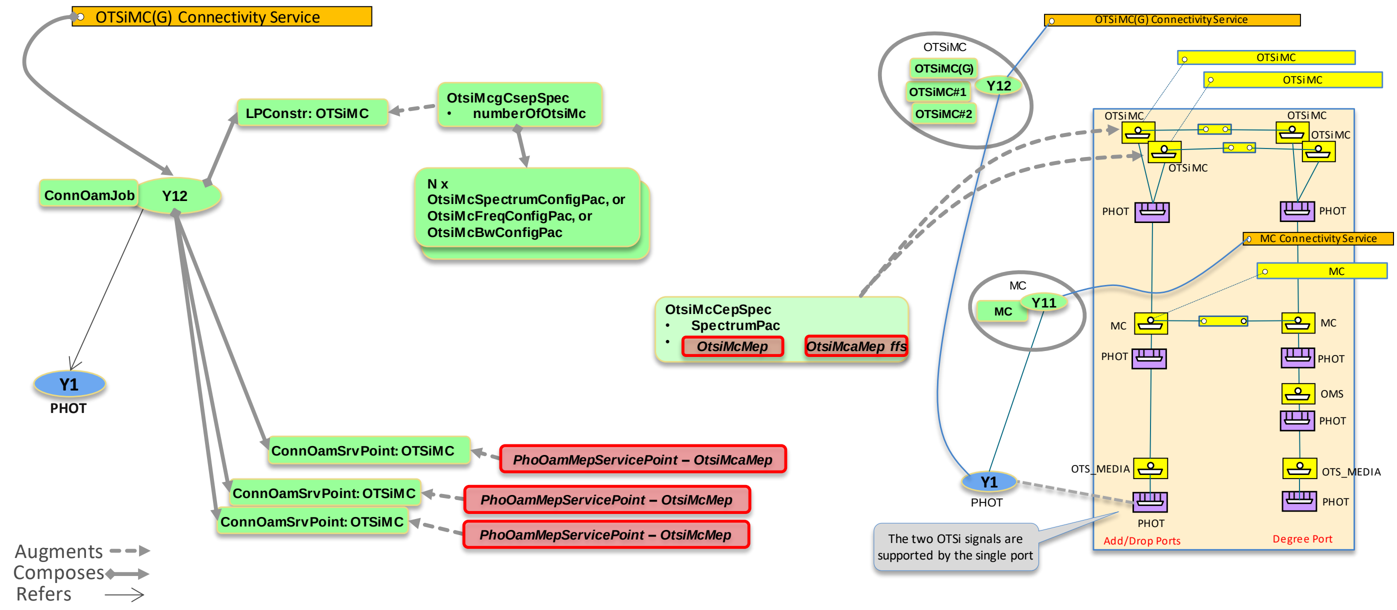


Figure 6-46 OTSiMCG CS on MC CS at Add/Drop side

6.2.2.7 MC Connectivity Service originating and/or terminating at Degree ports

Figure 6-47 shows the configuration parameters for the provisioning of the *ROADM-to-ROADM* connectivity based on the MC Connectivity Service, with the SIP on the Degree side of the ROADM.

The result includes the MC connection.

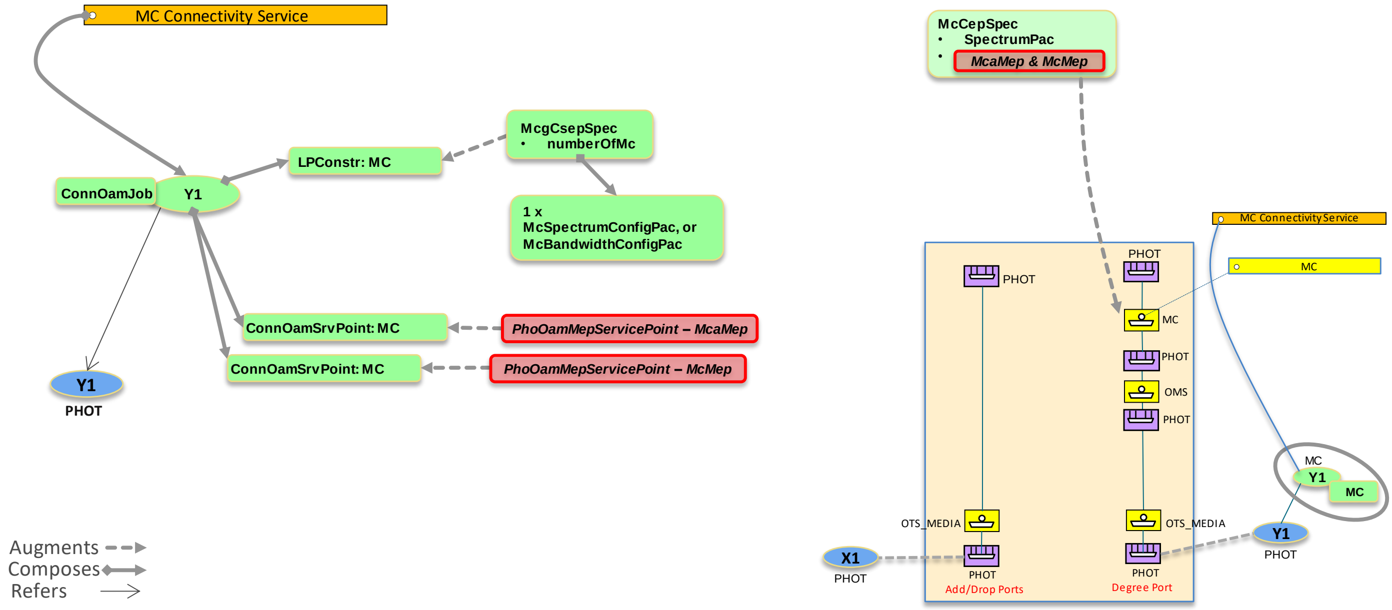


Figure 6-47 MC Connectivity Service at Degree side

Figure 6-48 shows the configuration parameters for the provisioning of the *ROADM-to-ROADM* connectivity based on the MCG Connectivity Service, with the SIP on the Degree side of the ROADM.

The result includes the MC connections which support the MCG.

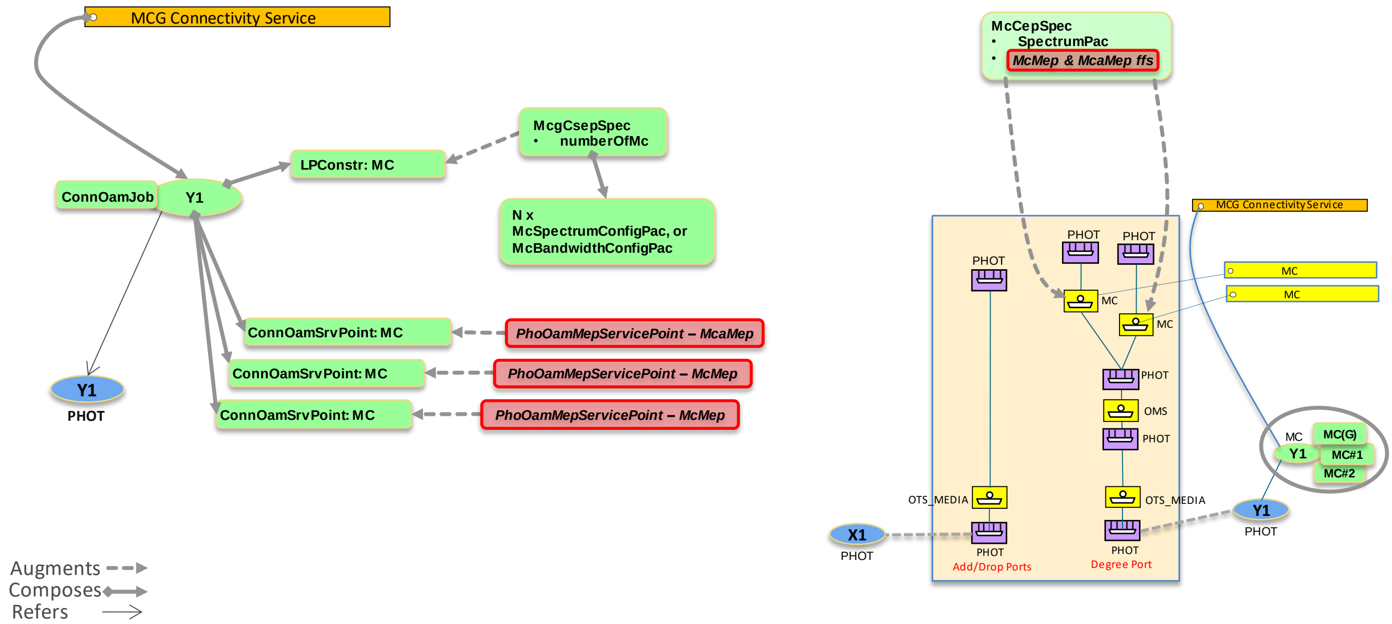


Figure 6-48 MCG Connectivity Service at Degree side

Figure 6-49 shows the configuration parameters for the provisioning of the OTSiMC(G) connectivity service.

It is assumed that the supporting MC connection is either automatically created or reused if already existing. Note that MC parameters MAY be specified together with OTSiMC parameters to drive the creation of the server MC connection.

Note that each OTSiMC of the OTSiMCG is a spectrum portion of the MC.

The result includes the OTSiMC connection(s) which support the OTSiMC(G) plus the MC connection.

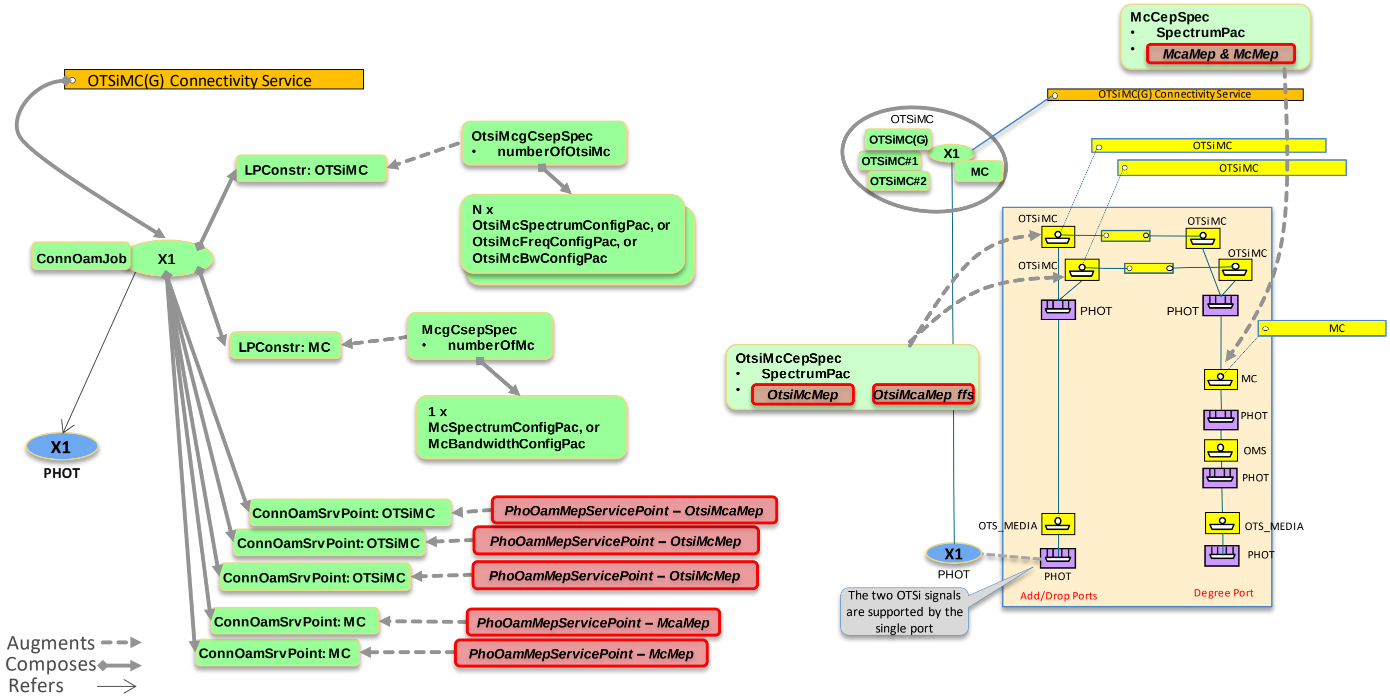


Figure 6-49 OTSiMC(G) CS on MC at Degree side, MC Connection automatically created or reused

Figure 6-50 shows a similar scenario with respect to Figure 6-49, with the server controller creating also the MC connectivity service.

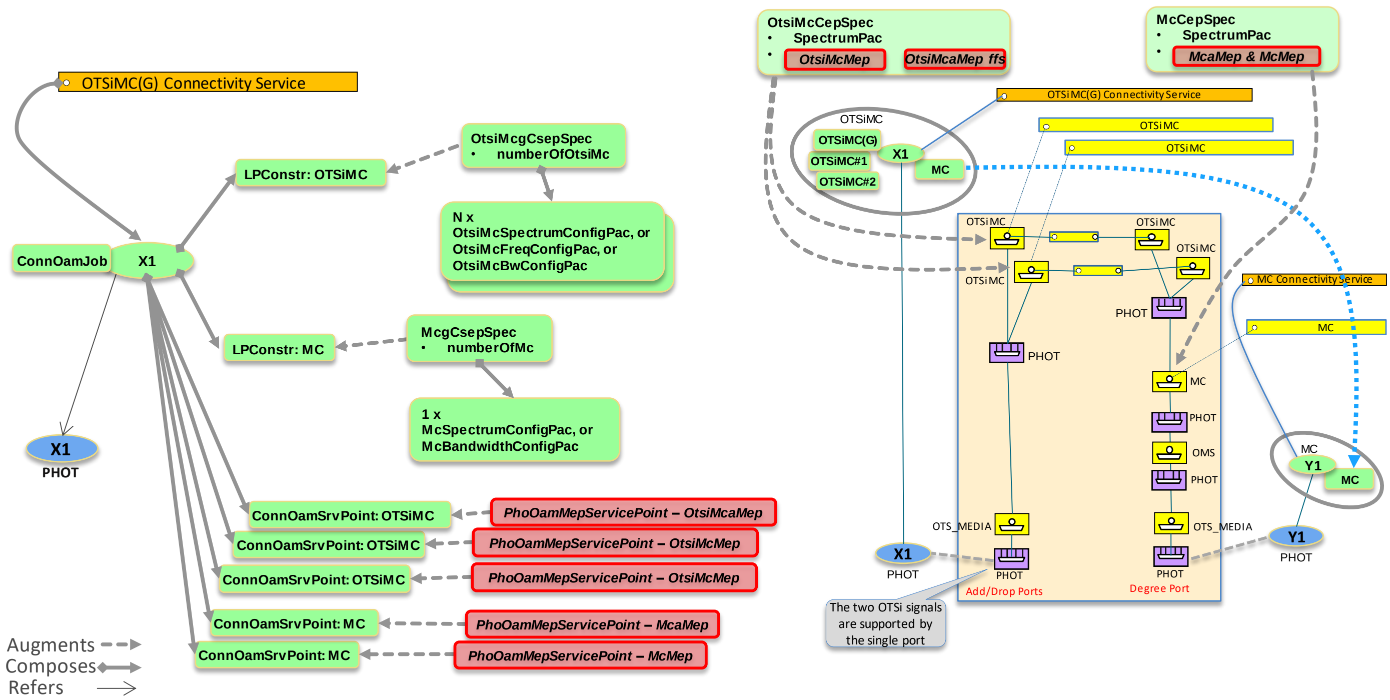


Figure 6-50 OTSiMCG CS on MC CS at Degree side, auto creation of MC CS

Figure 6-51 shows the configuration parameters for the provisioning of the OTSiMC(G) connectivity service on an existing MC Connectivity Service.

The result includes the OTSiMC connection(s) which support the OTSiMC(G).

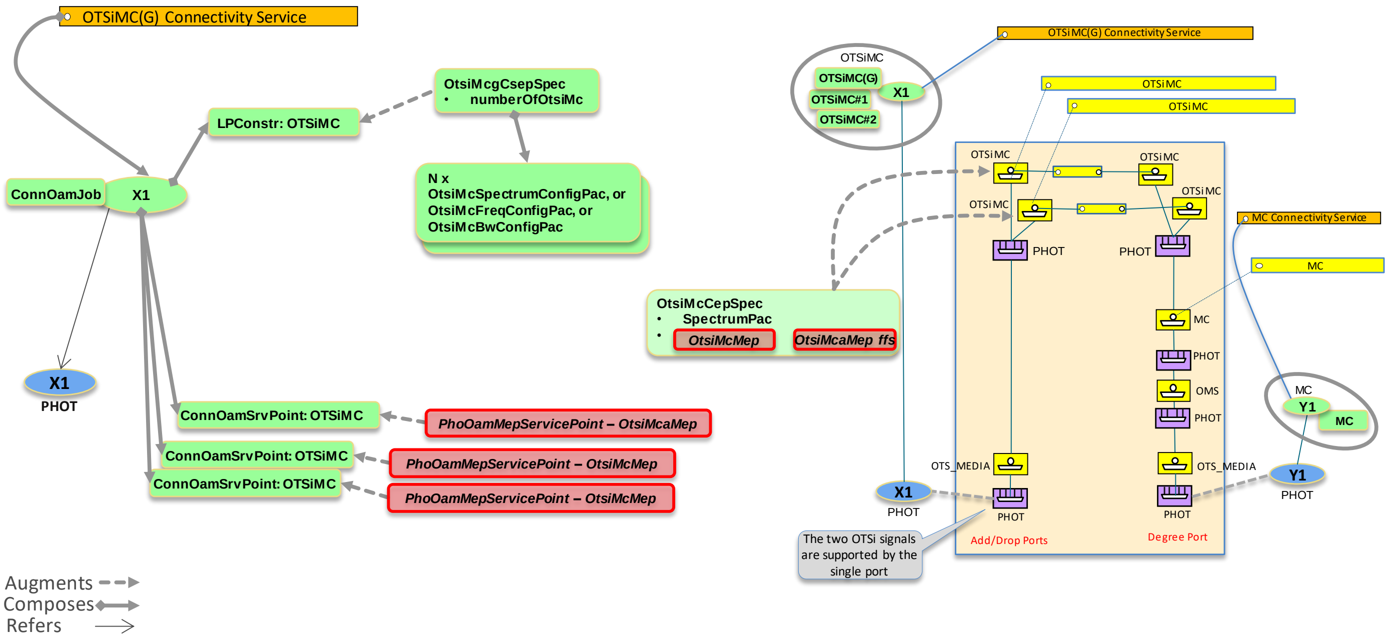


Figure 6-51 OTSiMC(G) CS on MC CS at Degree side

6.2.2.8 OTSiMC Connectivity Service without supporting MC connectivity

Figure 6-52 shows the configuration parameters for the provisioning of the OTSiMC(G) connectivity service without relying on or assuming any explicit MC connectivity supporting the OTSiMC(G). *Note that the presence or not of an MC layer connectivity is conveyed in the mux sequence capability of the PHOTONIC_MEDIA NEP (see UC0b). See also Section 6.2.2.6 regarding the configuration parameters for the provisioning of the OTSiMC(G) connectivity service with MC connectivity.*

In this scenario the OTSiMC(G) is directly supported by OMS Connections.

The result includes the OTSiMC connection(s) which support the OTSiMC(G).

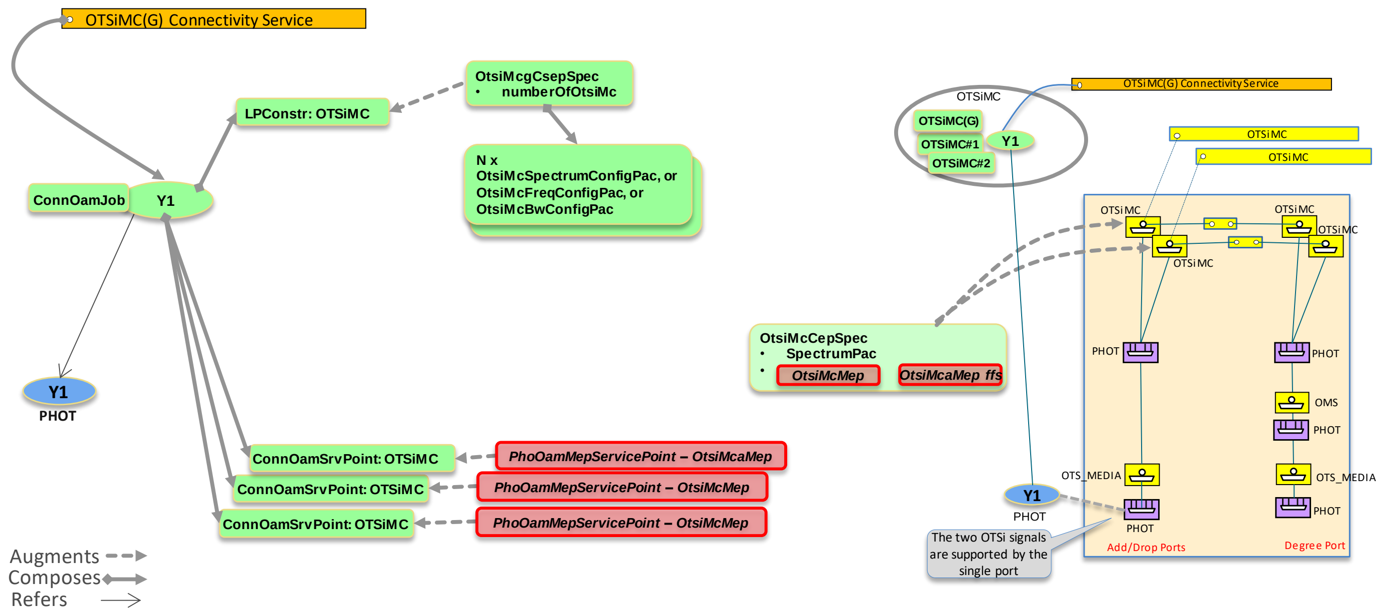


Figure 6-52 OTSiMC Connectivity Service without MC Layer

6.2.3 Use case 1.0: Generic Service Provisioning

The purpose of this generic UC is to provide an agreement in the connectivity service management, notably when a client requests a Connectivity Service between CSEPs (thus SIPs).

Number	**UC1.0**
Name	**Generic Unconstrained Service Provisioning**
Technologies involved	Optical, DSR, OTN
Process/Areas Involved	Planning and Operations
Brief description	The UC1.0 describes the provisioning of a GENERIC ***tapi-connectivity:connectivity-service*** instance between SIPs exposed by the TAPI-Server. It is a common framework for TAPI provisioning operations. Additional UC for specific layers will be detailed later. The underlying connection provisioning and management (including server layer connections e.g., ODU, OTU, OTSiMC, MC with intermediate regeneration connections if needed) is performed by the SDN Domain controller. The path of each server layer connection across the network topology is calculated by the controller and the connection(s) automatically provisioned. This UC defines the generic framework for the application of constraints in the provisioning of services. Specific constraints will be detailed in each applicable UC. Note that this UC also includes the parameters for the objects involved in the (subsequent) discovery of connectivity services and connections as per UC0c. In such discovery processes, Connectivity Services, Connections and CEP objects shall be understood as "provided by server" after the successful completion of the HTTP workflows.

<table>
<tr><td>Layers involved</td><td>DSR/DIGITAL_OTN/PHOTONIC_MEDIA</td></tr>
<tr><td>Type</td><td>Provisioning</td></tr>
<tr><td>Description & Workflow</td><td>The Use Case 1.0: Connectivity Service provisioning consists of the creation of a connectivity-service between SIPs at the either the DSR, DIGITAL_OTN or PHOTONIC_MEDIA layers and the retrieval of the generated connections information.

The first operation (1) triggers the creation of Connectivity-Service using the server NBI. If the operation is successful, the NBI server MUST return an HTTP Created 201 response message with the Location Header as specified in https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html#sec9.5.

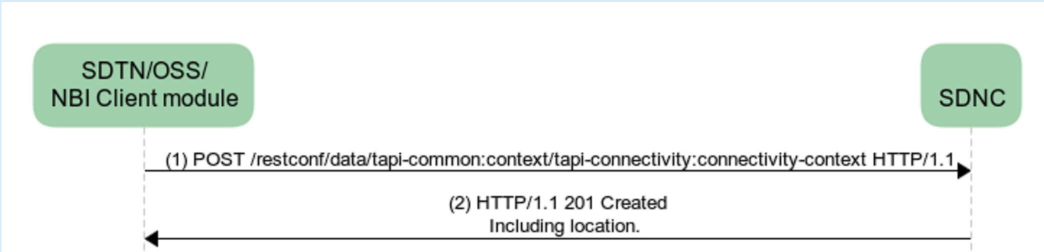


Figure 6-53 UC-1.0: Unconstrained end-to-end service provisioning.</td></tr>
</table>

6.2.3.1 Relevant parameters

Note that these tables are provided within use case 1.0 that deals with generic provisioning use cases. Nonetheless, they are also referred to by use cases related to connectivity service and connection discovery (UC 0c). Thus, they include both RW/RO parameters, but the latter shall not be used during the actual provisioning.

Note that the table lists the parameters of the CS object, the ones included in the POST are noted as “provided tapi-client”.

Table 29: Connectivity-service (**CS**) object definition.

connectivity-service	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
uuid	As defined in RFC 4122.	RW	M	• Provided by *tapi-client*
name	MUST include: "value-name": "SERVICE_NAME" "value": " [0-9a-zA-Z_]{64}"	RW	M	• Provided by *tapi-client and/or tapi-server.* For a client provisioned CS the server MUST store this SERVICE_NAME. For a server provisioned CS, the server MUST allocate a SERVICE_NAME. • *Mandatory status may be removed in a subsequent version of RIA.*
layer-protocol-name	One of the values	RW	M	• Provided by *tapi-client*

	"DSR", "DIGITAL_OTN" or "PHOTONIC_MEDIA"			
layer-protocol-qualifier	Depends on the Layer Protocol Name	RW	M	• Provided by *tapi-client* • For each layer (DSR, DIGITAL_OTN and PHOTONIC_MEDIA), all children identities MUST be supported (depending on hardware capabilities): children of DIGITAL_SIGNAL_TYPE, ODU_TYPE, OTU_TYPE and PHOTONIC_LAYER_QUALIFIER
direction	One of { "BIDIRECTIONAL" or "UNIDIRECTIONAL" }	RW	M	• Provided by *tapi-client* • *Note that the CSEPs direction may be different (e.g., a bidir CS uses 4 unidir CSEPs)*
administrative-state	One of {"UNLOCKED", "LOCKED"}	RW	O	• Provided by *tapi-client*
operational-state	One of {"ENABLED", "DISABLED"}	RO	M	• Provided by *tapi-server*
lifecycle-state	One of { "PLANNED", "POTENTIAL_AVAILABLE", "POTENTIAL_BUSY", "INSTALLED", "PENDING_REMOVAL" }	RO	O	• Provided by *tapi-server*
connectivity-constraint/requested-capacity/total-size	• "value": real, • "unit": *see tapi-common:capacity-unit*	RW	C	• Provided by *tapi-client.* *NOTES* • Whether this object is mandatory will depend on the layer and use case. • Mandatory for PHOTONIC_MEDIA/ OTSiMC, MC when specifying a slot width. • TAPI v2.4+ includes the **layer-protocol-qualifier** so the requested-capacity MAY be omitted if there is no ambiguity.
connectivity-constraint/service-type	"POINT_TO_POINT_CONNECTIVITY", or "POINT_TO_MULTIPOINT_CONNECTIVITY"	RW	M	• Provided by *tapi-client*
connection	List of {**connection-ref** - */tapi-common:context/tapi-connectivity:connectivity-context/connection/uuid}*	RO	M	• Provided by *tapi-server* • It MUST list the top-level connections supporting this connectivity service.
end-point	List of {**connectivity-service-end-point**}	RW	M	• Provided by *tapi-client* • Min elements 2.

Table 30: Connectivity-service-end-point (**CSEP**) object definition

connectivity-service-end-point	**/tapi-common:context/tapi-connectivity:connectivity-service/end-point**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
local-id	YANG string	RW	M	• Provided by *tapi-client*
name	MUST include "value-name": "CSEP_NAME" "value": " [0-9a-zA-Z_]{64}"	RW	M	• Provided by *tapi-client and/or tapi-server.* For a client provisioned CS, the server MUST store this CSEP_NAME. For a server provisioned CS, the server MUST allocate a CSEP_NAME. *Mandatory status may be removed in a subsequent version of RIA.*
layer-protocol-name	One of the values "DSR", "DIGITAL_OTN" or "PHOTONIC_MEDIA"	RW	O	• Provided by *tapi-client* • If present, this RIA only considers cases where this value matches the one provided in the CS.
layer-protocol-qualifier	Depends on the Layer Protocol Name	RW	O	• Provided by *tapi-client* • If present, this RIA only considers cases where this value matches the one provided in the CS.
administrative-state	One of {"UNLOCKED", "LOCKED"}	RW	O	• Provided by *tapi-client*
operational-state	One of {"ENABLED", "DISABLED"}	RO	O	• Provided by *tapi-server*
lifecycle-state	One of { "PLANNED", "POTENTIAL_AVAILABLE", "POTENTIAL_BUSY", "INSTALLED", "PENDING_REMOVAL" }	RO	O	• Provided by *tapi-server*
direction	One of { "BIDIRECTIONAL", "SINK", "SOURCE" }	RW	M	• Provided by *tapi-client* • *Unidirectional services are defined between a source and a sink CSEP. The definition is aligned with the notion of ITU-T trail, and from the internal viewpoint (within the domain), the data flows from the source to the sink CSEP.*
role	"SYMMETRIC"	RW	O	• Provided by *tapi-client* • This RIA only considers P2P and SYMMETRIC as port-role. If not present, it is considered SYMMETRIC.
csep-role	List of CSEP roles. Each role includes: role-name and connectivity-service-spec-reference (with connectivity-service-spec-name and connectivity-service-spec-id)	RW	O	• Provided by *tapi-client* • Depends on the Use Case.
protection-role	TAPI protection role of the CSEP (e.g., WORK, PROTECT, PROTECTED,...)	RW	C	• Provided by *tapi-client.* • Depends on the Layer and Use Case.
capacity	"total-size": {value: unit} • "value": decimal64 (fraction digits 7),	RW	O	• Provided by *tapi-client.* • Depends on the Layer and Use Case.

	• "unit": depends on the CS			• If present, this RIA only considers cases where this value matches the one provided in the CS. Please also see *connectivity-service/connectivity-constraint/requested-capacity/total-size*
service-interface-point	*"/tapi-common:context/service-interface-point/uuid"*	RW	M	• Provided by *tapi-client*
connection-end-point	List { *connection-end-point* }	RO	M	• Provided by *tapi-server* • List of CEPs of the connectivity service top-level connection at the same layer and qualifier than the CS that are instantiated over the NEP that the CSEP SIP is bound to (the CEPs of the immediate top-connection). [*Note this RIA only considers a single immediate top-connection, so there is only one CEP for each CSEP*]
profile	List of profile uuid refs	RW	C	• Provided by *tapi-client* • Selected profile(s) that apply to bidirectional CSEPs. • Depends on the Layer and Use Case.
sink-profile	List of profile uuid refs	RW	C	• Provided by *tapi-client* • Selected profile(s) that apply to Sink CSEPs • Depends on the Layer and Use Case.
source-profile	List of profile uuid refs	RW	C	• Provided by *tapi-client* • Selected profile(s) that apply to Source CSEPs • Depends on the Layer and Use Case.
protecting-connectivity-service-end-point	Reference to a protecting CSEP (CS uuid and CSEP local id)	RW	C	• Provided by *tapi-client* • Depends on the Layer and Use Case.
peer-fwd-connectivity-service-end-point	Reference to an associated CSEP instance (CS uuid and CSEP local id) from a forwarding perspective	RW	C	• Provided by *tapi-client* • Depends on the Layer and Use Case.
server-connectivity-service-end-point	Reference to a server CSEP (CS uuid and CSEP local id). *This option is deprecated in favor of the usage of layer protocol constraints*	RW	C	• Provided by *tapi-client* • Depends on the Layer and Use Case.
layer-protocol-constraint	List of { *layer-protocol-constraint* }	RW	C	• Depends on Use Case.

Table 31: Connectivity-service-end-point (**CSEP**) Layer Protocol Constraint object definition

layer-protocol-constraint	**/tapi-common:context/tapi-connectivity:connectivity-service/end-point/layer-protocol-constraint**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
local-id	YANG string, indexes the Layer Protocol Constraint (LPC)	RW	M	• Provided by *tapi-client*
name	List of name-value, value pairs	RW	O	• Provided by *tapi-client*
layer-protocol-name	One of the values "DSR", "DIGITAL_OTN" or "PHOTONIC_MEDIA"	RW	M	• Provided by *tapi-client*
layer-protocol-qualifier	Depends on the Layer Protocol Name	RW	M	• Provided by *tapi-client*
tapi-digital-otn:	Depends on the Layer Protocol Name	RW	C	• Provided by *tapi-client*

odu-connectivity-service-end-point-spec				• *Depends on the UC*
tapi-digital-otn: otu-connectivity-service-end-point-spec	Depends on the Layer Protocol Name Includes: tapi-digital-otn:otu-csep-ttp-pac	RW	C	• Provided by *tapi-client* • *Depends on the UC*
tapi-photonic-media: otsia-connectivity-service-end-point-spec	Depends on the Layer Protocol Name This RIA does not currently consider the independent provisioning of OTSi(A) services. The supported mechanism is to provision higher layer(s) and to convey info on the OTSiA sublayer (e.g., number of OTSi) as a dedicated *Layer Protocol Constraint* with **OTSiMC** Layer Protocol Qualifier.	RW	C	• Provided by *tapi-client* • *Depends on the UC* • *Notes:* otsia-connectivity-service-end-point-spec is decoupled from otu-connectivity-service-end-point-spec to enable (in a future release) clients other than DIGITAL_OTN (e.g., DSR over OTSi)
tapi-photonic-media: otsi-mcg-connectivity-service-end-point-spec	Depends on the Layer Protocol Name	RW	C	• Provided by *tapi-client* • *Depends on the UC*
tapi-photonic-media: mcg-connectivity-service-end-point-spec	Depends on the Layer Protocol Name	RW	C	• Provided by *tapi-client* • *Depends on the UC*

Table 32: ODU connectivity-service-end-point spec (**ODU CSEP SPEC**) object definition

odu-connectivity-service-end-point-spec	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/end-point/layer-protocol-constraint/tapi-digital-otn:odu-connectivity-service-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
odu-csep-common-pac	Includes: odu-rate in kb/s, opu-tributary-slot-size, one of 1G25 or 2G5. See yang for details.	RW	M	• Provided by *tapi-client*
odu-csep-ctp-pac/ tributary-slot-list	Set of distinct (i.e. unique) integers (e.g. 2, 3, 5, 9, 15 representing the tributary slots TS#2, TS#3, TS#5, TS#9 and TS#15) which represents the resources occupied by the ODUk CTP.	RW	C	• Provided by *tapi-client* Used in UC2b when selecting the channel. Refer to the Yang description
odu-csep-ctp-pac/tributary-port-number	Tributary port number that is associated with the ODUk CTP, when the ODUk CTP is multiplexed into a server layer ODU TTP object. See clause 14.4.1/G.709-2016, 14.4.1.4/G.709-2016 or 20.4.1.1/G.709-2016 for ODU-Cn	RW	O	• Provided by *tapi-client* Used in UC2b when selecting the channel.
odu-csep-ttp-pac	Includes: configured-mapping-type configured-client-type	RW	C	• Provided by *tapi-client* The configured mapping type is mandatory if there are several mapping types available for the DSR service. The *configured client type* is optional if this layer protocol constraint is used while provisioning the client. The *configured client type* could be used when provisioning ODU services in a bottom-up approach (use case not covered in this RIA).
odu-cn-csep-ttp-pac	Includes number-of-odu-c	RW	C	• Provided by *tapi-server* • *Used in ODU-Cn configurations.*

Table 33: OTU connectivity-service-end-point spec (**OTU CSEP SPEC**) object definition

otu-connectivity-service-end-point-spec	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/end-point/layer-protocol-constraint/tapi-digital-otn:otu-connectivity-service-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
otu-csep-ttp-pac	Includes: **fec-type** (either standard-fec-type, with an identity based on STANDARD_FEC_TYPE or proprietary-fec-type) **baud-rate** (uint64)	RW	C	• Provided by *tapi-client*

Table 34: MCG connectivity-service-end-point spec (**MCG CSEP SPEC**) object definition

mcg-connectivity-service-end-point-spec	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/end-point/layer-protocol-constraint/tapi-photonic-media:mcg-connectivity-service-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
number-of-mc	Number of component MC. Must be >= 1	RW	M	• Provided by *tapi-client* • *This RIA only considers an MCG provisioning from a single SIP (e.g, single add /drop port).* • Specifying spectrum by means of a list of grid configurations, or spectrum configurations or bandwidth configurations alternatives are usually exclusive, but this RIA does not enforce that.
mc-grid-config-pac	List of *MC Grid Configurations*, indexed by local-id. Each element contains: local-id and name list. **n, m** int64 (as per ITU-T G.694.1 grid) **frequency-constraint** with adjustment granularity and grid-type **power-management-config-pac**	RW	C	• Provided by *tapi-client* • Depends on the use case. It is used when the client specifies n and m • *power-management-config-pac is optional in all cases*
mc-spectrum-config-pac	List of *MC Spectrum Configurations*, indexed by local-id. Each element contains: local-id and name list. **spectrum** with upper-frequency and lower-frequency (in Hz) **edge-frequency-constraint** with adjustment granularity and grid-type **power-management-config-pac**	RW	C	• Provided by *tapi-client* • Depends on the use case. It is used when the client specifies upper and lower frenquency. • *power-management-config-pac is optional in all cases*
mc-bandwidth-config-pac	List of *MC Bandwidth Configurations*, indexed by local-id. Each element contains:	RW	C	• Provided by *tapi-client* • spectrum-bandwidth depends on the use case. It is used when the client only requires an amount of optical spectrum

	local-id and name list. **spectrum-bandwidth (**in Hz) **edge-frequency-constraint** with adjustment granularity and grid-type **power-management-config-pac**			• *power-management-config-pac is optional in all cases*

Table 35: OTSiA connectivity-service-end-point spec (**OTSiA CSEP SPEC**) object definition

otsia-connectivity-service-end-point-spec	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/end-point/layer-protocol-constraint/tapi-photonic-media:otsia-connectivity-service-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
number-of-otsi	Number of component OTSi. Must be >= 1	RW	C	• Provided by *tapi-client* • This RIA only considers an OTSiA provisioning from a single SIP (e.g., single transceiver line port).
total-power-warn-threshold-upper	To specify thresholds in the total power (for the group)	RW	O	• Provided by *tapi-client*
total-power-warn-threshold-lower	To specify thresholds in the total power (for the group)	RW	O	• Provided by *tapi-client*
otsi-config	List of *single* OTSi Config objects, indexed by local-id. Each entry includes: local-id and name array **central-frequency**: (in Hz) **laser-control**: One of {"FORCED-ON", "FORCED-OFF", "AUTOMATIC-LASER-SHUTDOWN", "UNDEFINED"} **otsi-threshold-power-config** with *total-power-warn-threshold-upper* *total-power-warn-threshold-lower* **power-management-config-pac**	RW	C	• Provided by *tapi-client* • The number of list elements MUST be equal to **number-of-otsi** • laser-control is optional • total-power-warn-threshold-* are used to specify thresholds in the total power (for the OTSi). These are optional. • power-management-config-pac is optional. The capability to set per OTSi launch power depends on the underlying controller exported capabilities (in some cases launch power is automatically selected by the controller based on optical line constraints). See Section 3.2.7 Implementations must document this feature

Table 36: OTSi-MCG connectivity-service-end-point spec (**OTSiMCG CSEP SPEC**) object definition

otsi-mcg-connectivity-service-end-point-spec	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/end-point/layer-protocol-constraint/tapi-photonic-media:otsi-mcg-connectivity-service-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
number-of-otsi-mc	Number of components OTSi-MC. Must be >= 1	RW	M	• Provided by *tapi-client*
OTSi MC configuration (Note: otsi-mc-spectrum-config, otsi-mc-grid-config, otsi-mc-bandwidth-config and otsi-frequency-config are exclusive and are different means to specify/constrain the requested OTSi media channel.				

otsi-mc-spectrum-config-pac	List of OTSiMC Spectrum Configurations, indexed by local-id. Each element contains: local-id and name list. **spectrum** with upper-frequency and lower-frequency (in Hz) **edge-frequency-constraint** with adjustment granularity and grid-type **center-frequency-constraint** with adjustment granularity and grid-type **center-frequency-offset** (in Hz) **power-management-config-pac**	RW	C	• Provided by *tapi-client* • power-management-config-pac is optional.
otsi-mc-grid-config-pac	List of *MC Grid Configurations*, indexed by local-id. Each element contains: local-id and name list. **n, m** int64 (as per ITU-T G.694.1 grid) **frequency-constraint** with adjustment granularity and grid-type **power-management-config-pac**	RW	C	• Provided by *tapi-client* • power-management-config-pac is optional.
otsi-mc-bandwidth-config-pac	List of *MC Bandwidth Configurations*, indexed by local-id. Each element contains: local-id and name list. **spectrum-bandwidth** in Hz **center-frequency-constraint** with adjustment granularity and grid-type **center-frequency-offset** **non-adjacent-spectrum** **edge-frequency-constraint** with adjustment granularity and grid-type **power-management-config-pac**	RW	C	• Provided by *tapi-client* • power-management-config-pac is optional.
otsi-mc-frequency-config-pac	List of *MC Frequency Configurations*, indexed by local-id. Each element contains: local-id and name list. **central-frequency (M)** **center-frequency-constraint** with adjustment granularity and grid-type **center-frequency-offset** **edge-frequency-constraint** with adjustment granularity and grid-type **power-management-config-pac**	RW	C	• Provided by *tapi-client* • power-management-config-pac is optional.

Table 37: Connection object definition

connection	**/tapi-common:context/tapi-connectivity:connectivity-context/connection**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
uuid	As defined in RFC 4122. The canonical representation uses lowercase	RO	M	• Provided by *tapi-server*
name	List of {value-name, value} MUST include "value-name": "CONNECTION_NAME" "value": "[0-9a-zA-Z_]{64}"	RO	M	• Provided by *tapi-server* • *This is mandatory for Top-Level Connection*
layer-protocol-name	One of the values "DSR", "DIGITAL_OTN" or "PHOTONIC_MEDIA"	RO	M	• Provided by *tapi-server* • *Depends on the UC*
layer-protocol-qualifier	Depends on the Layer Protocol Name	RO	M	• Provided by *tapi-server* • *Depends on the UC*
operational-state	One of ["ENABLED", "DISABLED"]	RO	M	• Provided by *tapi-server*
lifecycle-state	One of { "PLANNED", "POTENTIAL_AVAILABLE", "POTENTIAL_BUSY", "INSTALLED", "PENDING_REMOVAL"}	RO	O	• Provided by *tapi-server*
direction	One of ["UNIDIRECTIONAL", "BIDIRECTIONAL"]	RO	M	• Provided by *tapi- server* • See tapi-common:forwarding-direction
server-connection	List of top-connections (connection-ref) of the **immediate** supporting server layer. *Note: this parameter enables inter-layer navigation of connections without relying on NEP/CEP stack knowledge.*	RO	C	• Provided by *tapi-server* • *This only applies to top-connections* • *If a server only lists the immediate top-connection for a connectivity-service, then all top-connections MUST include its server-connection list.*
lower-connection	List of connection-refs (leafrefs to *tapi-common:context/tapi-connectivity:connectivity-context/connection/uuid)*	RO	M	• Provided by *tapi-server* • There are two cases where the lower-connection list attribute MUST NOT be present: i) Cross-connections, ii)Top-connections where the representation of lower partitioning levels does not provide further information.
connection-end-point	List of connection-end-point-refs, including leafrefs to the respective topology, node, NEP and CEP uuid.	RO	M	• Provided by *tapi-server*
route	List of { **route** }	RO	C	• Provided by *tapi-server* • Mandatory for each Top Connection, see [TAPI-CONN-MODEL-REQ-4]
switch-control	List of { **switch-control** }	RO	C	• Provided by *tapi-server* • The use of this attribute is only applicable on the relevant connection objects which implement the protection logic described in UCs 5a, 5b, 5c, etc.
supported-client-link	List of {link-ref , topology-uuid + link-uuid }	RO	O	• Provided by *tapi-server*

	This applies only in the implementations where links other that the bottom-most in the flat topology are explicit. In such case the supporting top-connection SHOULD include the link ref.			• Note that links are only supported by terminated connections. In other words, only terminated CEPs support a NEP. • *This RIA only considers a connection supporting a single link.* • *This RIA only considers links supported by terminated connections.*

Table 38: Connection-end-point (**CEP**) object definition

connection-end-point	**/tapi-common:context/tapi-topology:topology-context/topology/node/owned-node-edge-point/tapi-connectivity:cep-list/connection-end-point**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
uuid	As defined in RFC 4122. The canonical representation uses lowercase	RO	M	• Provided by *tapi-server*
name	List of {value-name: value} MUST include "value-name": "CEP_NAME" "value": " [0-9a-zA-Z_]{64}"	RO	M	• Provided by *tapi-server*
layer-protocol-name	One of "DSR", "DIGITAL_OTN", "PHOTONIC_MEDIA" depending on the Layer of the connection	RO	M	Provided by *tapi-server*
layer-protocol-qualifier	Depends on the Layer Protocol Name	RO	M	• Provided by *tapi-server* All children identities defined for ["DIGITAL_SIGNAL_TYPE", "ODU_TYPE", "OTU_TYPE", "PHOTONIC_LAYER_QUALIFIER"] MAY be supported depending on the relevant protocol name.
direction	One of ["BIDIRECTIONAL", "SINK", "SOURCE"], describes the CEP direction.	RO	M	• Provided by *tapi-server* • *Unidirectional connections are defined between a source and a sink CEP. The data flows from the source to the sink CEP*
cep-role	List of CEP roles, each including : role-name connection-spec-reference (with connection-spec-name and connection-spec-id)	RO	O	• Provided by *tapi-server*
mep-mip-list	Container showing the supported list of MEPs and MIPs.	RO	C	• Provided by *tapi-server*
connection-port-role	One of ["SYMMETRIC", "ROOT", "LEAF", "TRUNK" or "UNKNOWN"]	RO	M	• Provided by *tapi-server* • *NOTE: This RIA only considers SYMMETRIC roles*
protection-role	TAPI protection role of the CSEP (e.g., WORK, PROTECT, PROTECTED,...)	RO	C	• Provided by *tapi-server* • Depends on the Layer and Use Case.
operational-state	One of {"ENABLED", "DISABLED"}	RO	M	• Provided by *tapi-server*
lifecycle-state	One of { "PLANNED", "POTENTIAL_AVAILABLE", "POTENTIAL_BUSY", "INSTALLED", "PENDING_REMOVAL"}	RO	O	• Provided by *tapi-server*
termination-state	One of { "CAN_NEVER_TERMINATE", "PERMANENTLY_TERMINATED" }	RO	M	• Provided by *tapi-server* • *Mandatory for all protocol layer names and qualifiers.* NOTE on DIGITAL_OTN: • *In the case of an ODU CEP that is terminated, the ODU-TTP PAC MUST be present (client adaptation). In the*

				case the CEP represents a container multiplexed into a higher order container, the ODU-CTP MUST also be present. • *In the case of an ODU CEP that is not terminated, the ODU-CTP PAC MUST be present (including the slot position).* NOTE on OTSi/OTSiMC: • *In the case of an OTSiMC CEP that is terminated, the OTSI Termination PAC MUST be present, and the Spectrum PAC MAY be present (to project the MC information bound to the OTSi to the node modeling a transceiver device)* • *In the case of an OTSiMC CEP that is not terminated, only the Spectrum PAC MUST be present and the OTSi PAC MUST NOT be present (since it is not applicable in the ROADM)*
aggregated-connection-end-point	List of { ***node-edge-point-ref*** }	RO	O	• Provided by *tapi-server*
parent-node-edge-point	List of { ***node-edge-point-ref*** }	RO	M	• Provided by *tapi-server* • *This RIA only considers CEP instances over a single parent NEP.*
client-node-edge-point	List of { ***node-edge-point-ref*** }	RO	M	• Provided by *tapi-server* • *This RIA only considers CEP instances supporting a single client NEP.*
profile	List of profile uuid refs	RO	C	• Provided by *tapi-server* • Profiles used to reflect properties that are either applicable to bidirectional CEPs or are common to either Sink/Source directions (avoid duplication) or the direction can be inferred from the properties in the profile. • MUST appear if the CEP supports specific profiles.
sink-profile	List of profile uuid refs	RO	C	• Provided by *tapi-server* • Profiles that apply to the sink direction of the CEP. • MUST appear if the CEP supports specific sink profiles.
source-profile	List of profile uuid refs	RO	C	• Provided by *tapi-server* • Profiles that apply to the source direction of the CEP. • MUST appear if the CEP supports specific source profiles.
Technology Specific Parameters				
tapi-digital-otn: odu-connection-end-point-spec	**{ odu-connection-end-point-spec }**	RO	C	• Provided by *tapi-server* • MUST augment CEPs at the ODU layer qualifier
tapi-digital-otn: otu-connection-end-point-spec	**{ otu-connection-end-point-spec }**	RO	C	• Provided by *tapi-server* • MUST augment CEPs at the OTU layer qualifier
tapi-photonic-media: otsi-mc-connection-end-point-spec	**{ otsi-mc-connection-end-point-spec }**	RO	C	• Provided by *tapi-server* • MUST augment CEPs at the PHOTONIC_MEDIA layer with OTSiMC qualifier that are not terminated (e.g., ROADM ports) and

				MAY augment CEPs at the PHOTONIC_MEDIA layer that are terminated (e.g. • transceiver line ports)
tapi-photonic-media: mc-connection-end-point-spec	**{ mc-connection-end-point-spec }**	RO	C	• Provided by *tapi-server* • MUST augment CEPs at the PHOTONIC_MEDIA layer with MC qualifier.
tapi-photonic-media: oms-connection-end-point-spec	**{ oms-connection-end-point-spec }**	RO	C	• Provided by *tapi-server* • MUST augment CEPs at the PHOTONIC_MEDIA layer with MC qualifier.
tapi-photonic-media: ots-media-connection-end-point-spec	**{ ots-media-connection-end-point-spec }**	RO	C	• Provided by *tapi-server* • MUST augment CEPs at the PHOTONIC_MEDIA layer with OTS-MEDIA qualifier.

Table 39: odu-connection-end-point-spec (**ODU CEP**) object definition

odu-connection-end-point-spec	**/tapi-common:context/tapi-topology:topology-context/topology/node/owned-node-edge-point/tapi-connectivity:cep-list/connection-end-point/tapi-digital-otn:odu-connection-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
odu-common	• odu-rate: uint64 • odu-rate-tolerance: uint64	RO	C	• Provided by *tapi-server* • **odu-rate** is only meaningful for ODUFlex. • **odu-rate-tolerance** Standardized values are defined in Table 7-2/G.709. It is optional. • Note: TAPI v2.1.3 included *odu-type*, which is no longer used here (the information is already included in the layer protocol qualifier)
odu-term-and-adapter	• opu-tributary-slot-size: ["1G25", "2G5"] • auto-payload-type? boolean • configured-client-type: [DIGITAL_SIGNAL_TYPE] • configured-mapping-type: ["AMP", "BMP", "GFP_F", "GMP", "TTP_GFP_BMP", "NULL"] • accepted-payload-type, including ○ "named-payload-type": ["UNKNOWN", "UNINTERPRETABLE"] ○ "hex-payload-type": string, • number-of-odu-c: uint64 • odu-cn-effective-time-slot: List uint64: Set of distinct (i.e. unique) integers (e.g. 2, 3, 5, 9, 15, 34 representing the tributary slots TS#1.2, TS#1.3, TS#1.5, TS#1.9, TS#1.15, and TS#2.14) which represents the list of effective time slots which are available for carrying ODUk clients (see ITU-T Recommendation G.709 (v5) Clause 20.1). • odu-mep, including ○ "txti" ○ "otn-oam-common", including ex-dapi, ex-sapi, deg-thr, tim-det-mode, tim-act-disabled, deg-m ○ "odu-mep-status" with "acti" and "tcm-fields-in-use"	RO	C	• Provided by *tapi-server* • odu-term-and-adapter *is mandatory for CEPs that are TTP.* • *opu-tributary-slot-size* applies only to ODU2 and ODU3. • *configured-client-type* accepts any child identities defined for ["DIGITAL_SIGNAL_TYPE"] (Note that all currently defined DSR signal types can be payload of an ODU container. This may change in the future). • *number-of-odu-c* applies only to ODU-CN CEPs. • *hex-payload-type* attribute is a string containing a 2-digit Hex code that indicates the new accepted payload type in uppercase letters (e.g., “3F”) as-if pattern '[0-9A-F]{2}’ • *otn-oam-common, odu-mep-status:* attributes is optional.

odu-ctp	Includes { tributary-slot-list, tributary-port-number, accepted-msi} • tributary-slot-list : List of uint64 • tributary-port-number: uint64 • accepted-msi? string	RO	M	• Provided by *tapi-server*
odu-protection	aps-enable : Boolean aps-level: uint64	RO	O	• Provided by tapi-server

Table 40: otu-connection-end-point-spec (**OTU CEP**) object definition

otu-connection-end-point-spec	**/tapi-common:context/tapi-topology:topology-context/topology/node/owned-node-edge-point/tapi-connectivity:cep-list/connection-end-point/tapi-digital-otn:otu-connection-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
otu-ttp-pac	Includes: **otu-mep** including: ○ "txti" ○ "otn-oam-common", including ex-dapi, ex-sapi, deg-thr, tim-det-mode, tim-act-disabled, deg-m ○ "otu-mep-status" with "acti" ○ "fec-monitoring" : boolean ○ "fec-corrected-error-threshold" : uint64 ○ **"otsia-mep"** including ○ "total-power-warn-threshold-upper" and ○ "total-power-warn-threshold-lower" decimal64 **fec-type**: with ○ "standard-fec-type" : identity derived from STANDARD_FEC_TYPE , or ○ "proprietary-fec-type" : string **baud-rate**: uint64	RO	C	• Provided by *tapi-server.* • otu-ttp-pac *is mandatory for OTU CEPs.* • *otn-oam-common, otu-mep-status, otsia-mep:* attributes are optional. • *fec-type, baud-rate* is optional.

Table 41: otsi-mc-connection-end-point-spec (**OTSiMC CEP**) object definition

otsi-mc-connection-end-point-spec	**/tapi-common:context/tapi-topology:topology-context/topology/node/owned-node-edge-point/tapi-connectivity:cep-list/connection-end-point/tapi-photonic-media:otsi-mc-connection-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
otsi-termination-pac	Includes { **selected-central-frequency**, **selected-spectrum** with upper-frequency, lower-frequency, **laser-properties,** } With laser-properties{ laser-status, laser-application-type, laser-bias-current, laser-temperature	RO	C	• Provided by *tapi-server* • This is only present if the CEP is terminated. • The **selected-central-frequency** of the laser specified in Hz. It is the oscillation frequency of the corresponding electromagnetic wave. • The **selected-spectrum** is conditional (e.g., it is optional if the transceiver profile already allows to deduce a OTSi spectrum) • The selected application identifier and the selected modulation can be

	} • “laser-status”: [“ON”,”OFF”,”PULSING”, “UNDEFINED”] • “laser-application-type”: [“PUMP”, “MODULATED”, “PULSE”] • “laser-bias-current”: decimal64, “laser-temperature”: decimal64,			obtained from the transceiver profile referred to in the CEP (see connection-end-point/profile) • The **frequencies** are specified in Hz. • NOTE: single carrier vs multi-carrier considerations are for further study.
spectrum-pac	See Table 45	RO	C	• Provided by *tapi-server* • This is mandatory if the CEP is not terminated (Transceiver) and optional if the CEP is terminated (Transceiver) • This can be different from the value in the selected spectrum of the OTSI termination pac.
power-measurement-pac	See Table 45	RO	O	• Provided by *tapi-server*

Table 42: mc-connection-end-point-spec (**MC CEP**) object definition

mc-connection-end-point-spec	**/tapi-common:context/tapi-topology:topology-context/topology/node/owned-node-edge-point/tapi-connectivity:cep-list/connection-end-point/tapi-photonic-media:mc-connection-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
spectrum-pac	See Table 45	RO	M	• Provided by *tapi-server* •
power-measurement-pac	See Table 45	RO	C	• Provided by *tapi-server*

An OMS CEP includes the following augmentation show in the table below. Note that, as opposed to the MC and OTSiMC CEPs, the spectrum-pac attribute for the OMS CEP is a list of elements, which provides more flexibility for spectrum management.

Note that this RIA *does not mandate a single approach to model multiple optical bands* (e.g., C, L, S). Implementations may choose to have *a single OMS CEP instance* and manage pools or to have *an OMS CEP per band*. In both cases, each OMS CEP will then support a *single* PHOTONIC_MEDIA NEP with supported MC CEPs. Such NEP is expected to manage a list of supportable/available/occupied spectrum to reflect the different MC pools (bands) (see tapi-topology:owned-node-edge-point/tapi-photonic-media:photonic-media-node-edge-point-spec/spectrum-capability-pac/supportable-spectrum)

Table 43: oms-connection-end-point-spec (**OMS CEP**) object definition

oms-connection-end-point-spec	**/tapi-common:context/tapi-topology:topology-context/topology/node/owned-node-edge-point/tapi-connectivity:cep-list/connection-end-point/tapi-photonic-media:oms-connection-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
spectrum-pac	List of Elements, for the description of each Element See Table 45	RO	M	• Provided by *tapi-server*
power-measurement-pac	See Table 45	RO	C	• Provided by *tapi-server*
amplification	List of Amplification elements. Each element includes frequency-range with lower- upper ingress-direction actual-gain	RO	C	• Provided by *tapi-server* • This parameter (list) is added for CEPs that support one or more logical amplification function. • It is encoded as a list which includes all the amplification functions

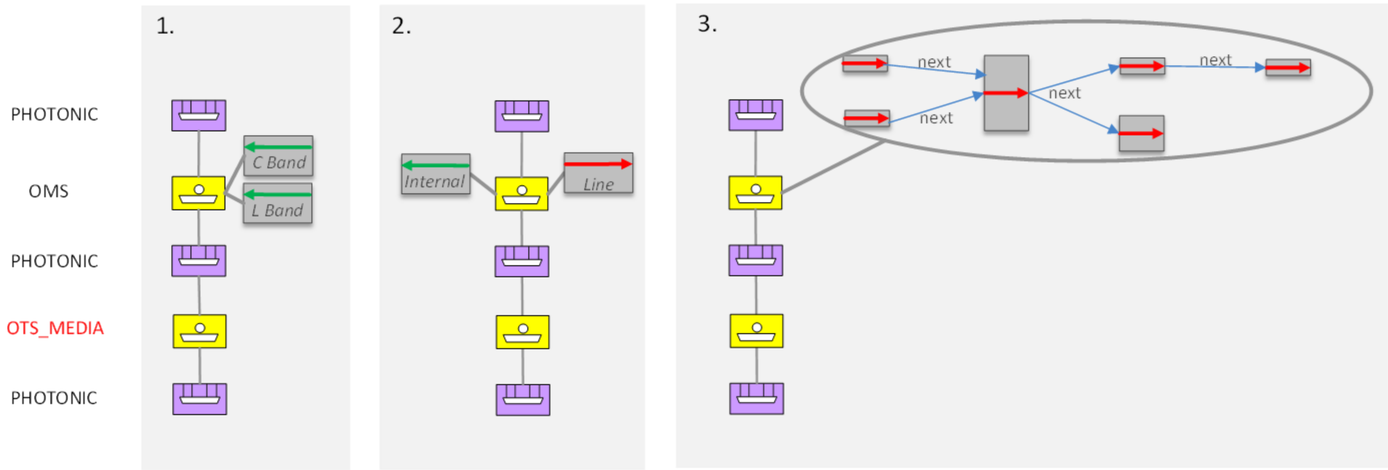
	actual-tilt out-voa in-voa optical-output-power optical-input-power profile (see next) geolocation (currently unused in RIA) local-id name			involved in the CEP (identified by their local id). • It is possible to have a "chain" of amplification functions (the contained amplification reference to one or more "next" elements in the chain).This chain must be traversed starting from the amplification function(s) with first-of-chain true. • More than one functions can be first-of-chain given their frequency ranges. • For bidirectional CEPs it may be possible to have 2 amplifications • The link with the physical equipment is for further study (e.g., NEP links to Access Port)
amplification/profile	List of applicable profiles	RO	C	• Provided by *tapi-server*
amplification/amplification	List of amplification function references , including topology-uuid, node-uuid, node-edge-point-uuid, connection-uuid, amplification-local-id	RO	C	• Provided by *tapi-server* • This is a list instead of a single ("next") element, because it may be possible to specify multiple next amplification functions depending e.g. (on their respective frequency ranges). Implementations should check the amplification chain based on this information • All amplification functions in a chain must have the same ingress-direction value.
oms-general-optical-params	*List of entries (max 2), which includes:* frequency-range/upper-frequency frequency-range/lower-frequency ingress-direction (bool) generalized-snr power-params/power-spectral-density/nominal-power-spectral-density (decimal64) power-params/channel-power/nominal-carrier-power (decimal64)	RO	C	• Provided by *tapi-server* • Used in UC12d to characterize an OMS connection. • GSNR Measured in dB@0.1nm (over 0.1 nm resolution bandwidth). • Note: ingress-direction For unidirectional CEPs, there is only one oms-general-optical-params (M) and the ingress direction is true for SINK CEPs (false for SOURCE CEPs) For bidirectional CEPs, if there is only one oms-general-optical-params (to avoid duplicating information in complementary CEPs of the OMS connection) it is related to the CEP sink/ingress direction. If there are 2 oms-general-optical-params (M), the one with ingress-direction true corresponds to the SINK function of the CEP. • Note: generalized-snr and power-params are optional.

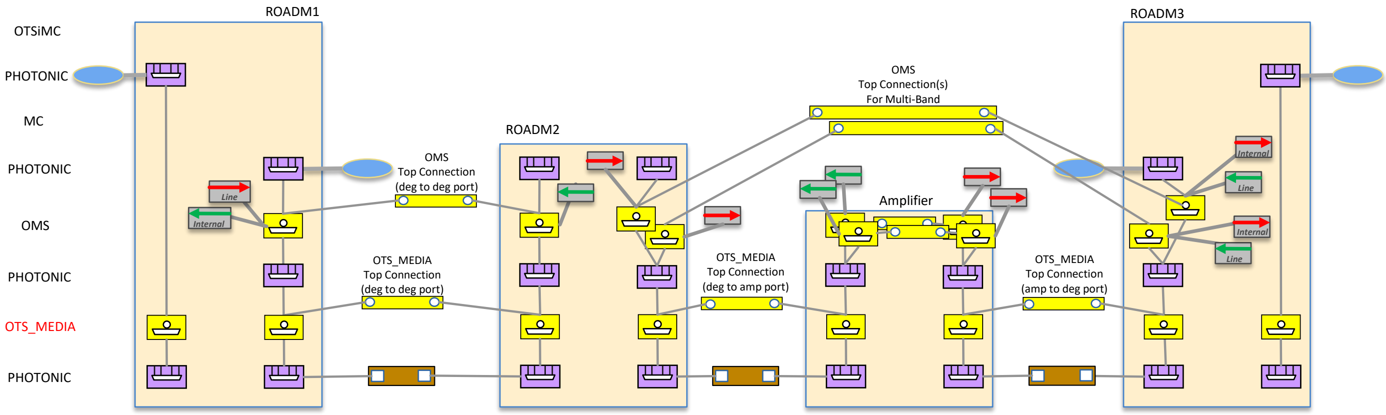
In the next figure you can notice:

1) Different amplification functions based on operating frequency ranges (e.g., C band and L band)

2) Ingress and Egress amplification functions supported over the same CEP (e.g., booster and preamplifier in the same ROADM degree)

3) Parallel chain(s) of amplification with one common “stage” which splits based e.g., on frequency range.





Note, as shown in the previous figure that:

- It is an implementation choice to decide which CEPs in each node better support one or more amplification functions as per the underlying hardware capabilities.
- For a given amplification function (gray boxes) the red and green arrows specify e.g., ingress and egress functions. For example, in ROADM1 the red “line” amplification function is the output *(booster)* amplification (ingress-direction is false), and the green “internal” amplification function is the input *(pre-amplifier)* amplification (ingress-direction is true). In ROADM2 the CEP that terminates the OMS from ROADM1 is bidirectional yet only defines an output amplification function (no pre-amplifier).

Table 44: ots-media-connection-end-point-spec (**OTS-MEDIA CEP**) object definition

ots-media-connection-end-point-spec	**/tapi-common:context/tapi-topology:topology-context/topology/node/owned-node-edge-point/tapi-connectivity:cep-list/connection-end-point/tapi-photonic-media:ots-media-connection-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**

spectrum-pac	List of Elements, for the description of each Element See Table 45	RO	M	• Provided by *tapi-server*
power-measurement-pac	See Table 45	RO	C	• Provided by *tapi-server*
ots-impairements	List of up to two entries. In case of bidirectional OTS CEPs one must have ingress-direction TRUE. Each OTS impairment element of the list includes: **ingress-direction** and **impairment-route-entry** which, in turn is a list of elements (or chain, typically one per link span) , each element either *ots-concentrated-loss*/concentrated-loss or *ots-fiber-span-impairements* with fiber-type-variety pmd length total-loss (*) or loss-coef, connector-in, connector-out	RO	C	• Provided by *tapi-server* • *NOTE: ots-concetrated-loss and ots-fiber-span-impairments are expected to be used exclusively.* For bidirectional CEPs, • If only one instance of ots-impairments parameters is present, it is related to the CEP sink/ingress direction. In such case, ingress-direction MUST be true. • If two instances of ots-impairments parameters are present, the instance with ingress-direction true applies to the CEP sink/ingress direction. The other instance MUST have ingress-direction false, since applies to the CEP source/egress direction. For unidirectional CEPs, • At most one instance MUST be present (it is expected that the remote CEP contains the instance if this CEP does not). This attribute MUST match the direction of the CEP (true for CEPs with SINK direction and false with SOURCE direction) • The impairment-route-entry list is a sequence, so each element is either a concentrated loss or an ots-fiber-span-impairments structure. NOTE (*): For *ots-fiber-span-impairements,* a single span entry MAY list a total-loss value or decompose into loss-coeff, connector-in, connector-out NOTE (**): The usage of *physical-context/tapi-equipment:physical-span/ abstract-strand* to support physical impairments data will be addressed in a future version.

Table 45: mc-connection-end-point-spec (MC CEP), oms-connection-end-point-spec (OMS CEP), ots-media-connection-end-point-spec (OTS_MEDIA CEP) spectrum and power management object definition(s)

mc-connection-end-point-spec, oms-connection-end-point-spec, ots-media-connection-end-point-spec	**/tapi-common:context/tapi-topology:topology-context/topology/node/owned-node-edge-point/tapi-connectivity:cep-list/connection-end-point/tapi-photonic-media:mc-connection-end-point-spec, oms-connection-end-point-spec, ots-media-connection-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**

spectrum-pac	Includes { **occupied-spectrum** with upper-frequency, lower-frequency local-id (local identifier) name (name value pairs) }	RO	C	• Provided by *tapi-server* • For OTSiMC CEPs, this MAY be present in case the CEP is terminated and MUST be present if the CEP is not terminated. • For MC, OMS and OTS_MEDIA, this MUST be present • For OMS and OTS_MEDIA the CEPs include a list of spectrum pac • The **frequencies** are specified in Hz.
power-measurement-pac	Includes { measured-input-power and measured-output-power } both with total-power and power-spectral-density	RO	C	• Provided by *tapi-server* • *Depends on hw power monitoring capabilities*

Table 46: Route object definition

route	**/tapi-common:context/tapi-connectivity:connection/route**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
local-id	"[0-9a-zA-Z_]{32}"	RO	M	• Provided by *tapi-server*
name	MUST include "value-name": "ROUTE_NAME" "value": " [0-9a-zA-Z_]{64}"	RO	M	• Provided by *tapi-server*
resilience-route	Including: route-state (e.g., CURRENT, NOT_CURRENT, UNKNOWN) priority (uint64)	RO	M	• Provided by *tapi-server* 0 (zero) means "unspecified". 1 is preferred/main/intended is the highest priority .2 has lower priority than 1, 3 has lower priority than 2, etc.
connection-end-point	List of {"connection-end-point-ref - */tapi-common:context/tapi-topology:topology-context/topology/node/owned-node-edge-point/tapi-connectivity:cep-list/connection-end-point/uuid* "}	RO	M	• Provided by *tapi-server*

6.2.3.2 Expected results

The state of the network after the successful provisioning of a connectivity service is detailed in Section 6.2.2.

6.2.4 Use case 1a: Unconstrained DSR Service Provisioning (=<100G).

Number	**UC1a**
Name	**Unconstrained DSR Service Provisioning (=<100G).**
Technologies involved	Optical, DSR
Process/Areas Involved	Planning and Operations

Brief description	The UC1a describes the provisioning of a ***tapi-connectivity:connectivity-service*** instance between SIPs exposed by the TAPI-Server at the DSR networking layer. The underlying connection provisioning and management (including lower layer connections e.g., ODU, OTU, OTSiMC, MC and intermediate regeneration connections if needed) is performed by the SDN Domain controller. The routes of all lower layer top-connections (e.g., ODU or OTSiMC) across the network topology are calculated by the controller, and the connections automatically provisioned as necessary. The TAPI-Client is not providing technology specific Traffic-Engineering constraints.
Layers involved	DSR/DIGITAL_OTN/PHOTONIC_MEDIA
Type	Provisioning
Description & Workflow	This UC is implemented following the same workflow described in “Description & Workflow” of UC1.0

6.2.4.1 Examples of Time Zero Scenarios

For this UC the “time zero scenarios” (previous to the provisioning of the DSR service) of Figure 6-54, Figure 6-55, Figure 6-56 apply.

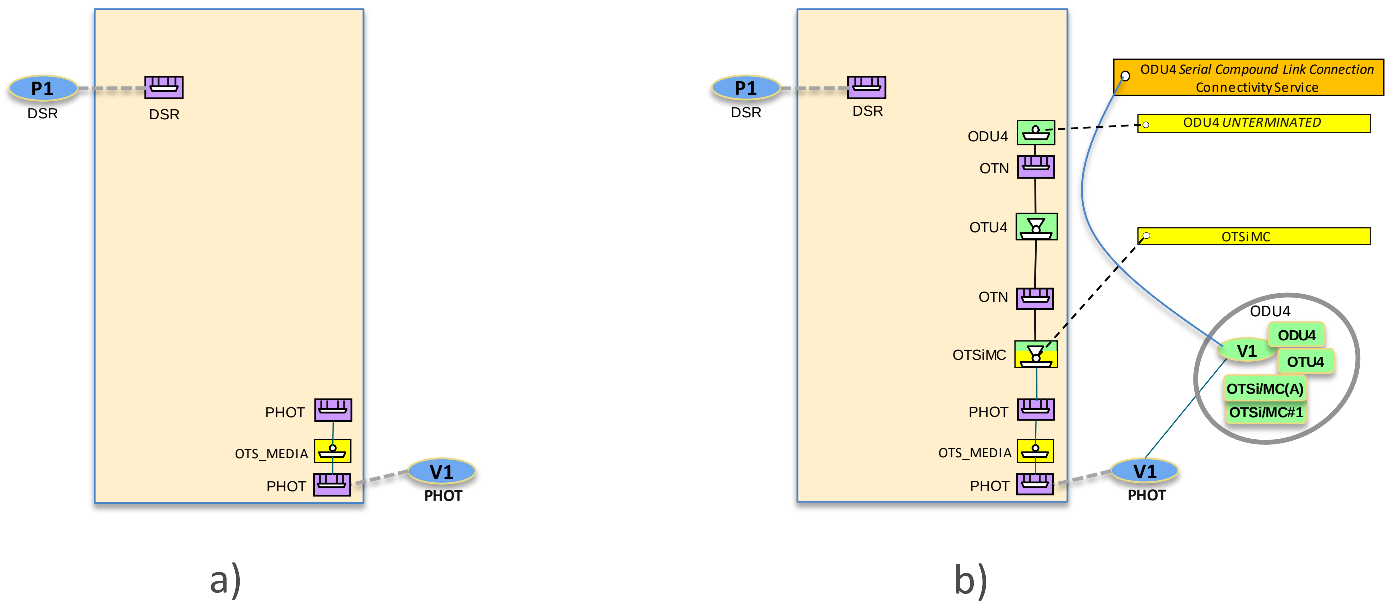
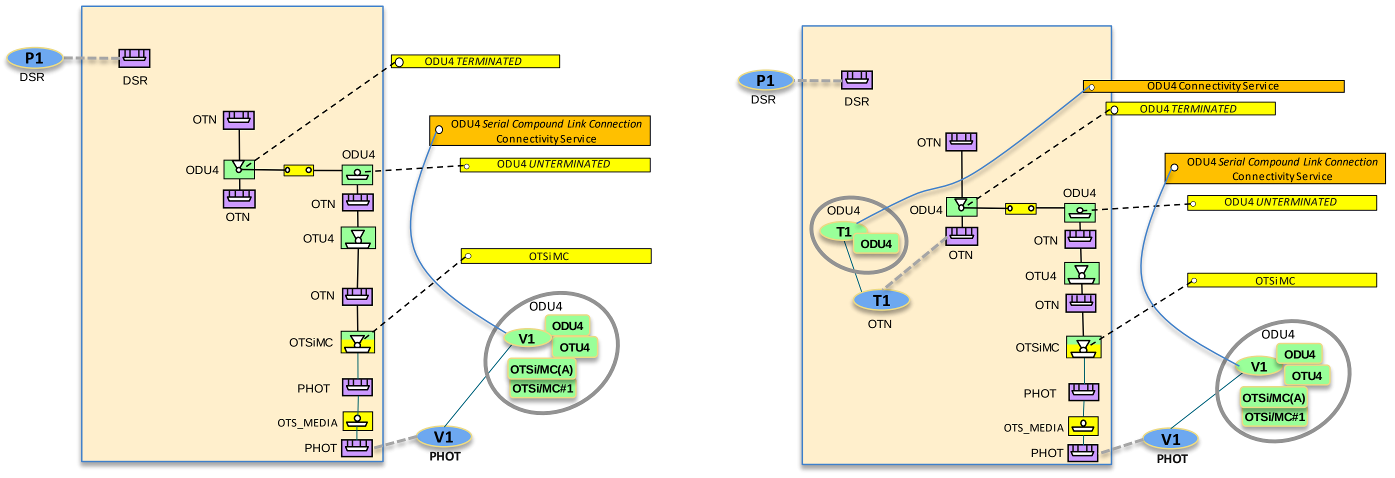


Figure 6-54 a) No server connections, b) Server ODU SCLC Connectivity Service



a) b)

Figure 6-55 a) Server ODU SCLC CS and HO ODU connection, b) Server ODU SCLC CS and HO ODU CS

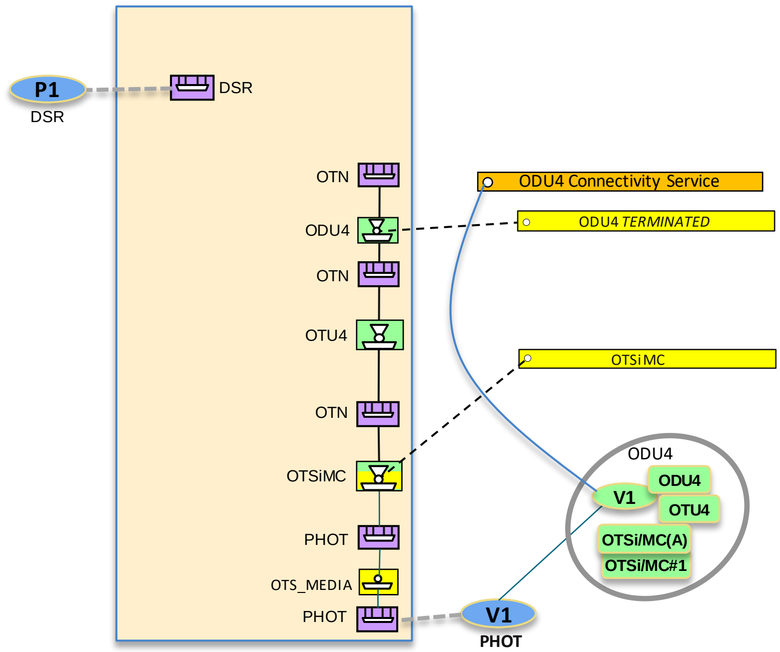


Figure 6-56 Server ODU CS, HO ODU always terminated

6.2.4.2 Applicable Provisioning Scenarios

For this UC the following provisioning scenarios apply, with no constraints on OTN layers:

- Figure 6-9 DSR/ODUk Connectivity Service on ODUk SCLC CS
- Figure 6-10 DSR/ODUj CS on ODUk SCLC CS, ODUk Terminated Connection automatically created or reused
- Figure 6-11 DSR/ODU2 CS on ODU3 SCLC CS, ODU3 Terminated Connection automatically created or reused
- Figure 6-12 DSR/ODUj CS on ODUk SCLC CS, ODUk Term. Conn. autom. created or reused, no ODUj flexibility
- Figure 6-13 DSR/ODUj Connectivity Service on ODUk SCLC CS, auto creation of ODUk CS
- Figure 6-15 DSR/ODUj Connectivity Service on ODUk CS on ODUk SCLC CS

- Figure 6-40 DSR/ODUj CS on ODUk CS
- Figure 6-41 DSR CS on ODUj on ODUk CS (DSR flexibility)

6.2.4.3 Relevant Parameters

Table 47: Connectivity-service (**CS**) object definition (DSR UC1a)

connectivity-service	**/tapi-common:context/tapi-connectivity:connectivity-context/tapi-connectivity:connectivity-service**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
connectivity-constraint /requested-capacity/total-size	• "value": real, • "unit": *see tapi-common:capacity-unit*	RW	C	• Provided by *tapi-client* • This parameter MUST be present if the layer-protocol-qualifier is LAYER_PROTOCOL_QUALIFIER_UNSPECIFIED
direction	"BIDIRECTIONAL"	RW	M	• Provided by *tapi-client* • *This UC only considers BIDIRECTIONAL DSR services.*
connectivity-constraint/service-type	"POINT_TO_POINT_CONNECTIVITY"	RW	M	• Provided by *tapi-client*
layer-protocol-name	"DSR"	RW	M	• Provided by *tapi-client*
layer-protocol-qualifier	Any of the DSR DIGITAL_SIGNAL_TYPE qualifiers.	RW	M	• Provided by *tapi-client* • Support based depending on hardware capabilities

Table 48: Connectivity-service-end-point (**CSEP**) object definition (DSR UC1a)

connectivity-service-end-point	**/tapi-common:context/tapi-connectivity:connectivity-context/tapi-connectivity:connectivity-service/tapi-connectivity:end-point**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
direction	"BIDIRECTIONAL"	RW	O	• Provided by *tapi-client* • If not specified, the default intended value is BIDIRECTIONAL

6.2.5 Use Case 1b: Unconstrained DSR Service Provisioning (Beyond 100G).

Number	**UC1b**
Name	**Unconstrained DSR Service Provisioning multi wavelength (beyond 100G).**
Technologies involved	Optical, DSR
Process/Areas Involved	Planning and Operations
Brief description	This UC follows UC1a, but with the difference that the service relies on an ODU-Cn/OTU-Cn top-level connection(s) which, in turn, is realized by one or more OTSiMC connections.
Layers involved	DSR/DIGITAL_OTN/PHOTONIC_MEDIA

Type	Provisioning
Description & Workflow	This UC is implemented following the same workflow described in “Description & Workflow” of UC1.0

6.2.5.1 Examples of Time Zero Scenarios

For this UC the “time zero scenarios” (previous to the provisioning of the DSR service) of Figure 6-57 and Figure 6-58 apply.

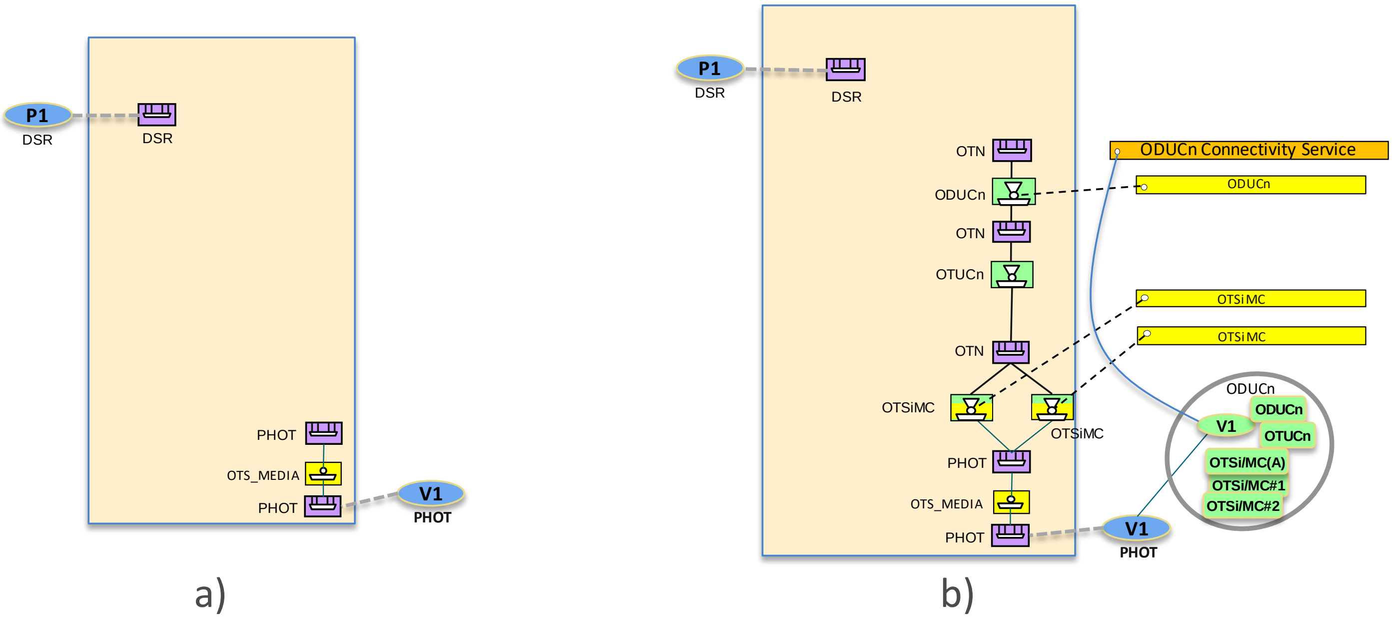


Figure 6-57 a) No server connections, b) Server ODUCn Connectivity Service

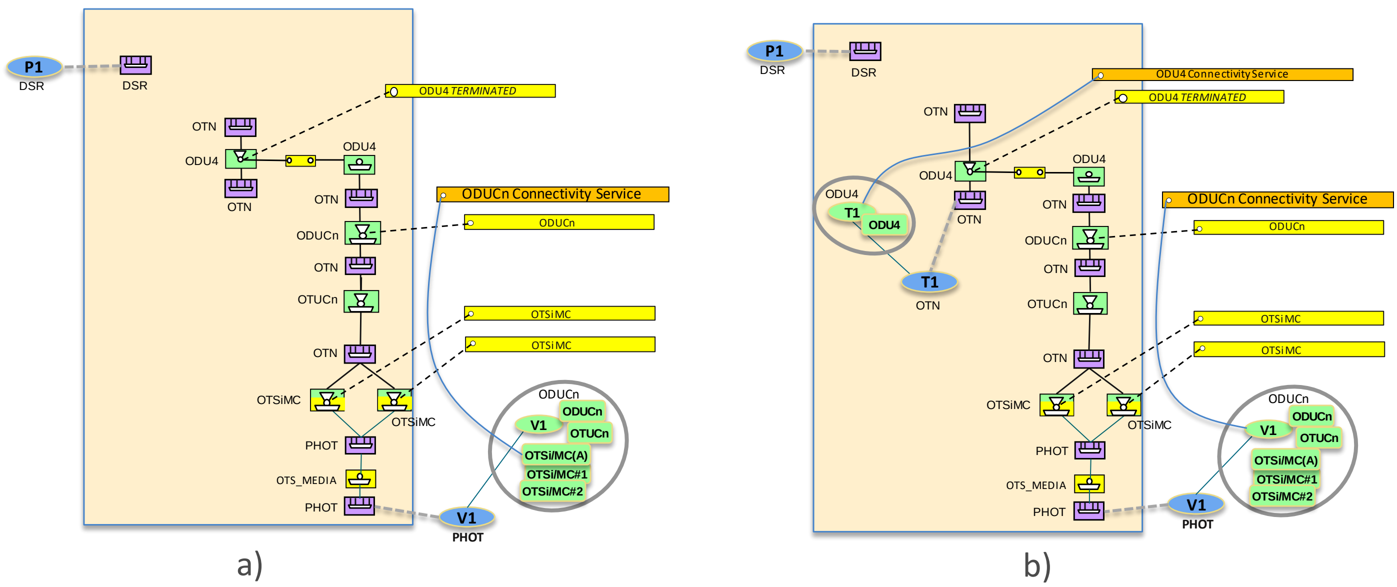


Figure 6-58 a) Server ODUCn CS and HO ODU connection, b) Server ODUCn CS and HO ODU CS

6.2.5.2 Applicable Provisioning Scenarios

For this UC the following provisioning scenarios apply, with no constraints on OTN layers:

- Figure 6-34 DSR/ODUFlex Connectivity Service on ODUCn CS
- Figure 6-35 DSR/ODUj CS on ODUk on ODUCn CS - ODUk Terminated Connection automatically created or reused
- Figure 6-36 DSR/ODUj CS on ODUk CS on ODUCn CS - Auto creation of ODUk CS
- Figure 6-38 DSR/ODUj CS on ODUk CS on ODUCn CS

6.2.6 Use case 1c: DSR over ODU Service Provisioning

Number	**UC1c**
Name	**DSR over ODU service provisioning**
Technologies involved	Optical, ODU, DSR layers
Process/Areas Involved	Planning and Operations
Brief description	The UC1c describes the provisioning of a TAPI connectivity-service instance between DSR SIPs, e.g., between transceiver client ports, including the mapping and or multiplexing of such client signal into the line G.709 OTN frame. Both UC1c and UC2b aim at enabling the provisioning of a DSR over ODU. The DSR signal is encapsulated either in a lower order ODU (which in turn is encapsulated in a high-order ODU, *ODUk slot selection is covered in UC2b*) or in a high-order ODU. This UC MAY require the prior provisioning of transponder-to-transponder connectivity.
Layers involved	DSR/DIGITAL_OTN/PHOTONIC_MEDIA
Type	Provisioning
Description & Workflow	This UC is implemented following the same workflow described in “Description & Workflow” of UC1.0 with **server restrictions**.

6.2.6.1 Examples of Time Zero Scenarios

For this UC the “time zero scenarios” are the same as UC1a and UC1b.

6.2.6.2 Applicable Provisioning Scenarios

For this UC the provisioning scenarios of UC1a and UC1b apply, with the UC’s specific constraints on OTN layers.

6.2.6.3 Detailed Workflow

Note that this Use Case assumes that the ODU TTP is configurable (otherwise, this UC reduces to UC1a and UC1b). Two cases are considered: **Case I** (mapping) the ODU container is directly carried by an OTU container or **Case II** (multiplexing) the ODU container is carried by a server layer ODU container object.

This UC is illustrated in, for example, Figure 6-9 -- DSR/ODUk Connectivity Service (mapping) -- and Figure 6-10 (multiplexing). Let's consider for mapping (100GE over ODU4) and for multiplexing 10G over ODU2 over ODU4).

In the mapping case it is possible to specify: i) the odu-rate – for ODuflex -- and ii) the mapping type and client type (odu-csep-ttp-pac with configured-mapping-type and configured-client-type). In the case of multiplexing, it is also possible to specify iii) the tributary slot size - when applicable -. This is encoded as Layer Protocol Constraints of the qualifier that is directly carrying the DSR signal. Note that the selection of tributary port number and slot list is done in UC2b.

6.2.6.4 Relevant Parameters

The workflow includes the inclusion of **ODU layer protocol constraint(s)** in the CSEP (tapi-connectivity:connectivity-service/end-point/layer-protocol-constraint/tapi-digital-otn:odu-connectivity-service-end-point-spec).

odu-connectivity-service-end-point-spec	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/end-point/layer-protocol-constraint/tapi-digital-otn:odu-connectivity-service-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
odu-csep-common-pac	Includes: odu-rate in kb/s, opu-tributary-slot-size, one of 1G25 or 2G5. See yang for details.	RW	M	• Provided by *tapi-client*
odu-csep-ttp-pac	Includes: configured-mapping-type configured-client-type	RW	C	• Provided by *tapi-client* The *configured mapping type* is mandatory if there are several mapping types available for the DSR service. The *configured client type* is optional if this layer protocol constraint is used while provisioning the client. The *configured client type* could be used when provisioning ODU services in a bottom-up approach (use case not covered in this RIA).
odu-cn-csep-ttp-pac	Includes number-of-odu-c	RW	C	• Provided by *tapi-server* • *Used in ODU-Cn configurations.*

6.2.6.5 Expected results

Upon instantiation, the ODU TTP CEP(s) MUST include the **tapi-digital-otn:odu-connection-end-point-spec** augment, including the **odu-common** and **odu-term-and-adapter** (with the configured-client-type and mapping-type), along with the rest of parameters presented in UC1.0.

It is assumed that the server ODUk (or ODUCn) connectivity is directly provisioned by the SDN controller, configuring an ODUk (or ODUCn) connection between the transponder line ports thus an instance of the ODU Connectivity Service for the ODUk (or ODUCn) is not required. Upon instantiation a TTP ODU CEP representing the ODUk (or ODUCn) connection MUST be instantiated over the ODU NEP. In the case of ODU-Cn, the **odu-cn-effective-time-slot-list** MUST list the ODU-Cn 5GHz available slots.

Note that the aforementioned figures show the "no flexibility at the DSR layer" option for UC1a-UC1c (as detailed in DSR UNI and OTN ENNI considerations) but it does not exclude other options where e.g., the flexibility at the DSR layer is shown explicitly.

6.2.7 Use case 1d: DIGITAL_OTN with PHOTONIC_MEDIA/OTSi Service Provisioning

Number	**UC1d**
Name	**DIGITAL_OTN with PHOTONIC_MEDIA/OTSi Service Provisioning**
Technologies involved	Optical
Process/Areas Involved	Planning and Operations
Brief description	This case is currently formulated as a specific case of UC-1e (with the number of OTSi being 1)
Layers involved	PHOTONIC_MEDIA
Type	Provisioning
Description & Workflow	This UC is implemented following the same workflow described in “Description & Workflow” of UC1.0 with **specific layer protocol constraints**.

6.2.7.1 Examples of Time Zero Scenarios

For this UC the “time zero scenario” (previous to the provisioning of the ODU/OTU/OTSi service) of Figure 6-59 applies.

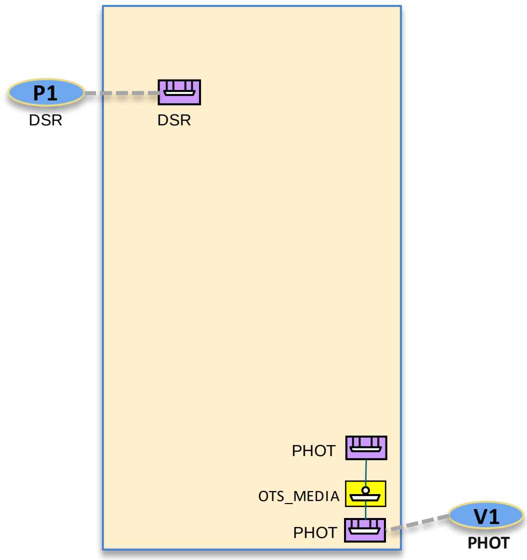


Figure 6-59 No server connections

6.2.7.2 Applicable Provisioning Scenarios

For this UC the following provisioning scenarios apply, with the UC’s specific constraints on OTN and OTSiMC layers:

- Figure 6-8 ODUk Serial Compound Link Connection Connectivity Service

- Figure 6-33 ODUCn Connectivity Service
- Figure 6-39 ODUk Trail Connectivity Service

6.2.7.3 Detailed Workflow

This UC corresponds to the "transponder to transponder" network scenarios for provisioning. Note that UC2a allows channel selection.

6.2.7.4 Relevant Parameters

The workflow potentially requires the inclusion of ODU, OTU, and OTSiMC layer protocol constraint(s) in the CSEP.

6.2.8 Use case 1e: DIGITAL_OTN with PHOTONIC_MEDIA/OTSiA Service Provisioning

Number	**UC1e**
Name	**DIGITAL_OTN with PHOTONIC_MEDIA/OTSiA Service Provisioning**
Technologies involved	Optical
Process/Areas Involved	Planning and Operations
Brief description	UC1e describes the provisioning of a ***tapi-connectivity:connectivity-service*** instance between service-interface-points exposed by the TAPI-Server at the PHOTONIC_MEDIA networking layer supporting the provisioning of ODU/OTU services. The TAPI-Client is not providing any constraints regarding optical-spectrum selection for the OTSiMC connections. While this service can include intermediate regeneration, if necessary, this use case only addresses OTSi(A) attributes at the first and last optical segments.
Layers involved	PHOTONIC_MEDIA
Type	Provisioning
Description & Workflow	This UC is implemented following the same workflow described in "Description & Workflow" of UC1.0

6.2.8.1 Examples of Time Zero Scenarios

For this UC the "time zero scenario" is the same as UC1d.

6.2.8.2 Applicable Provisioning Scenarios

For this UC the provisioning scenarios of UC1d apply, with the UC's specific constraints on OTN and OTSiMC layers.

6.2.8.3 Detailed Workflow

This UC corresponds to the "transponder to transponder" network scenarios for provisioning. Note that UC2a allows channel selection. This UC assumes that the transceiver line port NEP (thus SIP) listed transceiver profiles provide enough information to deduce the supportable / applicable combinations in terms of OTSi.

6.2.8.4 Relevant Parameters

This UC focuses on the selection of the number of OTSi components. UC1d assumes N=1

otsia-connectivity-service-end-point-spec	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/end-point/layer-protocol-constraint/tapi-photonic-media:otsia-connectivity-service-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
number-of-otsi	Number of component OTSi. Must be N >= 1	RW	C	• Provided by *tapi-client* • This RIA only considers an OTSiA provisioning from a single SIP (e.g., single transceiver line port). • This is based on hw capabilities.

6.2.8.5 Expected results

This case requires the generation of N number of OTSiMC Top Connections required to transport the service.

6.2.9 Use case 1e.1: DSR with PHOTONIC_MEDIA/OTSiA Service Provisioning

The use case related to provisioning of DSR services directly over OTSiA (thus no DIGITAL_OTN) is left for a further version of this specification.

6.2.10 Use case 1f: PHOTONIC_MEDIA/Media Channel(s) (MC/MCG) Service Provisioning

Number	**UC1f**
Name	**PHOTONIC_MEDIA/Media Channel(s) (MC/MCG) Service Provisioning**
Technologies involved	Optical
Process/Areas Involved	Planning and Operations
Brief description	The UC1f describes the provisioning of a MC (group) ***tapi-connectivity:connectivity-service***. **This service does not cover intermediate regeneration.** This use case is intended to define the way the TAPI Client can request the creation of a media-channel service which reserves a portion of optical spectrum across the PHOTONIC_MEDIA layer. Each MC may be wider than the OTSi(A) occupied spectrum (for example, due to guard bands). Multiple OTSi signals MAY be included in a MC. The TAPI-Client is not providing constrains regarding spectrum-band selection for the MC connections.
Layers involved	PHOTONIC_MEDIA
Type	Provisioning
Description & Workflow	This UC is implemented following the same workflow described in "Description & Workflow" of UC1.0

6.2.10.1 Examples of Time Zero Scenarios

For this UC the "time zero scenarios" (previous to the provisioning of the MC/MCG service) of Figure 6-60 apply.

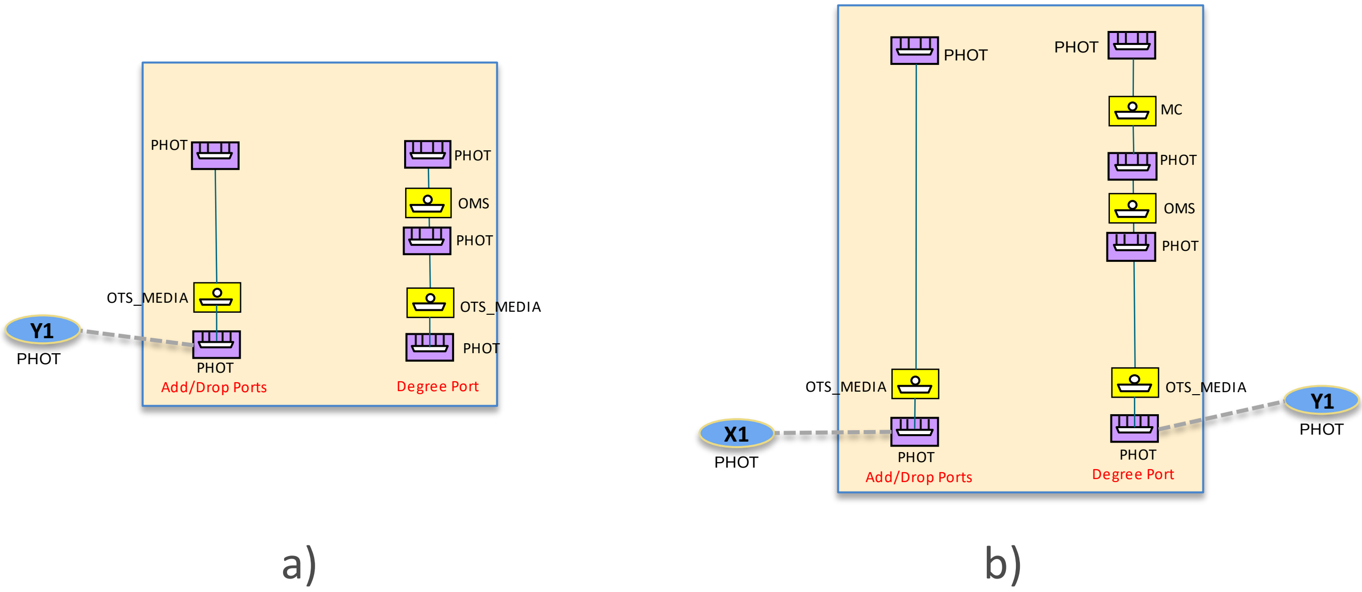


Figure 6-60 a) MC CS at Add/Drop side, b) MC CS at Degree side (Y1 SIP)

6.2.10.2 Applicable Provisioning Scenarios

For this UC the following provisioning scenarios apply, with the UC's specific constraints on MC layer:

- Figure 6-42 MC Connectivity Service at Add/Drop side
- Figure 6-47 MC Connectivity Service at Degree side

6.2.10.3 Relevant Parameters

The following MC CSEP parameters are required in case the request is for a group (with N > 1). For the case N=1 bandwidth configuration can be specified using the CSEP "capacity" (unit/value). Note that UC2c allows spectrum selection.

mcg-connectivity-service-end-point-spec	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/end-point/layer-protocol-constraint/tapi-photonic-media:mcg-connectivity-service-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
number-of-mc	Number of component MC. Must be >= 1	RW	M	• Provided by *tapi-client* • *This RIA only considers an MCG provisioning from a single SIP (e.g, single add /drop port).*
mc-bandwidth-config-pac	List of *MC Bandwidth Configurations*, indexed by local-id. Each element contains: local-id and name list. **spectrum-bandwidth** (in Hz)	RW	C	• Provided by *tapi-client* • Mandatory for N > 1

6.2.10.4 Expected results

MC CEP parameters are provided UC1.0. Note that this RIA only covers the establishment of bidirectional MC connectivity services. This use case accepts different variations according to the model directionality chosen to represent the PHOTONIC_MEDIA layer. The currently agreed solutions are three:

1. Full-bidirectional - UNI and PHOTONIC_MEDIA model.
2. Mixed-scenario - UNI bidirectional and topology unidirectional.
3. Full-unidirectional OLS scenario - UNI and PHOTONIC_MEDIA unidirectional

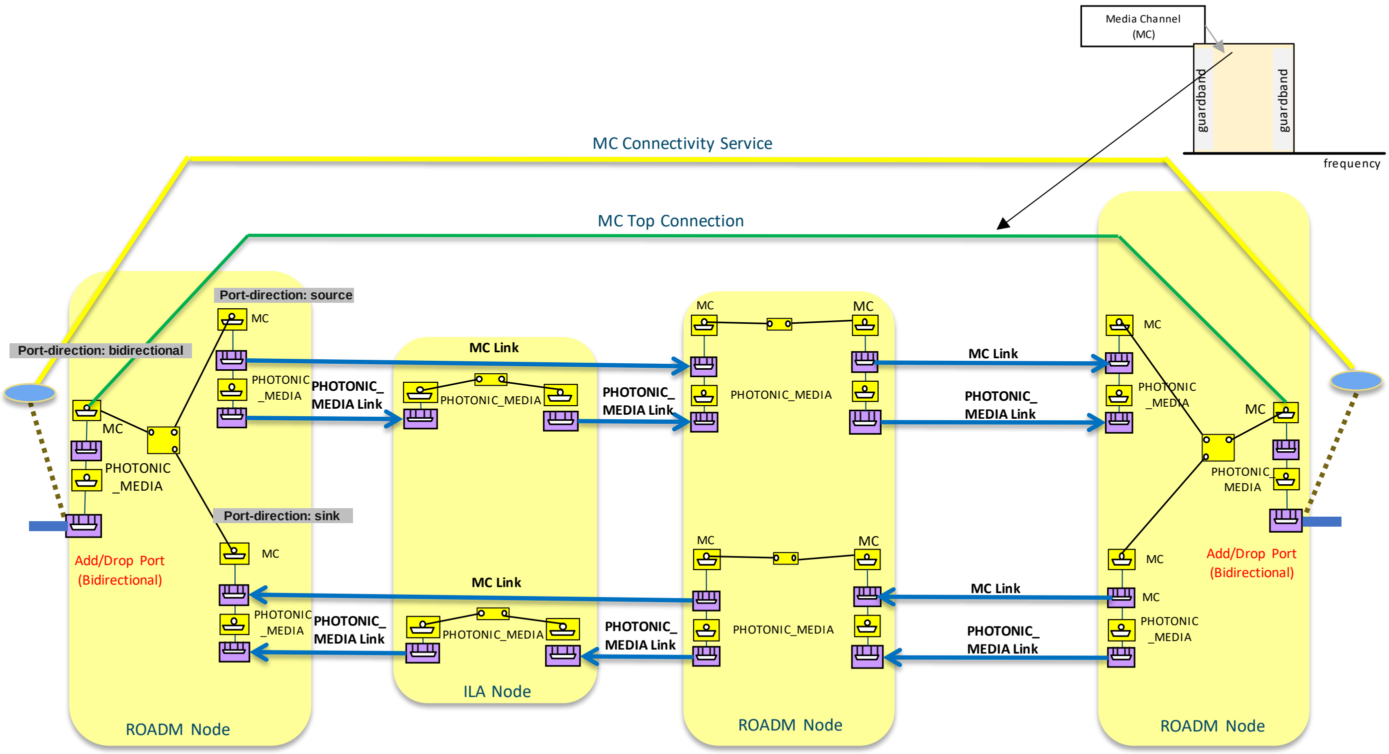
Model/Solution 1 is aligned with the assumptions defined in this RIA. The next sections detail models 2 and 3

6.2.10.4.1 Model 2: Mixed Scenario - UNI bidirectional and OMS unidirectional

The second alternative corresponds to a mixed solution exposed by the TAPI server where the relation between the Add/Drop directions of UNI interfaces is known by the TAPI server and thus, the MC UNI interfaces are represented as bidirectional SIPs associated to the Add/Drop PHOTONIC_MEDIA NEPs, however, the PHOTONIC_MEDIA layer is abstracted as a unidirectional link topology.

The MC Connectivity-service is modeled as bidirectional, with two references to the bidirectional Add/Drop SIPs. Once successfully provisioned, the Connectivity-Service MUST reference *a single bidirectional Top Connection* representing the end-to-end route across the PHOTONIC_MEDIA layer.

The MC Top Connection includes, within the ***tapi-connectivity:lower-connection*** attribute, the references both three-ended Cross-Connections (XCs) connecting the bidirectional Add/Drop UNI interfaces to the ROADM degree unidirectional interfaces. Then the route traverses the remaining unidirectional PHOTONIC_MEDIA nodes till the far end. All unidirectional XCs in the two directions MUST be included into the MC Top Connection lower-level connection list.



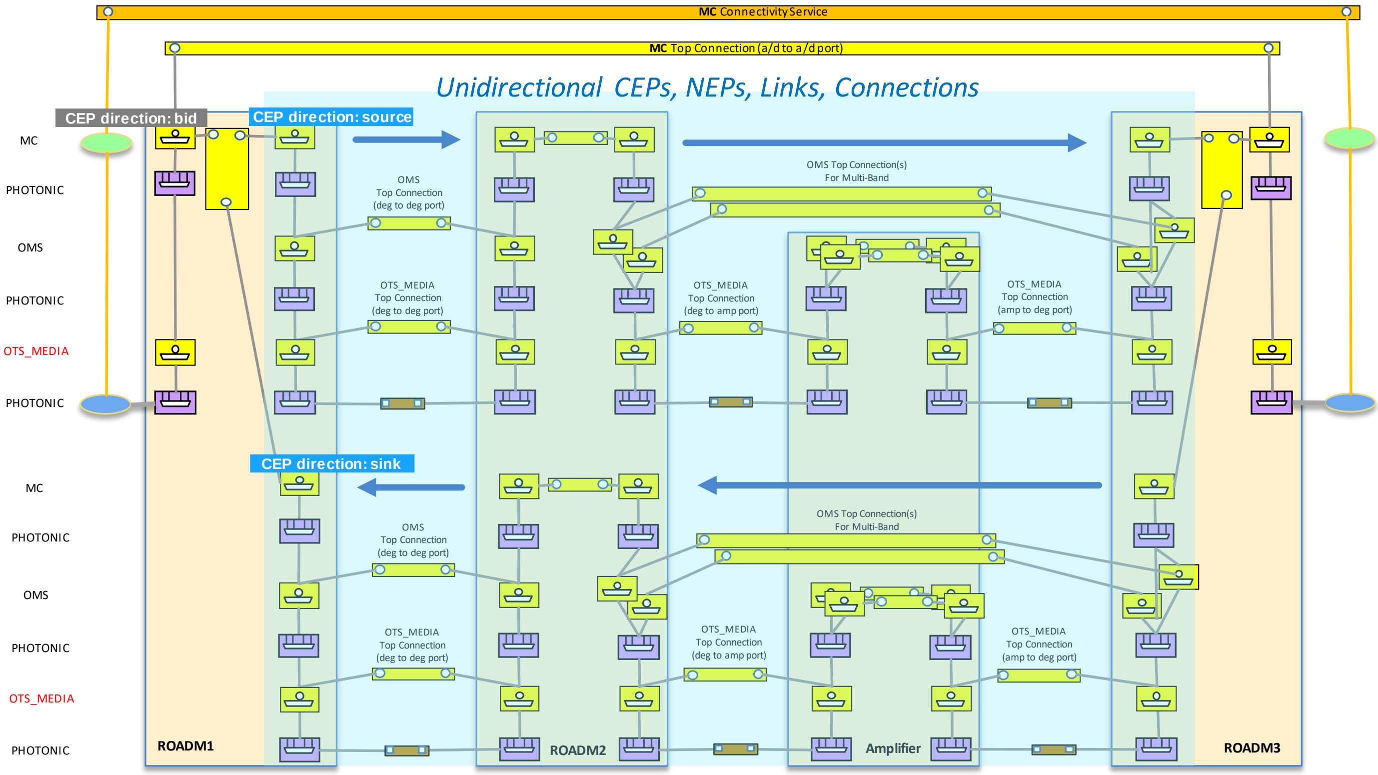


Figure 6-61 Mixed Scenario - UNI bidirectional and OMS unidirectional.

6.2.10.4.2 Model 3: Full-unidirectional OLS scenario - UNI and PHOTONIC_MEDIA unidirectional

In this scenario, *either* there are unidirectional relationships between ROADM Add/Drop ports and the transceiver line ports (UC not described in this RIA, since transceivers line ports are bidirectional) *or* the transceivers are not managed/controlled by the TAPI server.

In this modelling approach the MC UNI interfaces are represented as unidirectional SIPs associated to unidirectional Add/Drop NEPs.

To support *bidirectional* MC Connectivity-services four CSEPs are required (each referring to a unidirectional SIP). Once successfully provisioned, the Connectivity-Service MUST reference two unidirectional Top Connections representing the two end-to-end route directions across the PHOTONIC_MEDIA layer. Note that this is an exception to the common guideline of having only a single immediate top-connections.

Moreover, the MC Top Connections include within the ***tapi-connectivity:lower-connection*** attribute, the reference to the unidirectional Cross-Connections (XCs) between the PHOTONIC_LAYER_QUALIFIER_MC unidirectional CEPs over the unidirectional MC NEPs.

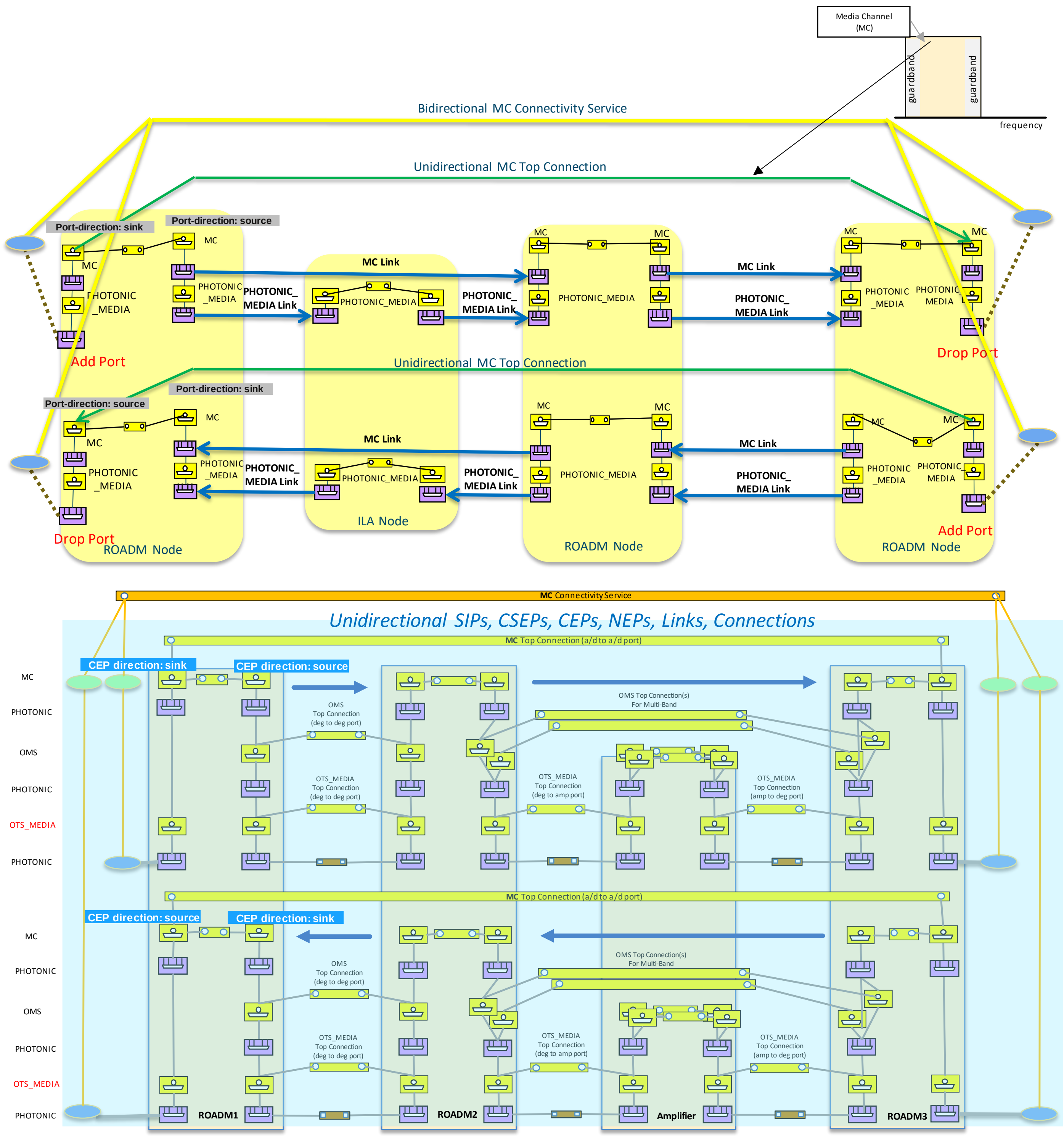
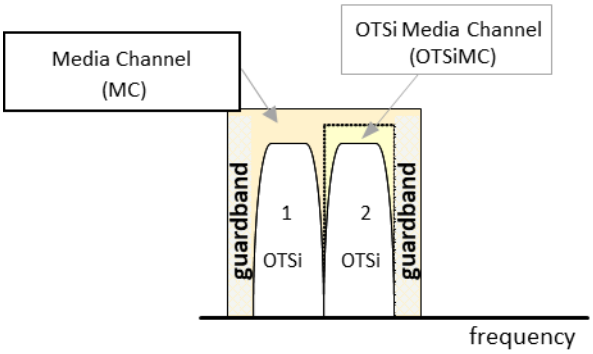


Figure 6-62 Full Unidirectional - UNI and OMS unidirectional scenario.

6.2.11 Use case 1g: PHOTONIC_MEDIA/OTSiMC (with optional MC) Service Provisioning

Number	**UC1g**
Name	**PHOTONIC_MEDIA/OTSiMC (with optional MC) Service Provisioning**

Technologies involved	Optical
Process/Areas Involved	Planning and Operations
Brief description	This use case builds on UC1f with additional information about the specific channels occupied by the OTSi signals. This UC adds server layer restrictions. The graphical representation of the relationship between MC, OTSIMC and OTSI signal is: 
Layers involved	PHOTONIC_MEDIA
Type	Provisioning
Description & Workflow	This UC is implemented following the same workflow described in “Description & Workflow” of UC1.0 with [**server restrictions**] This UC potentially uses Layer Protocol Constraints for the OTSiMC and MC layer protocol qualifiers.

6.2.11.1 Examples of Time Zero Scenarios

For this UC the “time zero scenarios” (previous to the provisioning of the OTSiMC service) of Figure 6-63, Figure 6-64, Figure 6-65 apply.

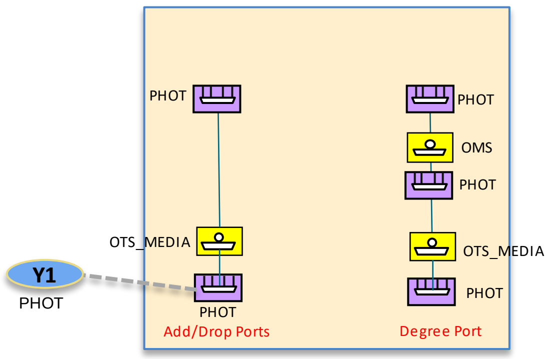


Figure 6-63 No “server” connections (auto creation of MC Conn/CS or no MC layer supported)

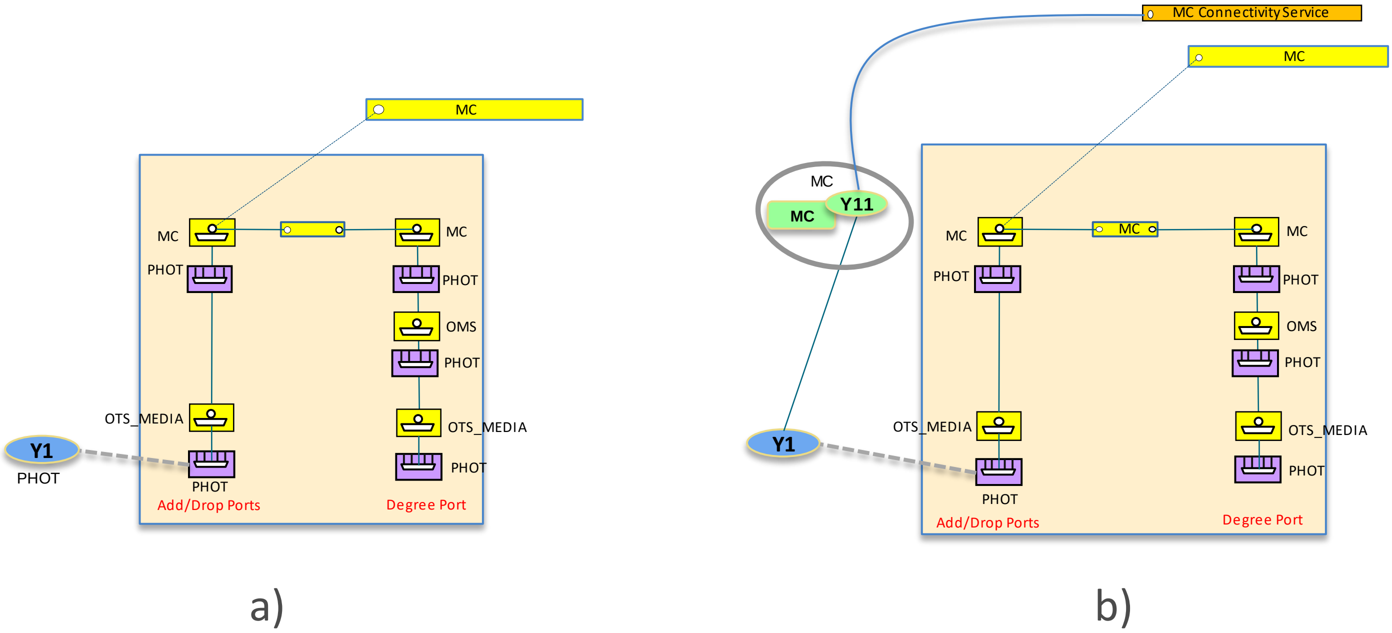


Figure 6-64 a) “Server” MC Connection, b) “Server” MC Connectivity Service

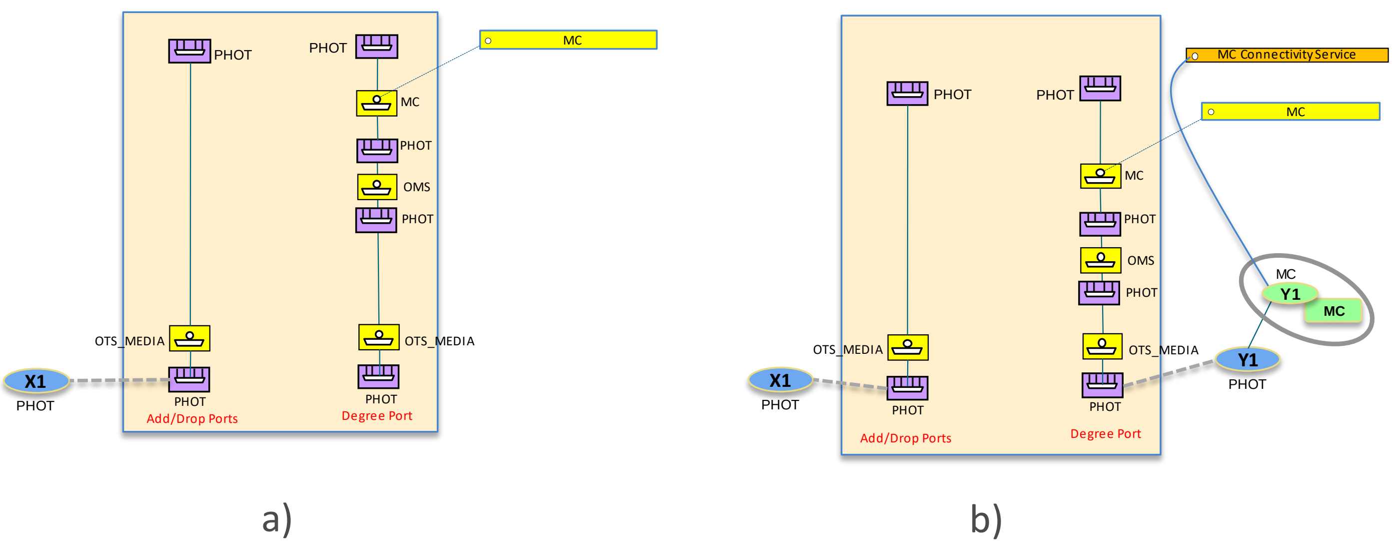


Figure 6-65 a) “Server” MC Connection at degree side, b) “Server” MC Connectivity Service at degree side

6.2.11.2 Applicable Provisioning Scenarios

For this UC the following provisioning scenarios apply, with the UC’s specific constraints on MC and OTSiMC layers:

- Figure 6-44 OTSiMCG CS on MC at Add/Drop side, MC Connection automatically created or reused
- Figure 6-45 OTSiMCG CS on MC CS at Add/Drop side, auto creation of MC CS
- Figure 6-46 OTSiMCG CS on MC CS at Add/Drop side

- Figure 6-49 OTSiMC(G) CS on MC at Degree side, MC Connection automatically created or reused
- Figure 6-50 OTSiMCG CS on MC CS at Degree side, auto creation of MC CS
- Figure 6-51 OTSiMC(G) CS on MC CS at Degree side

6.2.11.3 Relevant Parameters

The CS and its CSEPs have Layer Protocol Qualifier OTSiMC. Each CSEP includes (up to two) layer protocol constraints, including the *otsi-mcg-connectivity-service-end-point-spec* and the *mcg-connectivity-service-end-point-spec* respectively.

The following CSEP parameters are required in case the request is for a group of OTSiMC (with N > 1). For the case N=1 bandwidth configuration can be specified using the CSEP OTSiMC "capacity" (unit/value).

This UC focuses on the case that number-of-mc is 1. If specified, the bandwidth of the MC MUST be greater than the referred OTSiMC bandwidths. For more complex scenarios, this RIA recommends UC2c to avoid ambiguity in spectrum assignments between OTSiMC and MC.

otsi-mcg-connectivity-service-end-point-spec	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/end-point/layer-protocol-constraint/tapi-photonic-media:otsi-mcg-connectivity-service-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
number-of-otsi-mc	Number of components OTSi-MC. Must be >= 1	RW	M	• Provided by *tapi-client*
otsi-mc-bandwidth-config-pac	List of MC Bandwidth Configurations, indexed by local-id. Each element contains: local-id and name list. **spectrum-bandwidth** in Hz	RW	C	• Provided by *tapi-client*

mcg-connectivity-service-end-point-spec	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/end-point/layer-protocol-constraint/tapi-photonic-media:mcg-connectivity-service-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
number-of-mc	Number of component MC. Must be >= 1	RW	M	• Provided by *tapi-client* • *This RIA only considers an MCG provisioning from a single SIP (e.g, single add /drop port).*
mc-bandwidth-config-pac	List of *MC Bandwidth Configurations*, indexed by local-id. Each element contains: local-id and name list. **spectrum-bandwidth (**in Hz)	RW	C	• Provided by *tapi-client* • If this is not provided by the client, implementations are free to select the most appropriate bandwidth.

6.2.11.4 Expected results

For the expected results for this UC see the applicable provisioning scenarios.

6.2.12 Use case 1h: Asymmetric DSR Service Provisioning, DSR UNI to OTUk E-NNI grey interface.

Number	**UC1h**
Name	Asymmetric DSR Service Provisioning, DSR UNI to OTUk E-NNI grey interface.
Technologies involved	Optical, ODU, DSR layers
Process/Areas Involved	Planning and Operations
Brief description	This use case is intended to define the way the TAPI Client can request the creation of a *DSR connectivity-service* between UNI and E-NNI SIPs (see Section 5.2.3 for considerations of UNI and UNNI modelling aspects). The intention is to establish services which start in one network domain and handover to another network domain managed by a different (TAPI) Server. **UNI:** The corresponding UNI CSEP refers to a DSR SIP. **NNI:** The corresponding NNI CSEP refers to a DIGITAL_OTN SIP The underlying connection provisioning and management and the path of each lower layer connection, is calculated by the controller and the connection automatically provisioned, as described in the UC1a.
Layers involved	DSR/DIGITAL_OTN
Type	Provisioning
Description & Workflow	This UC is implemented following the same workflow described in "Description & Workflow" of UC1.0 and UC1a

6.2.12.1 Examples of Time Zero Scenarios

For this UC the "time zero scenarios" (previous to the provisioning of the DSR service) are the same of UC1a and UC1b on DSR UNI side. On OTU NNI side see the figures below. Note that in case the *transponder-to-transponder* connectivity is based on ODUCn container, then same scenarios apply, replacing the ODUk *Serial Compound Link Connection* CS with the ODUCn *Trail* CS.

Figure 6-66 and Figure 6-67 apply to Asymmetric Scenario 1: Handoff at ODU4 Layer, no ODU2 layer on ENNI.

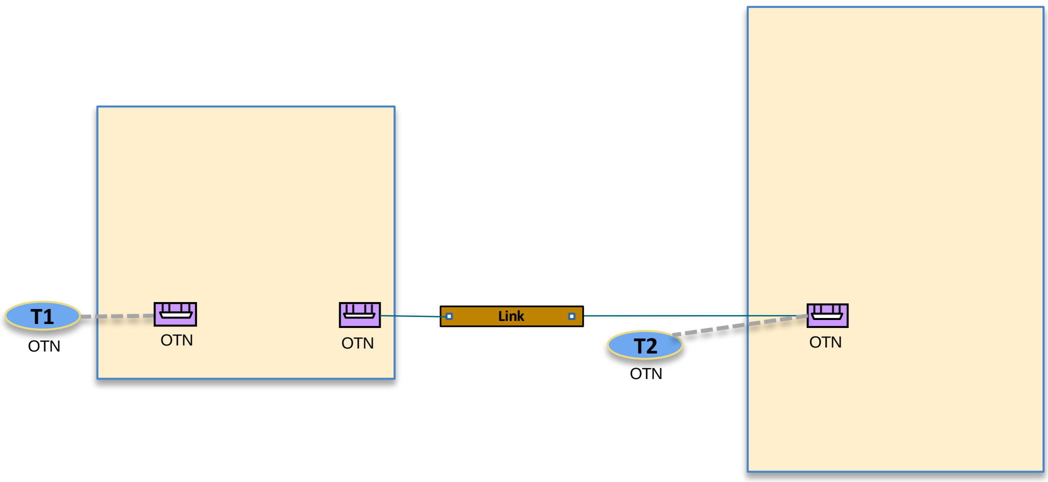


Figure 6-66 No “server” connections

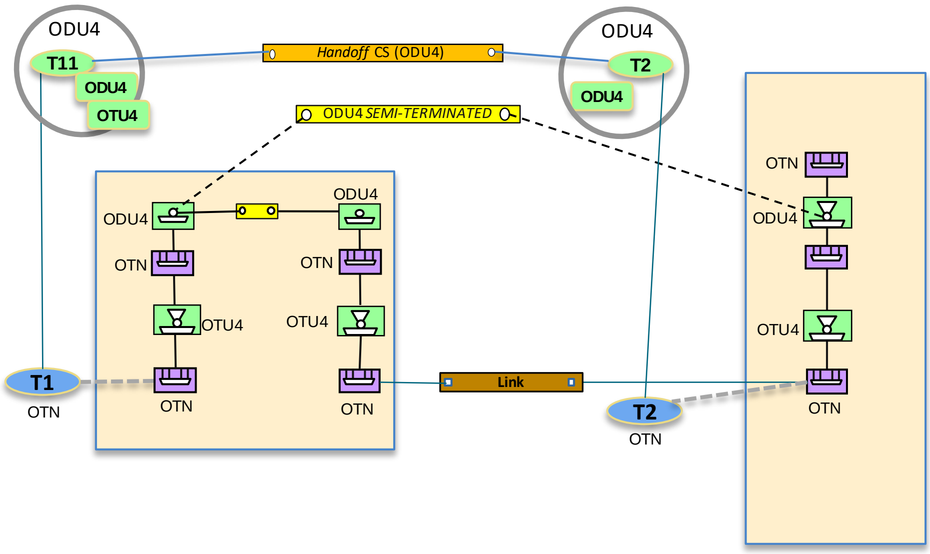


Figure 6-67 Server ODU *Handoff* Connectivity Service

Figure 6-68 and Figure 6-69 apply to Asymmetric Scenario 1: Handoff at ODU4 Layer, no ODU2 layer on ENNI, variation.

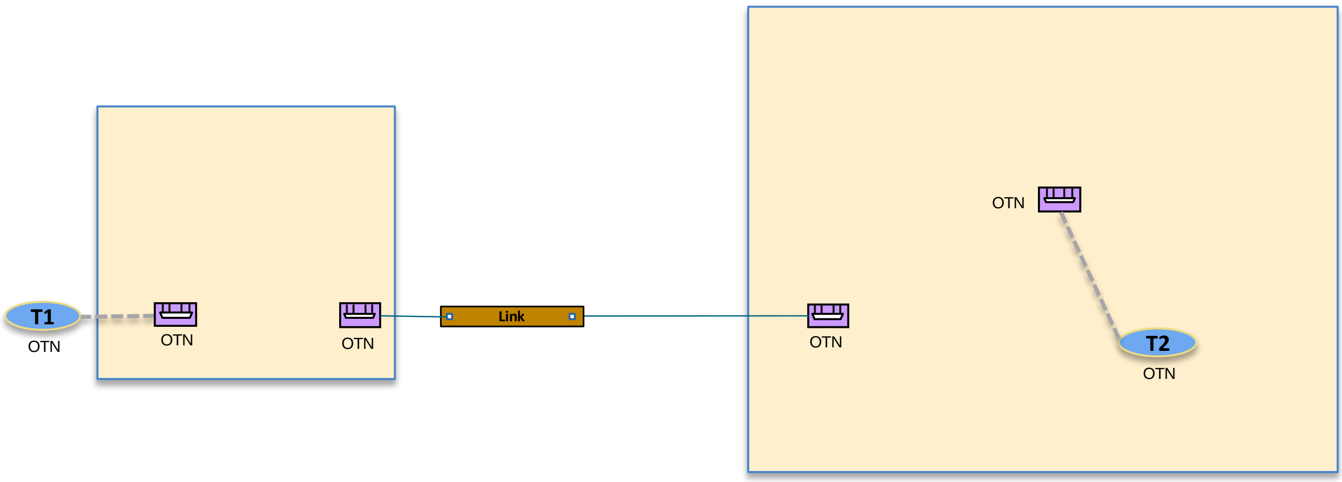


Figure 6-68 No “server” connections, variation

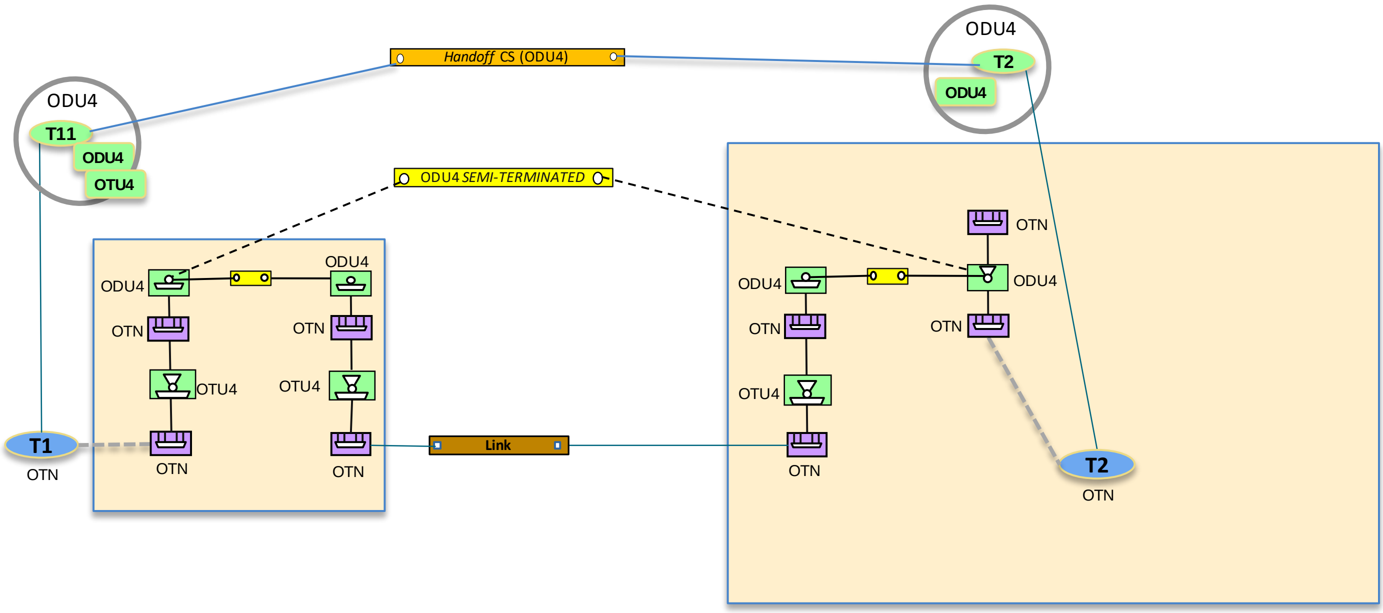


Figure 6-69 Server ODU *Handoff* Connectivity Service, variation

Figure 6-70 applies to Asymmetric Scenario 2: Handoff at ODU4 Layer, ODU2 layer switching on Edge Node.

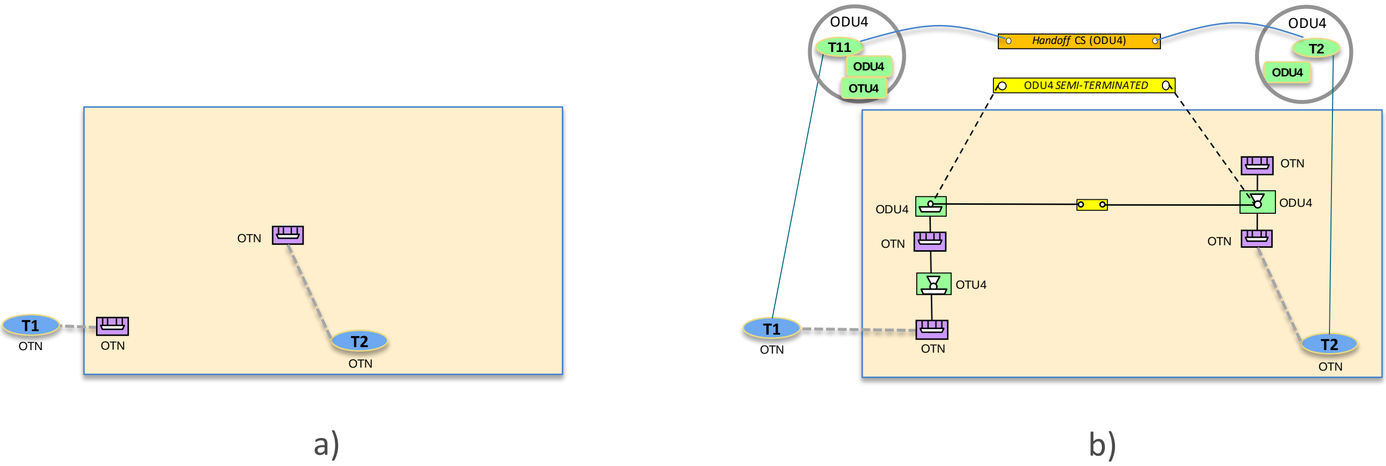


Figure 6-70 a) No “server” connections, b) Server ODU *Handoff* Connectivity Service

Figure 6-71 applies to Asymmetric Scenario 3: Handoff at ODU2 Layer.

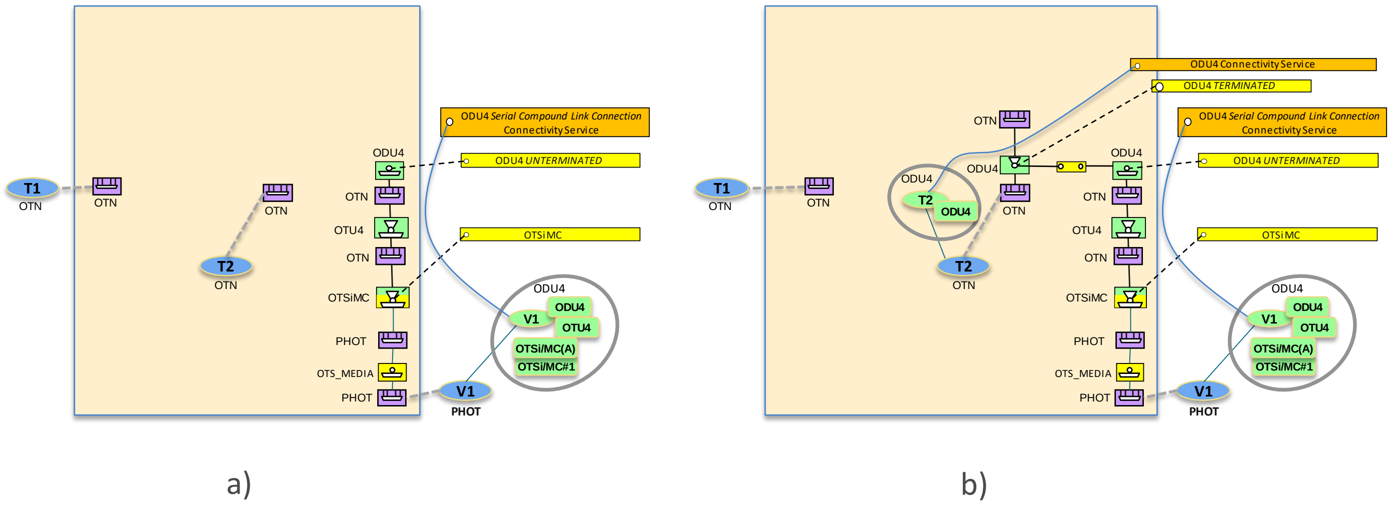


Figure 6-71 a) No ODU “server” connections, b) Server ODU Connectivity Service (not *Handoff*)

Figure 6-71 and Figure 6-72 apply to Asymmetric Scenario 4: Handoff at ODU4 Layer, ODU2 layer on ENNI.

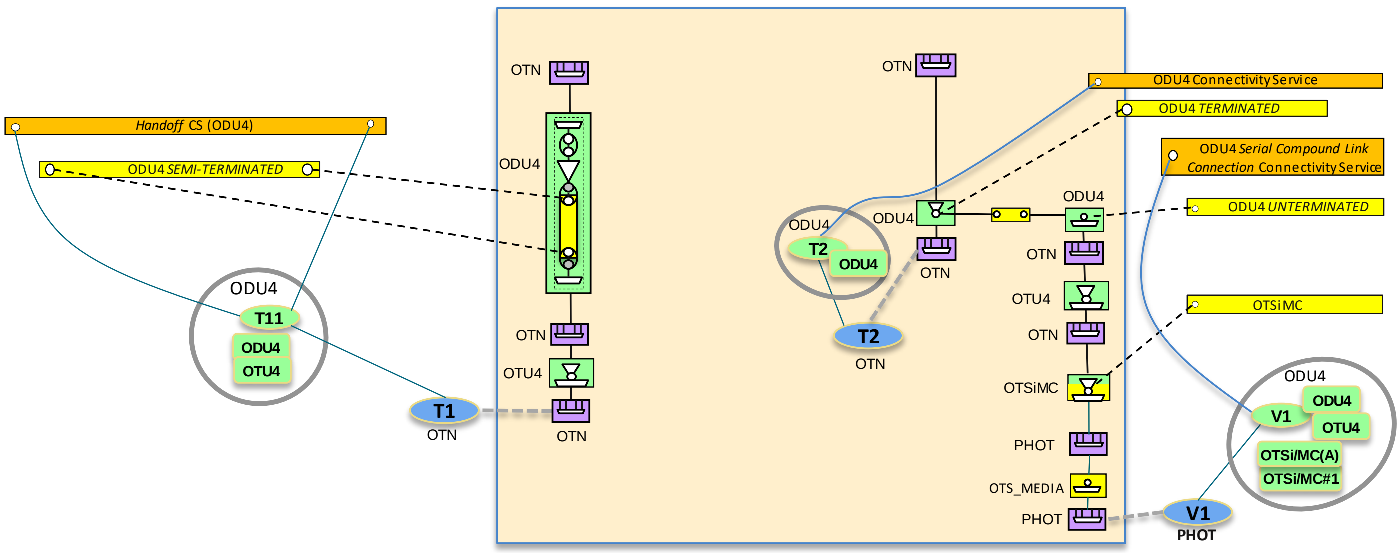


Figure 6-72 Server ODU *Handoff* Connectivity Service

6.2.12.2 Applicable Provisioning Scenarios

For this UC the provisioning scenarios defined in Section 6.2.2.3 apply, with applicable constraints on OTN layers. Note that in case the *transponder-to-transponder* connectivity is based on ODUCn container, then same scenarios apply, replacing the ODUk *Serial Compound Link Connection* CS with the ODUCn *Trail* CS.

6.2.12.3 Detailed Workflow

The initial scenario for this use case assumes the boundary interfaces between network domains to be E-NNI OTUk grey interfaces which shall be modeled as OTN NEPs with the "inter-domain-plug-id" identifier as described in UC0d.

6.2.12.4 Expected results

See Section 6.2.2.3 for examples on the expected results.

6.2.13 Use case 2a: DIGITAL_OTN with PHOTONIC_MEDIA/OTSiA Service Provisioning with channel selection

Number	**UC2a**
Name	**DIGITAL_OTN with PHOTONIC_MEDIA/OTSiA Service Provisioning with channel selection**
Technologies involved	Digital OTN, Optical
Process/Areas Involved	Planning and Operations
Brief description	This use case extends UC1d and UC1e by allowing the TAPI Client to define the spectrum, power management and further constraints, such the modulation-format or the application-identifier. The TAPI Server SHOULD provide the RESTCONF Response according to the criteria provided in Table 49. (RESTCONF responses are experimental).
Layers involved	PHOTONIC_MEDIA
Type	Provisioning
Description & Workflow	This UC is implemented following the same workflow described in “Description & Workflow” of UC1e

6.2.13.1 Examples of Time Zero Scenarios

For this UC the “time zero scenario” is the same as UC1d.

6.2.13.2 Applicable Provisioning Scenarios

For this UC the provisioning scenarios of UC1d apply, with the UC’s specific constraints on OTN and OTSiMC layers.

6.2.13.3 Relevant Parameters

This UC corresponds to the “transponder to transponder” network scenarios for provisioning. It allows channel selection. This UC assumes that the transceiver line port NEP (thus SIP) listed transceiver profiles provide enough information to deduce the supportable / applicable combinations in terms of OTSi. This case requires the generation of N number of OTSiMC Top Connections required to transport the service.

The client MAY specify the selected transceiver profile (which applies to the whole OTSiA)

The client MAY further constraint the service by adding the following layer protocol constraints:

OTU

otu-connectivity-service-end-point-spec	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/end-point/layer-protocol-constraint/tapi-digital-otn:otu-connectivity-service-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**

otu-csep-ttp-pac	Includes: **fec-type** (either standard-fec-type, with an identity based on STANDARD_FEC_TYPE or proprietary-fec-type) **baud-rate** (uint64)	RW	C	• Provided by *tapi-client*

See Table 35 for applicable OTSiA Layer Protocol Constraints. In this case, the attributes number-of-otsi and otsi-config/central-frequency is mandatory.

6.2.13.4 TAPI Server response behavior.

Please consider this list as preliminary. It will be updated based on received feedback.

Table 49: UC2a expected response behavior.

HTTP Response status code	Error-tag	Error-message	Condition description
201			Created
409	in-use	OTSi Spectrum resources not available across the network.	OTSi Spectrum resources not available across the network.
404	Invalid-value	OTSi Central frequency out of range	OTSi Central frequency out of supported range
404	Invalid-value	OTSi Central frequency adjustment granularity or grid type invalid	OTSi Central frequency adjustment granularity or grid type invalid
404	Invalid-value	Spectrum range invalid	Spectrum range not compatible with OTSi transmitter/receiver capabilities exposed in the related SIP.
404	Invalid-value	Invalid modulation	Modulation format code not supported by referenced OTSi/OTSiA SIP.
404	Invalid-value	Invalid application-identifier	Application Identifier not supported by referenced OTSi/OTSiA SIP.
404	Invalid-value	Transmit power out of range	Transmit power out of range supported by referenced OTSi/OTSiA SIP.

6.2.14 Use case 2b: DSR service provisioning with ODU channel selection

Number	**UC2b**
Name	**DSR service provisioning with ODU channel selection**
Technologies involved	Optical, DIGITAL_OTN layers
Process/Areas Involved	Planning and Operations
Brief description	This UC is intended to define the way the TAPI Client can request the creation of a DSR service with the selection of the ODU tributary slot (*channel selection*). This UC assumes that the DSR service is mapped into a Lower Order (LO) ODU container and multiplexed into a Higher Order (HO) ODU container. The channel selection involves such multiplexing. NOTE: Current RIA version only considers the selection of the position of the LO ODU in the HO ODU in the first encapsulation. It does not consider the effect of LO ODU switching; its applicability is limited to specific scenarios (such as when the LO ODU is used to frame the DSR service). Further versions will address the selection of resources in a more flexible way.
Layers involved	ODU
Type	Provisioning
Description & Workflow	See the detailed workflow UC1.0 with [**server-restrictions**].

6.2.14.1 Examples of Time Zero Scenarios

For this UC the “time zero scenarios” are the same as UC1a and UC1b.

6.2.14.2 Applicable Provisioning Scenarios

For this UC the following provisioning scenarios apply, with the UC’s specific constraints on OTN layers:

- Figure 6-10 DSR/ODUj CS on ODUk SCLC CS, ODUk Terminated Connection automatically created or reused
- Figure 6-11 DSR/ODU2 CS on ODU3 SCLC CS, ODU3 Terminated Connection automatically created or reused
- Figure 6-12 DSR/ODUj CS on ODUk SCLC CS, ODUk Term. Conn. autom. created or reused, no ODUj flexibility
- Figure 6-13 DSR/ODUj Connectivity Service on ODUk SCLC CS, auto creation of ODUk CS
- Figure 6-15 DSR/ODUj Connectivity Service on ODUk CS on ODUk SCLC CS
- Figure 6-40 DSR/ODUj CS on ODUk CS
- Figure 6-41 DSR CS on ODUj on ODUk CS (DSR flexibility)
- Figure 6-34 DSR/ODUFlex Connectivity Service on ODUCn CS
- Figure 6-35 DSR/ODUj CS on ODUk on ODUCn CS - ODUk Terminated Connection automatically created or reused
- Figure 6-36 DSR/ODUj CS on ODUk CS on ODUCn CS - Auto creation of ODUk CS
- Figure 6-38 DSR/ODUj CS on ODUk CS on ODUCn CS

6.2.14.3 Relevant Parameters

This extends UC1c with the selection of tributary slot list and port number.

odu-connectivity-service-end-point-spec	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/end-point/layer-protocol-constraint/tapi-digital-otn:odu-connectivity-service-end-point-spec**

Attribute	Allowed Values/Format	Mod	Sup	Notes
odu-csep-common-pac	Includes: odu-rate in kb/s, opu-tributary-slot-size, one of 1G25 or 2G5. See yang for details.	RW	M	• Provided by *tapi-client*
odu-csep-ctp-pac/ tributary-slot-list	Set of distinct (i.e., unique) integers (e.g., 2, 3, 5, 9, 15 representing the tributary slots TS#2, TS#3, TS#5, TS#9 and TS#15) which represents the resources occupied by the ODUk CTP.	RW	C	• Provided by *tapi-client* Used in UC2b when selecting the channel. Refer to the Yang description
odu-csep-ctp-pac/tributary-port-number	Tributary port number that is associated with the ODUk CTP, when the ODUk CTP is multiplexed into a server layer ODU TTP object. See clause 14.4.1/G.709-2016, 14.4.1.4/G.709-2016 or 20.4.1.1/G.709-2016 for ODU-Cn	RW	O	• Provided by *tapi-client* Used in UC2b when selecting the channel.
odu-csep-ttp-pac	Includes: configured-mapping-type configured-client-type	RW	C	• Provided by *tapi-client* The *configured mapping type* is mandatory if there are several mapping types available for the DSR service. The *configured client type* is optional if this layer protocol constraint is used while provisioning the client. The *configured client type* could be used when provisioning ODU services in a bottom-up approach (use case not covered in this RIA).
odu-cn-csep-ttp-pac	Includes number-of-odu-c	RW	C	• Provided by *tapi-server* • *Used in ODU-Cn configurations.*

6.2.15 Use case 2c: PHOTONIC_MEDIA/Media Channel(s) (MC/MCG) Service Provisioning with spectrum selection

Number	**UC2c**
Name	PHOTONIC_MEDIA/Media Channel(s) (MC/MCG) Service Provisioning with spectrum selection
Technologies involved	Optical
Process/Areas Involved	Planning and Operations
Brief description	This use case extends UC1f by allowing the TAPI Client to define the spectrum constraints of the MC service(s). The UC relies on the **tapi-photonic-media:mcg-connectivity-service-end-point-spec** within the MC Protocol Layer Constraint of the CSEPs. The power management constraints are modeled by the power-management-config-pac object.
Layers involved	PHOTONIC_MEDIA

Type	Provisioning
Description & Workflow	This UC is implemented following the same workflow described in "Description & Workflow" of UC1.0

6.2.15.1 Examples of Time Zero Scenarios

For this UC the "time zero scenarios" are the same as UC1f.

6.2.15.2 Applicable Provisioning Scenarios

For this UC the provisioning scenarios of UC1f apply, with the UC's specific constraints on MC layer.

6.2.15.3 Relevant Parameters

mcg-connectivity-service-end-point-spec	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/end-point/layer-protocol-constraint/tapi-photonic-media:mcg-connectivity-service-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
number-of-mc	Number of component MC. Must be >= 1	RW	M	• Provided by *tapi-client* • *This RIA only considers an MCG provisioning from a single SIP (e.g, single add /drop port).*
mc-spectrum-config-pac	List of *MC Spectrum Configurations*, indexed by local-id. Each element contains: local-id and name list. **spectrum** with upper-frequency and lower-frequency (in Hz) **edge-frequency-constraint** with adjustment granularity and grid-type **power-management-config-pac**	RW	C	• Provided by *tapi-client*
mc-grid-config-pac	List of *MC Grid Configurations*, indexed by local-id. Each element contains: local-id and name list. **n, m** int64 (as per ITU-T G.694.1 grid) **frequency-constraint** with adjustment granularity and grid-type **power-management-config-pac**	RW	C	• Provided by *tapi-client*

6.2.15.4 TAPI Server response behavior.

Please consider this list as preliminary. It will be updated based on received feedback.

Table 50: UC2c expected response behavior.

HTTP Response status code	**Error-tag**	**Error-message**	**Condition description**
200			Success

409	in-use	MC Spectrum resources not available across the network.	MC Spectrum resources not available across the network.
404	Invalid-value	Spectrum range invalid	Spectrum range not compatible with Photonic Media network filtering capabilities exposed in the MC/MCA related SIP.
404	operation-failed	Intending minimum output power constrain cannot be met.	Intending minimum output power constrain cannot be met.
404	operation-failed	Intending maximum output power constrain cannot be met.	Intending maximum output power constrain cannot be met.
409	operation-failed	Expected minimum input power constrain is not sufficient for MC service provisioning.	Physical impairment validation for the requested channel has failed due to insufficient OTSi input power.
409	operation-failed	Expected maximum input power constrain is incompatible for MC service provisioning.	Expected maximum input power constrain exceeds the supported input power of the Photonc_media layer add/drop ports.

6.2.16 Use case 3a: Include/exclude one or more nodes.

Number	**UC3a**
Name	**Include/exclude one or more nodes**
Technologies involved	Optical, Digital OTN, DSR layers
Process/Areas Involved	Planning and Operations
Brief description	This use case covers requesting a connectivity service with the inclusion/exclusion of the nodes selected by the TAPI client. The inclusion/exclusion constraint applies to all layers of connectivity supporting the service. For example, if node A is excluded from an DSR service then it shall not appear in any route of the supporting connections. NOTE.1: The UC uses the include-node and exclude-node lists. Implementations cannot make any assumption on the intended ordering. An implementation that conforms to a request with several include-node(s) may compute a route in which the nodes appear in any order. NOTE.2: The inclusion list may be partial, not covering all nodes in a route.
Layers involved	DSR/DIGITAL_OTN/PHOTONIC_MEDIA
Type	Provisioning
Description & Workflow	This UC is implemented following the same workflow described in "Description & Workflow" of UC1.0

6.2.16.1 Relevant Parameters

Table 51 complements the information included in the unconstrained service provisioning use cases. The connectivity service object includes a list of topology constraints (index by their local-id). This RIA assumes that all involved nodes are listed in a single topology constraint.

Table 51: Connectivity-service node topology-constrains object definitions.

connectivity-service	**/tapi-common:context/tapi-connectivity:connectivity-context/tapi-connectivity:connectivity-service/topology-constraint={local-id}**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
include-node	List of valid node refs (with topology-uuid and node-uuid)	RW	C	• Unordered and partial list • Implementations MUST support the inclusion of nodes. The attribute may not be present in all cases. • *Declarative* routing constraints not in the scope.
exclude-node	List of valid node refs (with topology-uuid and node-uuid)	RW	C	• Implementations MUST support the exclusion of nodes. The attribute may not be present in all cases.

6.2.17 Use case 3b: Include/exclude a link or group of links.

Number	**UC3b**
Name	**Include/exclude a link or group of links**
Technologies involved	Optical, ODU, DSR layers
Process/Areas Involved	Planning and Operations
Brief description	This use case covers requesting a connectivity service with the inclusion/exclusion of the links selected by the TAPI client. As in UC3a, the inclusion/exclusion constraint applies to all layers of connectivity supporting the service and the link lists are unordered and may be partial.
Layers involved	DSR/DIGITAL_OTN/PHOTONIC_MEDIA
Type	Provisioning
Description & Workflow	This UC is implemented following the same workflow described in “Description & Workflow” of UC1.0

6.2.17.1 Relevant Parameters

Table 52 complements the information included in the unconstrained service provisioning use cases. The connectivity service object includes a list of topology constraints (index by their local-id). This RIA assumes that all involved links are listed in a single topology constraint.

Table 52: Connectivity-service link topology-constrains object definitions.

connectivity-service	**/tapi-common:context/tapi-connectivity:connectivity-context/tapi-connectivity:connectivity-service/topology-constraint={local-id}**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
include-link	List of valid links refs (with topology-uuid and link-uuid)	RW	C	• Unordered and partial list • Implementations MUST support the inclusion of links. The attribute may not be present in all cases. • *Declarative* routing constraints not in the scope.
exclude-link	List of valid links refs (with topology-uuid and link-uuid)	RW	C	• Implementations MUST support the exclusion of links. The attribute may not be present in all cases.

6.2.18 Use case 3c: Include/exclude the route used by another service.

Number	**UC3c**
Name	Include/exclude the route used by another service.
Technologies involved	Optical, Digital OTN, DSR layers
Process/Areas Involved	Planning and Operations
Brief description	This use case covers requesting a connectivity service with the inclusion or exclusion of the resources used by another connectivity service(s). **Coroute-Inclusion:** Implementations SHOULD proceed in such a way that the connectivity resources used by the *included* service are reused, at the highest possible layer, for the service being set up **Diversity-Exclusion:** Implementations SHOULD proceed in such a way that the connectivity resources used by the excluded services, at the lowest layer of the topology, are excluded from the service being set up *Examples: In this context, the wording "includes X" means "refers to X in its coroute-inclusion" and "excludes X" means "refers to X in its diversity-exclusion list"* - A DSR service that includes another DSR service means that implementations SHOULD encapsulate the new DSR in the same ODUs of the included service - An MC service that includes an MC service means that implementations SHOULD reuse the OMS/OTS sections. - An MC service that excludes an MC service means that implementations SHOULD exclude the OMS/OTS sections. - A DSR service that includes an ODU service means that implementations SHOULD encapsulate the new DSR in the ODU service In case the referenced CS by the coroute-inclusion or diversity-exclusion parameters changes its route (e.g., due to a restoration), the service may not change accordingly, i.e., the TAPI server is not required to maintain the relationship between resources as stated above.
Layers involved	DSR/ODU/PHOTONIC_MEDIA
Type	Provisioning
Description & Workflow	This UC is implemented following the same workflow described in "Description & Workflow" of UC1.0

6.2.18.1 Relevant Parameters

Table 53 complements the information included in the unconstrained service provisioning use cases.

Table 53: Connectivity-service coroute-inclusion and diversity-exclusion object definitions.

connectivity-service	**/tapi-common:context/tapi-connectivity:connectivity-context/tapi-connectivity:connectivity-service/connectivity-constraint**

Attribute	Allowed Values/Format	Mod	Sup	Notes
coroute-inclusion	connectivity-service-uuid: **connectivity-service-ref** - */tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/uuid*	RW	C	• Provided by *tapi-client* • *Implementations MUST support coroute-inclusion if a CS is referred to.*
diversity-exclusion	List of {connectivity-service-uuid}: **connectivity-service-ref** - */tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/uuid* }	RW	C	• Provided by *tapi-client* • *Implementations MUST support diversity-exclusion if one (or more) CS is (are) referred to.*

6.2.19 Use case 3d: Diverse Routing in SRG failure.

Number	UC3d
Name	Diverse Routing in SRG failure
Technologies involved	Optical, Digital OTN, DSR layers
Process/Areas Involved	Planning and Operations
Brief description	**Disclaimer: This use case is in a draft state, the final definition will be completed based on the feedback provided by the industry upon this release of the reference specification.** This use case deals with the provisioning of a connectivity service with a given level of protection and risk disjointness. As such, the expected result will be one top level connection for the service with two (disjoint) routes. This use case assumes 1) Shared Risk Groups (SRGs) are predefined (in links, nodes, etc.) and considers the provisioning of SRG policies and provide route disjointness upon these policies. 2) The TAPI client jointly specifies an **SRG disjoint-policy and a resilience-type.** The SDN-C MUST ensure that both routes (Nominal and Backup) do not share any SRG present in the network.
Layers involved	DSR/ODU/PHOTONIC_MEDIA
Type	Provisioning
Description & Workflow	This UC is implemented following the same workflow described in “Description & Workflow” of UC1.0

6.2.19.1 Relevant Parameters

Table 54: Connectivity-service diversity-policy for SRGs. complements the information included in the unconstrained service provisioning use cases

Table 54: Connectivity-service diversity-policy for SRGs.

Connectivity-service	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
resilience-constraint/resilience-type	With protection-type one value which shall not be “NO_PROTECTION”	RW	C	• Provided by *tapi-client* Depends on the supported protection types (see also UC.5X)
routing-constraint /diversity-policy	One of [“SRLG”,“SRNG”,]	RW	M	• Provided by *tapi-client* [mandatory for this use case: SRLG or SRNG values] See risk-characteristic attribute in Node and Link

6.2.20 Use case 3e: Provisioning based on min hops policy

Number	**UC3e**
Name	Provisioning based on min hops policy
Technologies involved	Optical, Digital OTN, DSR layers
Process/Areas Involved	Planning and Operations
Brief description	This use case covers requesting a connectivity service with the selection of the MIN_WORK_ROUTE_HOP route-objective-function, which requires the TAPI Server to minimize the number of links of the **lowest server layer and qualifier** in the context. In case of applying this use case for protection services, the TAPI client MAY alternatively use MIN_SUM_OF_WORK_AND_PROTECTION_ROUTE_HOP. In this case, the expected behavior is the TAPI server will the best combination of WORK and PROTECTION routes which minimizes the number of hops as previously defined.
Layers involved	DSR/DIGITAL_OTN/PHOTONIC_MEDIA
Type	Provisioning
Description & Workflow	This UC is implemented following the same workflow described in “Description & Workflow” of UC1.0

6.2.20.1 Relevant Parameters

Table 55: Connectivity-service route-objective-function (UC3e). complements the information included in the unconstrained service provisioning use cases.

Table 55: Connectivity-service route-objective-function (UC3e).

Connectivity-service	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/routing-constraint**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
route-objective-function	One of ["MIN_WORK_ROUTE_HOP", "MIN_SUM_OF_WORK_AND_PROTECTION_ROUTE_HOP"]	RW	M	• Provided by tapi-client

6.2.21 Use case 3f: Provisioning based on min latency policy

Number	**UC3f**
Name	Provisioning based on min latency policy
Technologies involved	Optical, Digital OTN, DSR layers
Process/Areas Involved	Planning and Operations
Brief description	This use case covers requesting a connectivity service with the selection of the MIN_WORK_ROUTE_LATENCY route-objective-function, which shall enforce the TAPI Server to minimize the end-to-end latency of the service. In case of applying this use case for protection services, the TAPI client MAY alternatively use MIN_SUM_OF_WORK_AND_PROTECTION_ROUTE_LATENCY. In this case, the expected behavior is the TAPI server will the best combination of WORK and PROTECTION routes which minimizes the latency as previously defined.
Layers involved	DSR/DIGITAL_OTN/PHOTONIC_MEDIA
Type	Provisioning
Description & Workflow	This UC is implemented following the same workflow described in "Description & Workflow" of UC1.0

6.2.21.1 Relevant Parameters

The table below complements the information included in the unconstrained service provisioning use cases.

Table 56: Connectivity-service route-objective-function (UC3f)

connectivity-service	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/routing-constraint**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
route-objective-function	One of [“MIN_WORK_ROUTE_LATENCY", “MIN_SUM_OF_WORK_AND_PROTECTION_ROUTE_LATENCY”]	RW	M	• Provided by *tapi-client*

6.3 Inventory

NOTE: In some examples, the equipment category has been abbreviated for convenience as:

- RACK,
- SUBRACK,
- CIRCUIT_PACK,
- SMALL_FORMFACTOR_PLUGGABLE,
- STAND_ALONE_UNIT.

the formal values are:

- EQUIPMENT_CATEGORY_RACK,
- EQUIPMENT_CATEGORY_SUBRACK,
- EQUIPMENT_CATEGORY_CIRCUIT_PACK,
- EQUIPMENT_CATEGORY_SMALL_FORMFACTOR_PLUGGABLE,
- EQUIPMENT_CATEGORY_ STAND_ALONE_UNIT.

6.3.1 Use case 4a: Introduction of references to external inventory model.

Number	**UC4a**
Name	Introduction of references to external inventory model.
Technologies involved	Physical
Process/Areas Involved	Planning and Operations
Brief description	The INVENTORY_ID tag must be included in the following TAPI objects: • *tapi-topology:node-edge-point* • *tapi-common:service-interface-point* Note: The INVENTORY_ID value format is defined in Section 4.2, which defines how to express the relative position of each component.
Layers involved	Not applicable
Type	Inventory
Description & Workflow	See UC0a, UC0b on the Context, SIP and topology discovery.

6.3.2 Use case 4b: Complete Inventory model for NBI Interface.

Number	**UC4b**
Name	Complete Inventory model for NBI Interface.
Technologies involved	Physical
Process/Areas Involved	Planning and Operations
Brief description	This use case involves the retrieval of inventory information managed by the SDN controller that implements the **/tapi-common:context/tapi-equipment:physical-context**
Layers involved	Not applicable
Type	Inventory
Description & Workflow	The workflow consists of the retrieval of the inventory information. The TAPI server MUST support: • Full inventory of all “devices” with all their parameters • Full inventory of equipment (chassis, slot, ports/pluggables) and the hierarchy representation of the equipment within a device or a group of devices (by iteration) with their parameters. • Full inventory of the equipment used within a connectivity service or a precalculated-path • Full inventory of "physical spans" with their parameters.

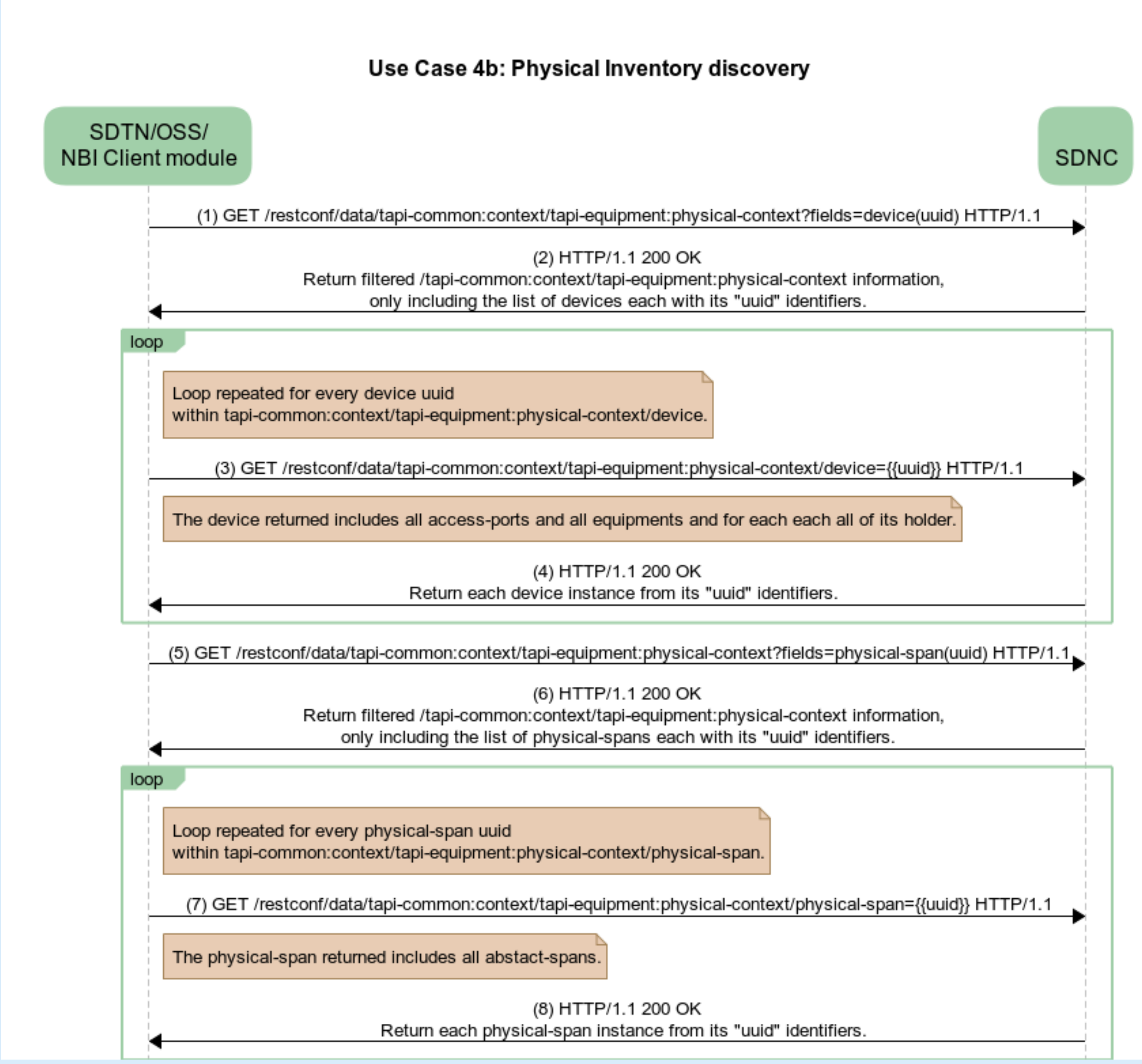


Figure 6-73 UC-4b: Discovery of Physical Inventory (devices, equipment, and physical span)

6.3.2.1 Relevant Parameters

The following parameters must be included for each item, and they must be present in the following path**: /tapi-common:context/tapi-equipment:physical-context.** Note that some commonly used concepts are mapped into TAPI equivalents such as "Equipment type" is category, the relative position of the component into the network element is mapped to contained-holder/actual-holder/common-holder-properties/holder-location

Table 57: Device and Equipment object's parameters required for UC4b.

Device	**/tapi-common:context/tapi-equipment:physical-context/device**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
equipment	List of pieces of equipment (see next table)	RO	M	• Provided by *tapi-server*

name	List of {value-name: value} "value-name": "NW-NE-NAME" "value": " [0-9a-zA-Z_]{64}"	RO	M	• Provided by ***tapi-server*** • NW-NE-NAME is described in Section 4.2
uuid	Device uuid as per RFC 4122	RO	M	• Provided by *tapi-server*
access-port	List of Access Ports with {uuid, connector-pin, name} • **uuid**: Access Port uuid • **connector-pin**: List of {connector-identification, pin-identification, equipment-uuid} • **name** MUST include { "value_name": "PORT_NUMBER", "value": " [0-9a-zA-Z_]{64}" }	RO	M	• Access port is the bridge between the logical model (NEPs etc.) and the Physical Model (a NEP is augmented with an access-port uuid and device uuid) • **connector-pin**: The list of Pins that support the Access Port. Each connector pin identifies the corresponding equipment-uuid • *Starting from a NEP, it is possible to obtain the list of equipment supporting it via its supporting-access-port augmentation and the equipment-uuids referred in each of its connector-pins.*

The following table applies to the equipment. Note that since TAPI 2.4 does not include admin and operational state yang leaves for physical context objects such as equipment, this RIA recommends that such states be reflected into all the supported logical elements (NEP) (Operational state /tapi-common:context/tapi-topology:topology-context/topology/node/owned-node-edge-point/operational-state and /tapi-common:context/tapi-topology:topology-context/topology/node/owned-node-edge-point administrative-state)

equipment	**/tapi-common:context/tapi-equipment:physical-context/device/equipment**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
contained-holder	List of { occupying-fru, expected-holder, actual-holder, uuid , name} • occupying-fru {device-uuid, equipment-uuid} • expected-holder/**common-holder-properties** • actual-holder/**common-holder-properties** • uuid • name {value-name, value} ○ "value-name":"HOLDER_NAME" ○ "value":"[0-9a-zA-Z_]{64}"	RO	M	• Provided by *tapi-server* Represent all the children contained in the equipment
category	One of { EQUIPMENT_CATEGORY_RACK, EQUIPMENT_CATEGORY_SUBRACK, EQUIPMENT_CATEGORY_CIRCUIT_PACK, EQUIPMENT_CATEGORY_SMALL_FORMFACTOR_PLUGGABLE, EQUIPMENT_CATEGORY_ STAND_ALONE_UNIT }	RO	M	• Provided by *tapi-server*
equipment-location	String	RO	O	• Provided by tapi-server
geographical-location	String	RO	O	• Provided by *tapi-server*
is-expected-actual-mismatch	Boolean	RO	M	• Provided by *tapi-server*
expected-equipment	List of { expected-non-field-replaceable-module, holder, common-equipment-properties	RO	M	• Provided by *tapi-server* • *The field* expected-non-field-replaceable-module are expected

	}			to encode non removeable pieces of equipment.
actual-equipment	Container with { actual-non-field-replaceable-module, common-actual-properties, common-equipment-properties }	RO	M	• Provided by *tapi-server* • *In common-equipment-properties, field asset-type-identifier SHALL correspond to the concept of "Part Number" and /or "Operator ID type"*
name	List of {value-name: value} "value-name": "EQUIPMENT_NAME" "value": " [0-9a-zA-Z_]{64}"	RO	M	• Provided by *tapi-server*
uuid	Equipment uuid, as per RFC 4122	RO	M	• Provided by *tapi-server*

Table 58: Common-holder-properties object's parameters required for UC4b.

common-holder-properties	**/tapi-common:context/tapi-equipment:physical-context/device/equipment/contained-holder/actual-holder/common-holder-properties**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
holder-category	"HOLDER_CATEGORY_SLOT"	RO	M	• Provided by *tapi-server* A guided holder with fixed connectors. The guided holder is designed to take a particular form of CIRCUIT_PACK or SMALL_FORMFACTOR_PLUGGABLE
is-guided	Boolean	RO	M	• Provided by *tapi-server* This attribute indicates whether the holder has guides that constrain the position of the equipment in the holder or not.
holder-location	String	RO	M	• Provided by *tapi-server* The relative position of the holder in the context of its containing equipment along with the position of that containing Equipment (and further recursion).

Table 59: Common-equipment-properties object's parameters required for UC4b.

common-equipment-properties	**/tapi-common:context/tapi-equipment:physical-context/device/equipment/actual-equipment/common-equipment-properties**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
asset-type-identifier	String	RO	O	• Provided by *tapi-server* Represents the invariant properties of the equipment asset allocated by the operator that define and characterize the type **Operator_ID_type**
equipment-type-description	String	RO	M	• Provided by *tapi-server* Text describing the type of Equipment.
equipment-type-identifier	String	RO	M	• Provided by *tapi-server* This attribute identifies the part type of the equipment
equipment-type-name	String	RO	M	• Provided by *tapi-server* This attribute identifies the type of the equipment.
equipment-type-version	String	RO	M	• Provided by *tapi-server* This attribute identifies the version of the equipment.
manufacturer-identifier	String	RO	O	• Provided by *tapi-server* The formal unique identifier of the manufacturer.
manufacturer-name	String	RO	M	• Provided by *tapi-server* The formal name of the manufacturer of the Equipment.

Table 60: Common-actual-properties object's parameters required for UC4b.

common-actual-properties	**/tapi-common:context/tapi-equipment:physical-context/device/equipment/actual-equipment/common-actual-properties**

Attribute	Allowed Values/Format	Mod	Sup	Notes
asset-instance-identifier	String	RO	M	• Provided by *tapi-server* This attribute represents the asset identifier of this instance from the operator's perspective.
is-powered	Boolean	RO	O	• Provided by *tapi-server* The state of the power being supplied to the equipment. Note that this attribute summarizes the power state.
manufacture-date	Date-and-time	RO	C	• Provided by *tapi-server* This attribute represents the date on which this instance is manufactured.
serial-number	String	RO	M	• Provided by *tapi-server* This attribute represents the serial number of this instance
temperature	Decimal64	RO	O	• Provided by *tapi-server* The temperature is mandatory for FAN Cards (CIRCUIT-PACK) and SMALL_FORM_FACTOR equipment, it can be provided for any other equipment when available in the supplier equipment. The measured temperature of the Equipment.

Note: A device includes a list of access ports, which in turn has a list of connector pins, keyed by *connector-identification*, *pin-identification* and *equipment-uuid*. In case the connector-identification and/or pin-identification is not present for a given access-port the used key to access a given connector-pin MUST be the concatenation of empty strings for the missing values and equipment-uuid (according to RESTCONF RFC8040 Sec 3.5.3). Each key leaf value except the last one is followed by a comma character. E.g., for a given access-port's connector-pin entry, the resource URI should be:

…/tapi-equipment:access-port={uuid}/connector-pin=",,{equipment-uuid}"

In other words, when accessing a list entry, keys are separated by commas and missing keys for list entries correspond to empty strings.

The following table provides the list of value names that MUST be added to a given device (**/tapi-common:context/tapi-equipment:physical-context/tapi-equipment:device/tapi-equipment:name**) with their respective "value-name".

Table 61: Additional device object's parameters required for UC4b (via name value pairs).

device	/tapi-common:context/tapi-equipment:physical-context/device			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
NE_NAME	"value-name": "NE_NAME" "value": " [0-9a-zA-Z_]{64}"	RO	M	• Provided by *tapi-server*
NE_ID	"value-name": "NE_ID" "value": "{NE_ID}"	RO	M	• Provided by *tapi-server*
GATEWAY	"value-name": "GATEWAY" "value": "{Name_Gateway_Device}"	RO	O	• Provided by *tapi-server* • It should be filled with the NE_NAME of the Gateway device, it is only mandatory if there is another NE acting as IP GATEWAY for this NE in the DCN
NE_TYPE	"value-name": "NE_TYPE" "value": {Name_NE_type}"	RO	M	• Provided by *tapi-server*
IP	"value_name": "IP" "value": "{IP_Device}"	RO	M	• Provided by *tapi-server*
MASK	"value_name": "MASK", "value": "{Mask_Device}"	RO	C	• Provided by *tapi-server*
CREATION_TIME	"value_name": "CREATION_TIME" "value": "{ Creation_time _Device}"	RO	C	• Provided by *tapi-server* • IETF date-and-time format: *'\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}(\.\d+)?' + '(Z|[\+\-]\d{2}:\d{2})*

Table 62: Additional physical-span parameters required for UC4b

device	**/tapi-common:context/tapi-equipment:physical-context/physical-span**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
uuid	As per RFC 4122	RO	M	Provided by *tapi-server*
name	List of names for the span	RO	C	Provided by *tapi-server*
access-port	Including: device-uuid, access-port-uuid	RO	M	Provided by *tapi-server* One or more access ports.
abstract-strand	Including, optionally: List of adjacent strands List of spliced strands List of connector-pin List of to-strand-joint List of strand joints List of strand-media-characteristics	RO	O	Provided by *tapi-server* Depends on the composition of the physical-strand. See Section 3.2.5 for a description. Strand media characteristics MAY encode properties of e.g., fiber, etc. and the current format is unspecified.

6.3.2.2 Relative location of component with TAPI using holder location

The following picture shows the relative position of each “equipment” (chassis, slot, sublot, port) in a graphical representation. The relation between TAPI naming and the picture is the following:

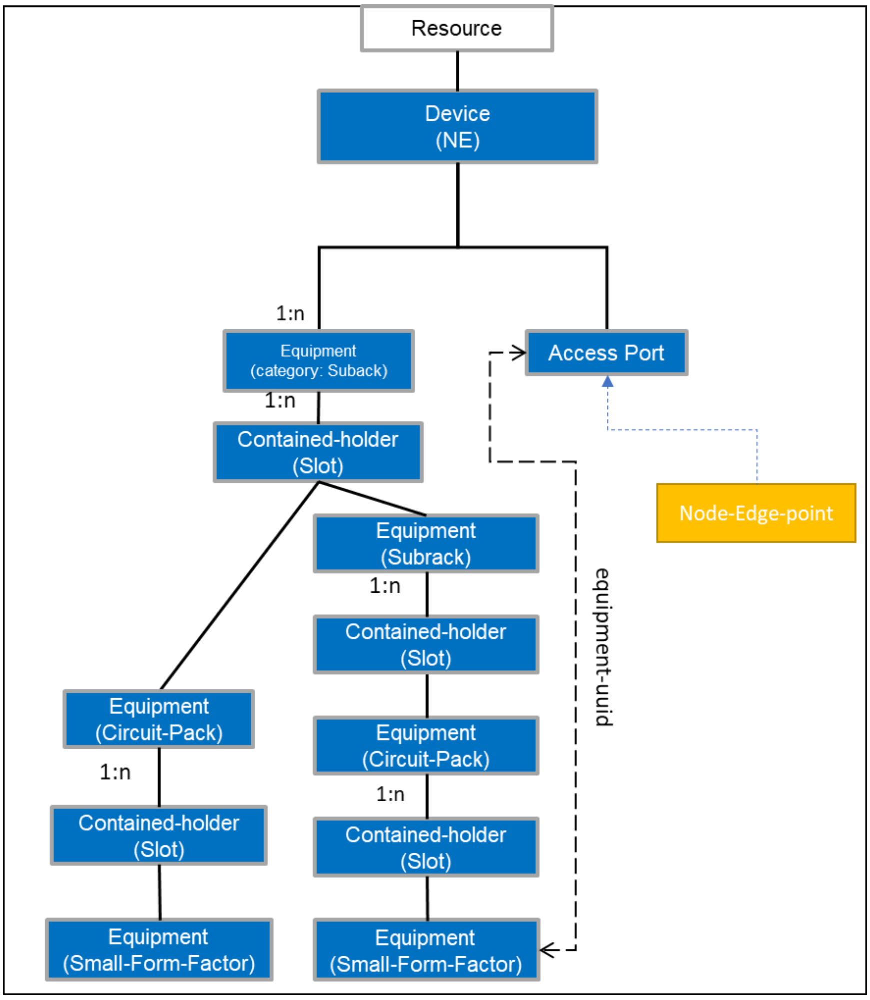


Figure 6-74 UC-4b Hierarchical arrangement of equipment objects with TAPI 2.1.3.

- Chassis=SUBRACK
- Card in slot= CIRCUIT_PACK/ SUBRACK
- Port in circuit pack= SMALL_FORMFACTOR_PLUGGABLE

The TAPI Server MUST use the ***tapi-equipment:contained-holder/actual-holder/common-holder-properties/holder-location*** to represent the **relative position of the contained-holders within the SUBRACK** equipment. The format of the holder-location string MUST be: *"SlotPosition"-"SubSlotPosition"* For convention, **if there is not sub-slot within a slot, the sub-slot value must be 0.**

There are some considerations needed to be taken to define a rule convention for filling this attribute. Three different scenarios are considered:

a. **Division**: The equipment slot structure is fixed, there is only one level of Holder objects, which may represent both "full slot" space or "half-sized slot" space cases. In other words, the Holder always represents the smallest granularity occupancy model. In this case, the ***holder-location*** MUST be: *"SlotPosition"-"0"*

b. **Hierarchy**: If the equipment slot structure can change dynamically (i.e., by software configuration of the SUBRACK equipment), an additional dimension of holder-location (i.e., a "sub-slot") must be introduced. In order to represent this sub-slot dimension, the list of ***tapi-equipment:contained-holder*** objects shall be dynamically increased with the new elements representing the partitioning. In this case, the ***holder-location*** MUST be: *"SlotPosition"-"SubSlotPosition"*.

c. **Specific Hardware (HW)**: In this case, a specific hardware is necessary to implement "sub-slotting". In this case, the existing Holder object will host an Equipment object (which MUST be a SUBRACK category equipment object) which at the time it is plugged-in, it enables the sub-slotting capability of the parent hardware. Then, the parent SUBRACK equipment holder-location arrangement shall follow one of the previous two models (depending on whether specific HW enabling sub-slotting is plugged or not). Please note, this extra-HW equipment is considered not implementing any control logic but just 'enables' the subslots space within the parent SUBRACK.

Then, according to the previous definition, the ***container-location*** **string represents the relative location of the container holder within an equipment**.

The following examples shows all the different possibilities and how to model them.

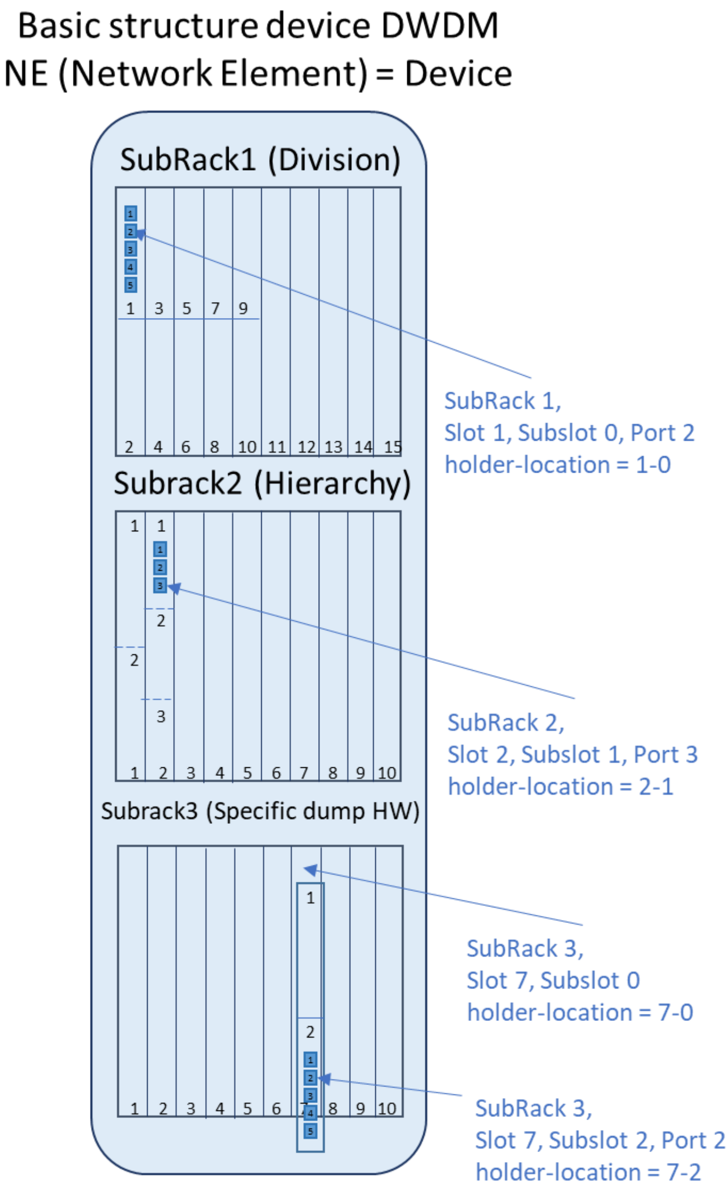


Figure 6-75 UC-4b Network Element Subracks container-holder location examples.

To complete the picture, the examples illustrated in Figure 6-75 are developed in TAPI model, including the holder-location value and the mapping to the INVENTORY_ID format presented in UC4a. Please note that the INVENTORY_ID will represent the absolute location of each equipment component, so it is derived from the position of the equipment within the tree.

Example Subrack1

Linecard holder-location in Subrack1

tapi-equipment:equipment[category=SUBRACK]/contained-holder/actual-holder/

"holder-location": "1-0"

tapi-equipment:equipment[category=SUBRACK]/contained-holder/

"name": "/ne=MadridNorte/r=1/sh=1/sl=1/s_sl=0"}]

Port2 holder-location in Linecard

tapi-equipment:equipment[category=CIRCUIT_PACK]/contained-holder/actual-holder/

"holder-location": "2-0"

tapi-equipment:equipment[category=CIRCUIT_PACK]/contained-holder/

"name": "/ne=MadridNorte/r=1/sh=1/sl=1/s_sl=0/p=2"}]

Example Subrack2

Linecard holder-location in Subrack2

tapi-equipment:equipment[category=SUBRACK]/contained-holder/actual-holder/

"holder-location": "2-1"

tapi-equipment:equipment[category=SUBRACK]/contained-holder/

"name": "/ne=MadridNorte/r=1/sh=2/sl=2/s_sl=1"}]

Port holder-location in Linecard

tapi-equipment:equipment[category=CIRCUIT_PACK]/contained-holder/actual-holder/

"holder-location": "3-0"

tapi-equipment:equipment[category=CIRCUIT_PACK]/contained-holder/

"name": [{"value_name": "INVENTORY_ID",

"value": "/ne=MadridNorte/r=1/sh=2/sl=2/s_sl=1/p=3"}]

Example Subrack3

Extra HW SUBRACK holder-location in Subrack3

tapi-equipment:equipment[category=SUBRACK]/contained-holder/actual-holder/

"holder-location": "7-0"

tapi-equipment:equipment[category=SUBRACK]/contained-holder/

"name": [{"value_name": "INVENTORY_ID",

"value": "/ne=MadridNorte/r=1/sh=3/sl=7/s_sl=0"}]

Linecard holder-location in Subrack3

tapi-equipment:equipment[category=SUBRACK]/contained-holder/actual-holder/

"holder-location": "7-2"

tapi-equipment:equipment[category=SUBRACK]/contained-holder/

"name": [{"value_name": "INVENTORY_ID",

"value": /ne=MadridNorte/r=1/sh=3/sl=7/s_sl=2"}]

Port holder-location in Linecard

tapi-equipment:equipment[category=CIRCUIT_PACK]/contained-holder/actual-holder/

"holder-location":"2-0"

tapi-equipment:equipment[category=CIRCUIT_PACK]/contained-holder/

"name":[{"value_name": "INVENTORY_ID",

"value": "/ne=MadridNorte/r=1/sh=3/sl=7/s_sl=2/p=2"}]

Some examples of INVENTORY_ID for the node-edge-points potentially mapped to the ports described in the previous examples:

Example 1:

"name": [{"value_name": "INVENTORY_ID", "value":

"/ne=MadridNorte/r=1/sh=1/sl=1/s_sl=0"}]

Example 2:

"name": [{"value_name": "INVENTORY_ID", "value": "/ne=MadridNorte/r=1/sh=2/sl=2/s_sl=1/p=3"}]

Example 3:

"name": [{"value_name": "INVENTORY_ID", "value": "/ne=MadridNorte/r=1/sh=3/sl=7/s_sl=2/p=2"}]

6.4 Resiliency

This section deals with use cases covering resiliency (i.e., protection and restoration).

6.4.1 Reversion Modes

In the cases involving *protection* (either 1:1 or 1+1) the TAPI client MUST specify the expected behavior regarding the reversion to the preferred connection's route. This applies to use cases 5a, 5b, 7a, 7b and 8. In particular, the reversion mode may be the following (with resource commonly referring to a route):

- NON_REVERTIVE, where a Connection switched to a lower priority (non-preferred/spare/protection) resource will not revert to a higher priority (preferred/intended/nominal) resource when that recovers.

- REVERTIVE, where a Connection switched to a lower priority (non-preferred/spare/protection) resource will revert to a higher priority (preferred/intended/nominal) resource when that recovers (potentially after some wait-to-revert-time).

In this sense,

- *wait-to-revert-time*: if the reversion mode is REVERTIVE, this attribute specifies the time to wait after a fault clears on a higher priority (preferred) resource before reverting to the preferred resource.

This is shown in the following tree snippet:

```
module: tapi-connectivity
  augment /tapi-common:context:
    +--rw connectivity-context
       +--rw connectivity-service* [uuid]
          +--rw resilience-constraint
          |  +--rw reversion-mode?                       reversion-mode
          |  +--rw wait-to-revert-time
                +-- rw value?  uint64
                +-- rw unit?   time-unit
```

For the resilience use cases, the following parameters apply.

Table 63: Connectivity-service parameters for reversion

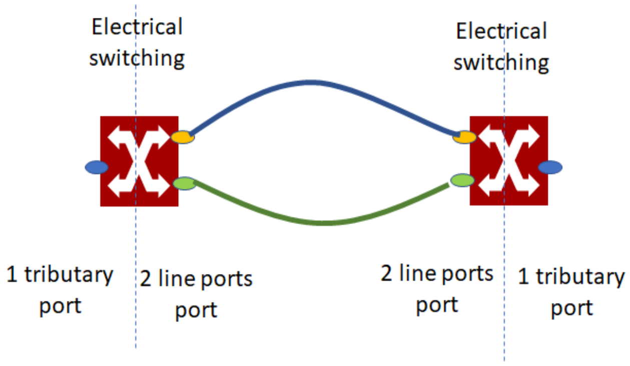
	/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/resilience-constraint			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
reversion-mode	One of ["REVERTIVE", "NON_REVERTIVE"]	RW	M	• Provided by *tapi-client*
wait-to-revert-time	With value and unit	RW	C	• Provided by *tapi-client* in provisioning • When provided by server, • This attribute is mandatory in connection objects when the reversion-mode is REVERTIVE. • The supported values MAY be additionally constrained by the underlying hardware. A config operation with unsupported values MUST fail.

NOTE: Updates for OLP Use cases for TAPI 2.4 are left for a subsequent release (Feb 2023)

6.4.2 Use case 5a: OLP OMS/OTS_MEDIA Protection Discovery

6.4.3 Use case 5b: OLP-based Transponder to Transponder Protection with Diverse Service Provisioning

6.4.4 Use case 5c: 1+1 DSR/ODU protection with Diverse Service Provisioning (eSNCP)

Number	**UC5c**
Name	1+1 protection DSR/ODU with Diverse Service Provisioning (eSNCP)
Technologies involved	DSR, DIGITAL_OTN
Process/Areas Involved	Planning and Operations
Brief description	This use case covers the use of the electrical SubNetwork Connection Protection (eSNCP, also referred to as ODU SNCP) for protected services at the DIGITAL_OTN layer. Cross-connections are used to implement dual feeding and selective receiving and protection switching is triggered by network conditions and should generate the corresponding OTN alarms (see figure).  The protection process MUST be triggered automatically and the TAPI server MUST notify the TAPI client about the service condition changes. For example, as defined in UCs 15X.
Layers involved	DIGITAL_OTN
Type	Resilience
Description & Workflow	The connectivity-service is requested between two DSR CSEPs and requires the reservation of two disjoint routes at the ODU layer between transponder’s line interfaces. The connectivity-service request includes SIPs representing the client layer interfaces. The Connectivity Service object sent to the TAPI Server MUST include the ***tapi-connectivity:connectivity-service/resilience-constraint/resilience-type/protection-type*** attribute with **ONE_PLUS_ONE_PROTECTION** attribute value. The TAPI Client MAY delegate the protection role selection to the TAPI Server during the CS provisioning process. Note that TAPI v.2.1.3 allowed a client to specify protection roles by using additional CSEPs, referring to the relevant available (internal) SIPs. In TAPI 2.4 it is preferred to specify routing constraints based on protection roles (e.g., WORK, PROTECT). The current approach is to use the resiliency route constraint list (tapi-connectivity:connectivity-

	service/resilience-constraint/resiliency-route-constraint) and add a topology-constraint as appropriate. In this case, the priority value 1 (tapi-connectivity:connectivity-service/resilience-constraint/resiliency-route-constraint['local-id']/ priority) MUST be associated to WORK protection role, and priority value 2 MUST be associated to PROTECT protection role. This UC is implemented following the same workflow described in "Description & Workflow" of UC1.0 *Note: this UC can be easily extended to ONE_FOR_ONE_PROTECTION and the same considerations apply.*

6.4.4.1 Expected result [example]

The expected result after the creation of the eSNCP DSR/ODU Connectivity Service is represented in Figure 6-76.

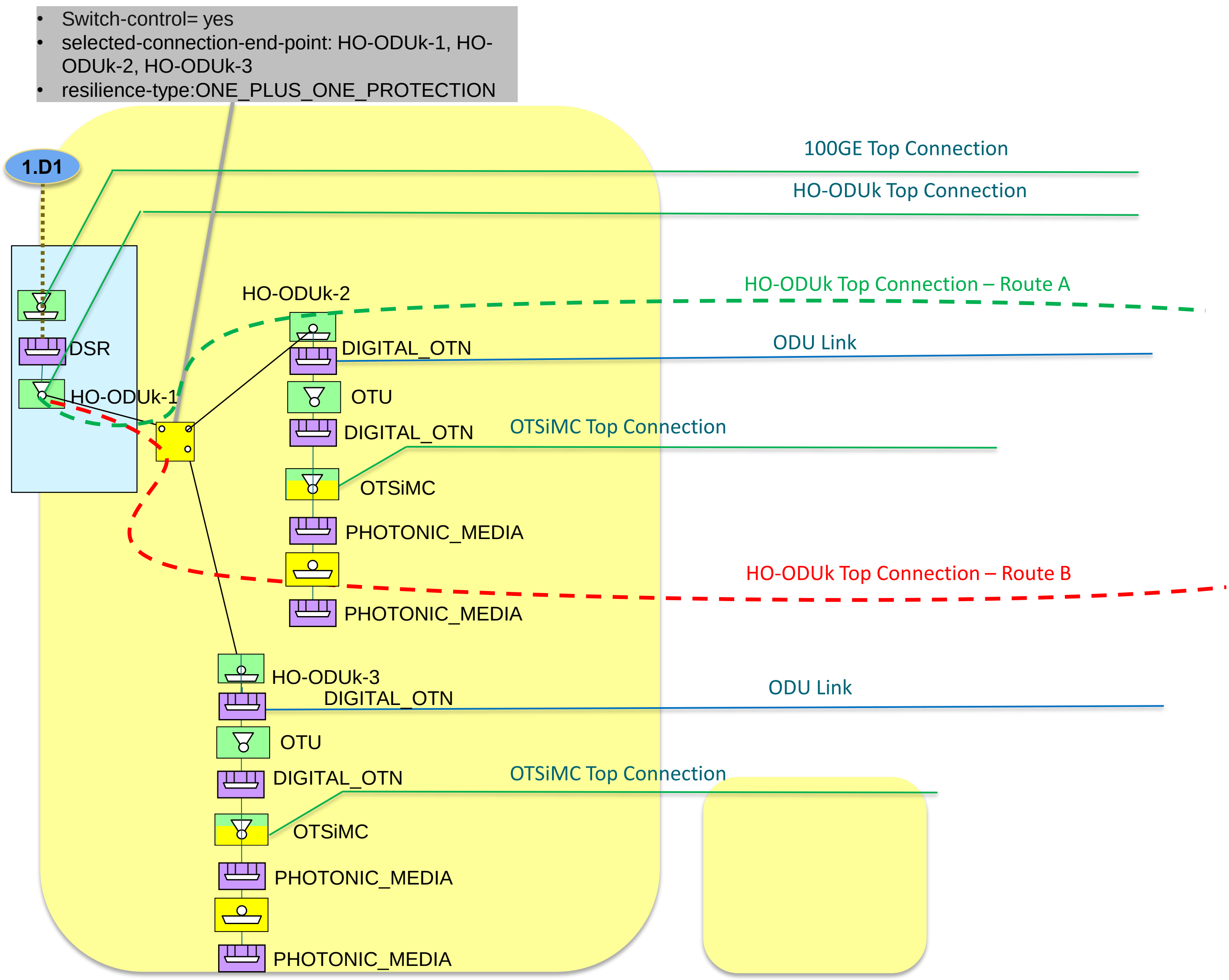


Figure 6-76 UC5c: eSNCP protection schema for HO-ODUk Top Connection

Once the CS is created, the TAPI Server is responsible of implementing the Switch control among the connections generated to support the protection schema. The requested DSR/ODU CS triggers the creation of:

- **A DSR Top Connection.**
- **An ODU Top Connection:** only addressing the ONE_PLUS_ONE case with no intermediate ODU cross-connections considered.
  - **selected-connection-end-points:** either HO-ODUk-2 or HO-ODUk-3 *in the receive direction*, according to the conditions.
    - There is no correlation between the switches at the protection scheme ends.
  - **selected-route:** Route-A (HO-ODU-Top-Connection-A) and Route-B (HO-ODU-Top-Connection-B)

6.4.4.2 Relevant Parameters

Table 64 complements the information included in the Use Case 1a and Use Case 5b definitions, with the Connectivity-Service parameters required implementing this use case.

Table 64: Connectivity-service parameters for UC5c.

	/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/resilience-constraint			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
resilience-type/protection-type	"ONE_PLUS_ONE_PROTECTION"	RW	M	• Provided by *tapi-client*
preferred-restoration-layer	If present, this leaf-list MUST be { "DIGITAL_OTN" }	RW	C	• Provided by *tapi-client*
hold-off-time	uint64 (ms)	RW	O	• Provided by *tapi-client*
max-switch-times	uint64	RW	O	• Provided by *tapi-client*
is-coordinated-switching-both-ends	[true, false]	RW	O	• Provided by *tapi-client*
is-lock-out	[true, false]	RW	O	• Provided by *tapi-client*
is-frozen	[true, false]	RW	O	• Provided by *tapi-client*

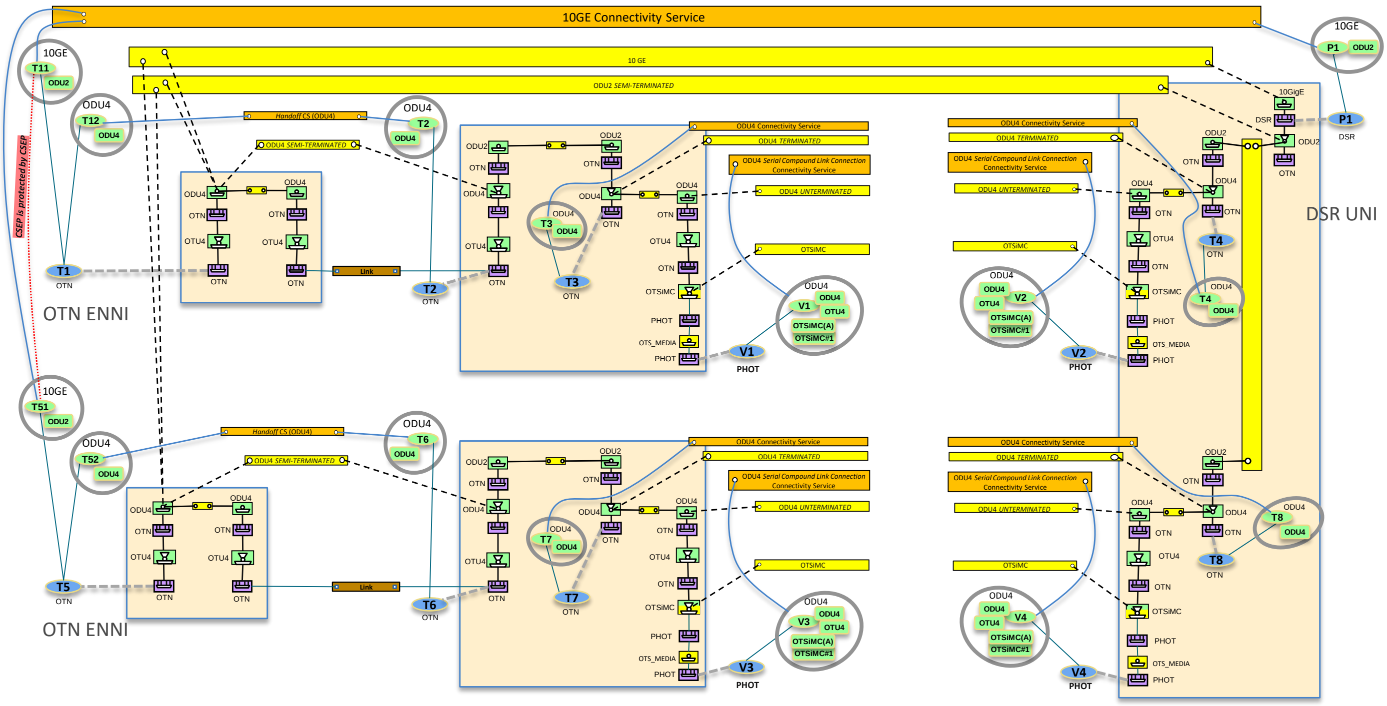
6.4.5 Use case 5d: 1+1 DSR/ODU protection with Diverse Service Provisioning (eSNCP) in Asymmetric scenarios

Number	**UC5d**
Name	1+1 DSR/ODU protection with Diverse Service Provisioning (eSNCP) in Asymmetric scenarios
Technologies involved	Optical
Process/Areas Involved	Planning and Operations
Brief description	**Disclaimer: This use case is in a draft state, the final definition will be completed based on the feedback provided by the industry upon this release of the reference specification.** This use case covers the provisioning of an asymmetric 1+1 protected connectivity-service implemented through eSNCP. This use case specifies the creation of a connectivity-service between

	UNI and E-NNI CSEPs, to support services which start in one network domain and hand-off to another network domain managed by a different TAPI Server (multi-domain scenario). The actual implementation of the ODU SNCP monitoring mode is out of the scope of this document, but it is assumed that at least the Path Monitoring (PM) OTN mechanism MUST be supported to provide end-to-end ODU protection switching for DSR client services. For this specific case, this implies the PM header is properly propagated across domains. The protection process MUST be triggered automatically by the TAPI server and the TAPI client MUST be notified about the service condition changes through the **tapi-notification** service (as defined in UCs 15X).
Layers involved	DIGITAL_OTN
Type	Resilience
Description & Workflow	The DSR connectivity-service is requested between one DSR UNI SIP and two DIGITAL_OTN E-NNI SIPs representing the boundary interfaces to handover the service signal towards the next domain. It requires the reservation of two disjoint routes at the ODU layer between transponder's line interfaces. The TAPI Client MUST explicitly state the E-NNI CSEPs ***protection-role*** attribute and include the **tapi-connectivity:connectivity-service/end-point/protecting-connectivity-service-end-point** (for involved primary CSEPs) attribute to define the relationship between the working and protection paths. The connectivity-service object MUST include the **tapi-connectivity:connectivity-service/resiliency-constraint/resilience-type/protection-type** attribute with **ONE_PLUS_ONE_PROTECTION** attribute value.

6.4.5.1 Detailed Workflow

The scenario assumes the boundary interfaces between network domains to be E-NNI OTUk interfaces which shall be modeled as DIGITAL_OTN NEPs with the "inter-domain-plug-id" identifier as described in UC0d. Note: the following figure describes a potential instance of this Use Case in which there are ODU4 SCLC with internal ODU2 switching showing the flexibility of the approach. Implementations for this US need only conform to externally visible behavior between the UNI and ENNI SIPs.



6.4.5.2 Connectivity Service request processing

The TAPI Client request MUST include the relevant parameters as shown. Note that the WORK/PROTECT CSEPs have "layer-protocol-name": "DSR" and "layer-protocol-qualifier": “10G” (or equivalent) .

```
{
"tapi-connectivity:connectivity-service": [
        {
            "end-point": [
                {
                    "direction": "BIDIRECTIONAL",
                    "protection-role": "PROTECTED",
                    "layer-protocol-qualifier": "tapi-dsr:DIGITAL_SIGNAL_TYPE",
                    "layer-protocol-name": "DSR",
                    "local-id": "end_point_1",
                    "service-interface-point": {
                        "service-interface-point-uuid":  "UUID1"
                    }
                },
               {
                    "direction": "BIDIRECTIONAL",
                    "protection-role": "WORK",
                    "layer-protocol-qualifier": "tapi-dsr:DIGITAL_SIGNAL_TYPE",
                    "layer-protocol-name": "DSR",
                    "local-id": "end_point_2",
                    "service-interface-point": {
                        "service-interface-point-uuid":  "UUID2" (OTN SIP)

                    }
                },
                {
                    "direction": "BIDIRECTIONAL",
                    "protection-role": "PROTECT",
                    "layer-protocol-qualifier": "tapi-dsr:DIGITAL_SIGNAL_TYPE ",
                    "layer-protocol-name": "DSR",
```



```
                    "local-id": "end_point_3",
                    "protecting-connectivity-service-end-point": {
                            "connectivity-service-uuid": "UUID_CS",    (this CS)
                            "connectivity-service-end-point-local-id": "end_point_2"
                    }
                    "service-interface-point": {
                        "service-interface-point-uuid": "UUID3" (OTN SIP)

                    }
                }
            ],
            "is-coordinated-switching-both-ends": false,
            "resilience-constraint":{
                 "resilience-type":{
                        "protection-type":ONE_PLUS_ONE_PROTECTION,
                  }
            }
            "uuid": "UUID_CS",
        }
    ]
}
```


6.4.5.3 Expected results

The TAPI server shall accept the above-mentioned asymmetric connectivity-service provisioning request and perform the route computation and connection provisioning within its domain.

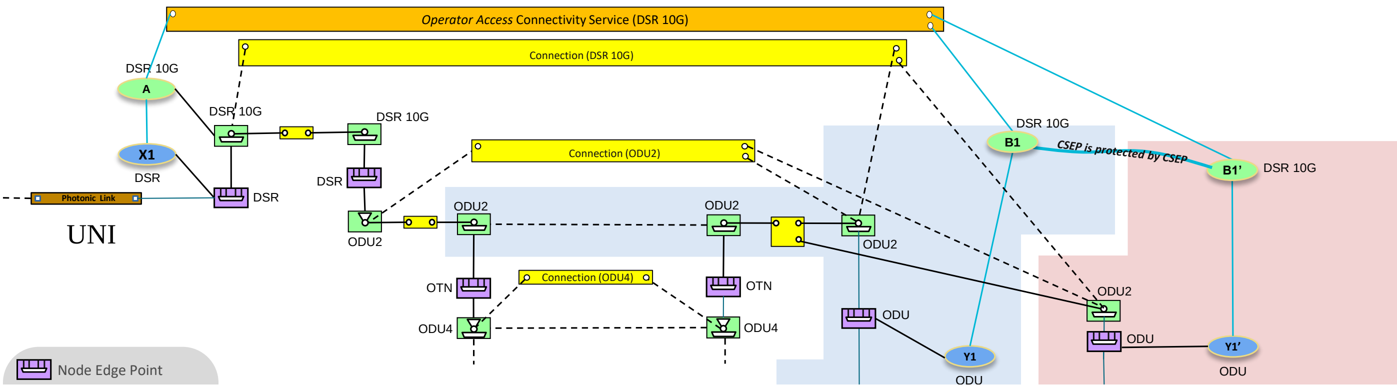


Figure 6-77 TAPI context after asymmetric connectivity-service with 1+1 Protection with Diverse Service Provisioning (eSNCP) provisioning between UNI DSR and E-NNI OTUk interfaces.

The example covers an asymmetric connectivity-service request between a 10GE DSR SIP (representing the UNI client interface in Domain A) and two DIGITAL_OTN SIP (representing the E-NNI inter-domain interface at the boundary between Domain A and B). The requested DSR CS triggers the creation of:

- **A DSR Top Connection.**
- **An ODU2 Top Connection**: which has two routes and includes one switch-control instance. Such switch-control instance recursively includes one subordinate switch-control (sub-switch-control) and no switch. The subordinate switch control references (*points to*) a switch-control included the appropriate (3-pointed) lower-connection (by using a connection-uuid and switch-control-uuid).
    - The 3-pointed lower-connection switch-control includes one switch instance.
    - The 3-pointed **ODU2 lower connection**: includes the switch control including the switch between working and protection CEPs [no sub-switch-control].

- **The additional ODU2 lower connections** without switch control.

In case of ONE_PLUS_ONE_PROTECTION:

- **selected-connection-end-points:** either one of the **ODU2 CEPs** is selected, *in the receive direction*, according to the conditions.
  - There is no correlation between the switches at the protection scheme ends.
- **selected-route:** both main route (A) and protecting route (B) are always selected.

6.4.6 Use case 6a: Dynamic restoration policy for connectivity services

Number	**UC6a**
Name	**Dynamic restoration policy for connectivity services**
Technologies involved	Optical
Process/Areas Involved	Planning and Operations
Brief description	This use case covers the provisioning of connectivity-services with restoration capabilities. The dynamic restoration capability can be requested at different layers. The TAPI client specifies two CSEPs including the restoration-type and protection-type parameters. The TAPI server is responsible for maintaining the SLA condition by configuring the dynamic restoration process. The TAPI server MUST notify the client about service condition changes through the tapi-notification service (as defined in UCs 15a and 15b). The restoration path is computed *after* the failure is detected. Additional constraints, such coroute-inclusion or diversity-exclusion, SHALL be considered as **loose constraints** at the time of the restoration occurs, i.e., applicable if possible.
Layers involved	DIGITAL_OTN, PHOTONIC_MEDIA
Type	Resilience
Description & Workflow	The connectivity service is requested between two CSEPs. The TAPI Client MAY include the ***tapi-connectivity:connectivity-service/resilience-constraint/preferred-restoration-layer*** object to specify the preferred restoration layer, but the final decision is responsibility of the TAPI server based on the current network conditions. The Connectivity Service MUST include the ***tapi-connectivity:connectivity-service/resilience-constraint/resilience-type/protection-type*** attribute with **DYNAMIC_RESTORATION** attribute value. This UC is implemented following the same workflow described in “Description & Workflow” of UC1.0. the following figure shows an example of the sequence of notifications that are generated by the TAPI server upon the failure.

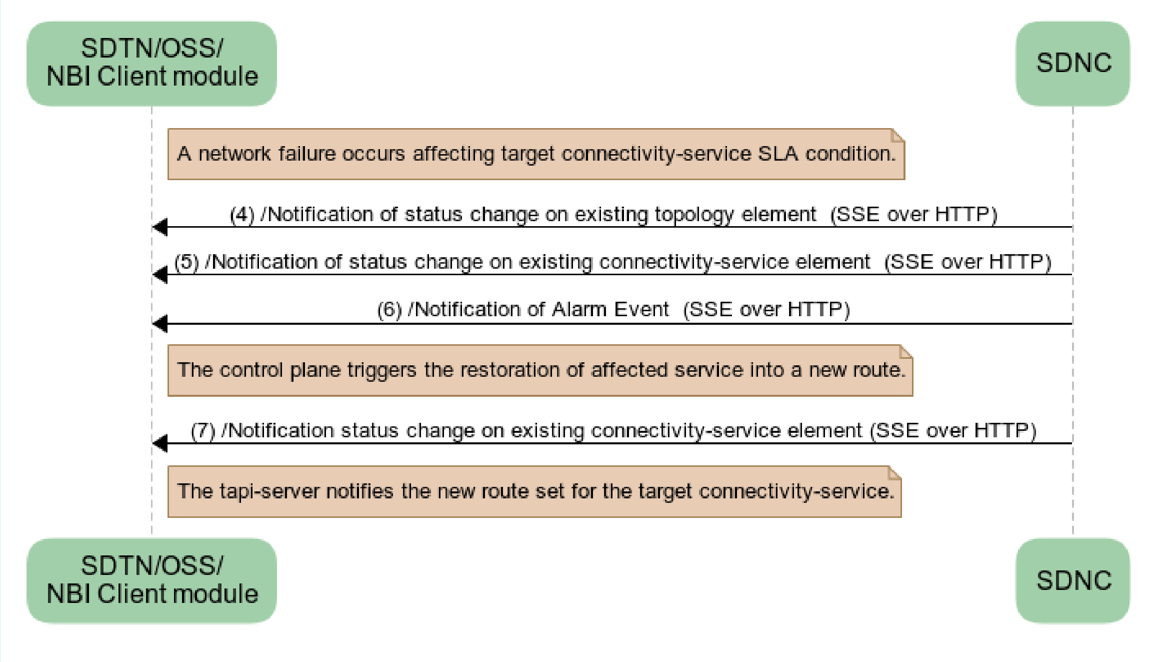


Figure 6-78 UC-6a: Resiliency workflow (note, the triggering of the restoration MAY happen prior to the notifications)

6.4.6.1 Relevant Parameters

Table 65 complements the information included in the Use Case 1.0 with the Connectivity-Service parameters required to implement this use case.

Table 65: Connectivity-service parameters for UC6a.

connectivity-service	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
resilience-constraint/resilience-type/protection-type	"DYNAMIC_RESTORATION"	RW	M	• Provided by *tapi-client*
resilience-constraint/preferred-restoration-layer	List of preferred restoration layers. This MAY include { "DIGITAL_OTN", "PHOTONIC_MEDIA" }	RW	O	• Provided by *tapi-client*
reversion-mode	One of { "REVERTIVE", "NON-REVERTIVE" }	RW	O	• Provided by *tapi-client* NOTE: Reversion modes for restoration (e.g., returning to the nominal path) is not specified.

6.4.7 Use case 6b: Pre-computed restoration policy for connectivity services.

Number	**UC 6b**
Name	Pre-computed restoration policy for connectivity services.
Technologies involved	Optical

Process/Areas Involved	Planning and Operations
Brief description	**Disclaimer: This use case is in a draft state, the final definition will be completed in a future release of this reference specification** This use case covers the provisioning of connectivity-services with restoration capabilities. It assumes the same definitions, workflow and specifications defined in UC6a. Additionally, the TAPI server MUST accept a pre-computed (preset) restoration path as part of the provisioning request (*). The TAPI server is responsible for maintaining the SLA condition by configuring the dynamic restoration process. The TAPI server MUST notify the client about service condition changes through the tapi-notification service (as defined in UCs 15a and 15b). (*) Please note this use case depends on the use case 12b.
Layers involved	ODU, PHOTONIC_MEDIA
Type	Resilience
Description & Workflow	In this UC, the nominal and restoration paths MUST be included within the **tapi-connectivity:connectivity-service/tapi-connectivity:topology-constraint/tapi-connectivity:include-path** attribute. Due to model limitations (the fact that the path list is not ordered), the specification of the nominal and restoration route SHOULD be done sequentially. 1) First the tapi-client includes the nominal path in the Connectivity-Service initial POST request. 2) Once the connectivity-service creation is completed, the tapi-client must modify the existing Connectivity-Service object by adding the pre-computed restoration path, into the **tapi-connectivity:connectivity-service/tapi-connectivity:topology-constraint/tapi-connectivity:include-path** leaf-list attribute. The TAPI user MUST use the PUT operation to modify the existing object (add a second item to the list) The Connectivity Service object sent to the SDN-C MUST include the tapi-**connectivity:connectivity-service/tapi-topology:resilience-type/protection-type** attribute with PRE_COMPUTED_RESTORATION attribute value. <u>Resiliency workflow</u>: The UC's protection workflow, of this UC, follows the same workflow defined in the "Procedure" section of UC6a.

6.4.7.1 Relevant Parameters

Table 66 complements the information included in the Use Case 1a and Use Case 5b definitions, with the Connectivity-Service parameters required to implement this use case.

Table 66: Connectivity-service parameters for UC6b.

connectivity-service	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
See parameters of UC6a				

include-path	After the RESTCONF POST, the list SHALL contain path uuids of the 1) nominal path and 2) restoration path.	RW	M	• Provided by *tapi-client*

6.4.8 Use case 7a: Dynamic restoration and 1+1 protection of DSR/ODU unconstrained service provisioning.

Number	**UC7a**
Name	Dynamic restoration and 1+1 protection of DSR/ODU unconstrained service provisioning
Technologies involved	Optical
Process/Areas Involved	Planning and Operations
Brief description	This use case covers the provisioning of connectivity-services with restoration capabilities and 1+1 protection capabilities. The 1+1 protection scheme can be implemented either: • Over the MC/PHOTONIC_MEDIA layer as the OLP Protection scheme defined in UC5b • Over the ODU/DIGITAL_OTN layer as the eSNCP protection scheme defined in UC5c. This use case introduces a *second level of resilience*, which is implemented through dynamic restoration *of the first connection affected by a failure*. The Connectivity-Service can be requested at different layers i.e., DSR, DIGITAL_OTN. The TAPI client specifies two CSEPs as well as the restoration-type and protection-type parameters. The TAPI server is responsible for maintaining the SLA condition by configuring the protection and dynamic restoration process. The TAPI server MUST notify the client about service condition changes through the tapi-notification service (as defined in UCs 15a and 15b). The restoration path is computed after the failure is detected. Additional constraints, such as **coroute-inclusion** or **diversity-exclusion** SHALL be considered as **loose constrains** at the time of the restoration occurs, i.e., applicable if possible. *This use case implies that the system needs to account for a single failure: only the first affected connection needs to be dynamically restored. In case of a second failure, the service is still protected by the 1+1 capability, but no further dynamic restorations are required. The ability to support multiple failures belongs to UC8.*
Layers involved	DSR, DIGITAL_OTN, PHOTONIC_MEDIA
Type	Resilience
Description & Workflow	This UC is implemented following the same workflow described in UC5b/5c but the Connectivity Service object MUST include ***tapi-connectivity:connectivity-service/tapi-topology:resilience-type/protection-type*** attribute with **ONE_PLUS_ONE_PROTECTION_WITH_DYNAMIC_RESTORATION**. Resiliency workflow: The UC assumes that the service with this SLA is able to support a failure affecting the nominal or protection paths (via protection switching) and, after the failure, to maintain the

	1+1 protection by dynamically restoring the affected path (which may imply a wavelength change). 1) Protection switching is described in UC5b 2) The dynamic restoration is described in UC6a

6.4.8.1 Relevant Parameters

Table 67 complements the information included in the Use Case 5b definitions, with the Connectivity-Service, Connectivity-Service-End-Points, Connections and Switch-control, parameters required to implement this use case.

Table 67: Connectivity-service parameters for UC7a.

resilience-constraint	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/resilience-constraint**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
resilience-type/protection-type	"ONE_PLUS_ONE_PROTECTION_WITH_DYNAMIC_RESTORATION"	RW	M	• Provided by *tapi-client*
preferred-restoration-layer	List of preferred restoration layers. This MAY include { "DIGITAL_OTN", "PHOTONIC_MEDIA" }	RW	O	• Provided by *tapi-client*
hold-off-time	uint64 (ms)	RW	O	• Provided by *tapi-client*
max-switch-times	uint64	RW	O	• Provided by *tapi-client*
is-coordinated-switching-both-ends	[true, false]	RW	O	• Provided by *tapi-client*
is-lock-out	[true, false]	RW	O	• Provided by *tapi-client*
is-frozen	[true, false]	RW	O	• Provided by *tapi-client*

6.4.9 Use case 7b: Pre-Computed restoration policy and 1+1 prot. of DSR/ODU unconstrained service prov.

Number	**UC7b**
Name	**Pre-Computed restoration policy and 1+1 protection of DSR/ODU unconstrained service provisioning.**
Technologies involved	Optical
Process/Areas Involved	Planning and Operations
Brief description	This use case covers the provisioning of connectivity-services with restoration capabilities and 1+1 protection capabilities and the demonstration of the effective protection and restoration process when a failure occurs.
	***NOTE**: As in UC6b, TAPI does not currently allow to specify explicit paths for each of the working/protecting and restoration path(s) unambiguously (the include-path list of path uuids does not allow ordering constraints).* *As a consequence, this UC only covers the scenario where the working and protected paths are dynamically provisioned (unconstrained) and the included-path refers to the common restoration path to be used by the failed path. Implementations supporting ONE_PLUS_ONE_PROTECTION_WITH_PRE_COMPUTED_RESTORATION MUST be aware that the include-path uuid does not define the path for the working/protecting connections.*

	Implementations MUST ensure that the routes used in the 1+1 protection connection are disjoint from the provided path-uuid. The 1+1 protection scheme can be implemented either: • Over the PHOTONIC_MEDIA layer as the OLP Protection scheme defined in UC5b • Over the DIGITAL_OTN layer as the eSNCP protection scheme defined in UC5c Additionally, the SDN-C MUST accept a pre-computed (preset) restoration path as part of the connectivity-service provisioning request (*). This use case introduces a *second level of resilience*, which is implemented through pre-computed restoration of the first connection affected by a failure. The Connectivity-Service can be requested at different layers i.e., DSR, ODU. The TAPI client specifies two CSEPs as well as the restoration-type and protection-type parameters. The TAPI server is responsible for maintaining the SLA condition by configuring the protection and dynamic restoration process. The TAPI server MUST notify the client about service condition changes through the tapi-notification service (as defined in UCs 15a and 15b). (*) Please note this use case depends on the use case 12b.
Layers involved	DSR, DIGITAL_OTN, PHOTONIC_MEDIA
Type	Resilience
Description & Workflow	The initial path computation procedure follows UC12b. The UC service provisioning and protection procedures follow the same workflow defined in the "Procedure" section of UC7a. Resiliency workflow: The UC assumes that the service with this SLA is able to support a failure affecting the nominal or protection paths (via protection switching) and, after the failure, to maintain the 1+1 protection by restoring through the pre-computed path (which may imply a wavelength change). 1) Protection switching is described in UC5b 2) The pre-computed restoration is described in UC6b

6.4.9.1 Relevant Parameters

Table 68 complements the information included in the Use Case 5b definitions, with the Connectivity-Service, Connectivity-Service-End-Points, Connections and Switch-control, parameters required to implement this use case.

Table 68: Connectivity-service parameters for UC7b.

	/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/resiliency-constraint			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
resilience-type/protection_type	"ONE_PLUS_ONE_PROTECTION_WITH_PRE_COMPUTED_RESTORATION"	RW	M	• Provided by *tapi-client*

preferred-restoration-layer	List of preferred restoration layers. This MAY include { "DIGITAL_OTN", "PHOTONIC_MEDIA" }	RW	O	• Provided by *tapi-client*
hold-off-time	uint64 (ms)	RW	O	• Provided by *tapi-client*
max-switch-times	uint64	RW	O	• Provided by *tapi-client*
is-coordinated-switching-both-ends	[true, false]	RW	O	• Provided by *tapi-client*
is-lock-out	[true, false]	RW	O	• Provided by *tapi-client*
is-frozen	[true, false]	RW	O	• Provided by *tapi-client*

6.4.10 Use case 8: Permanent protection 1+1 for use cases

Number	**UC8**
Name	Permanent protection 1+1 for use cases
Technologies involved	Optical
Process/Areas Involved	Planning and Operations
Brief description	This use case extends UC7a by allowing an indeterminate number of failures to affect either of the 1+1 routes and the respective subsequent dynamic restorations (note that UC7a only covered the first failure).
Layers involved	DSR, DIGITAL_OTN, PHOTONIC_MEDIA
Type	Resilience
Description & Workflow	The Connectivity Service object MUST include the resilience-type/protection-type attribute with PERMANENT_ONE_PLUS_ONE_PROTECTION attribute value.

6.4.10.1 Relevant Parameters

The following table complements the information included in the Use Case 1a and Use Case 5b definitions, with the Connectivity-Service, Connectivity-Service-End-Points, Connections and Switch-control, parameters required to implement this use case.

Table 69: Connectivity-service parameters for UC8 (same as of 7a).

	/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/resilience-constraint			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
resilience-type/protection-type	"PERMANENT_ONE_PLUS_ONE_PROTECTION"	RW	M	• Provided by *tapi-client*

6.4.11 Use case 9: Reverted protection

Number	**UC9**
Name	Reverted protection
Technologies involved	Optical

Process/Areas Involved	Planning and Operations
Brief description	This use case covers the behavior of the system as defined in UCs 5a-5d, 7a, 7b and 8 with the different reversion modes.
Layers involved	DSR, DIGITAL_OTN, PHOTONIC_MEDIA
Type	Resilience
Description & Workflow	This use case intends to extend UCs 5a-5d, 7a, 7b and 8, thus the workflow to be implemented shall be the same regarding each specific use case.

6.4.11.1 Relevant Parameters

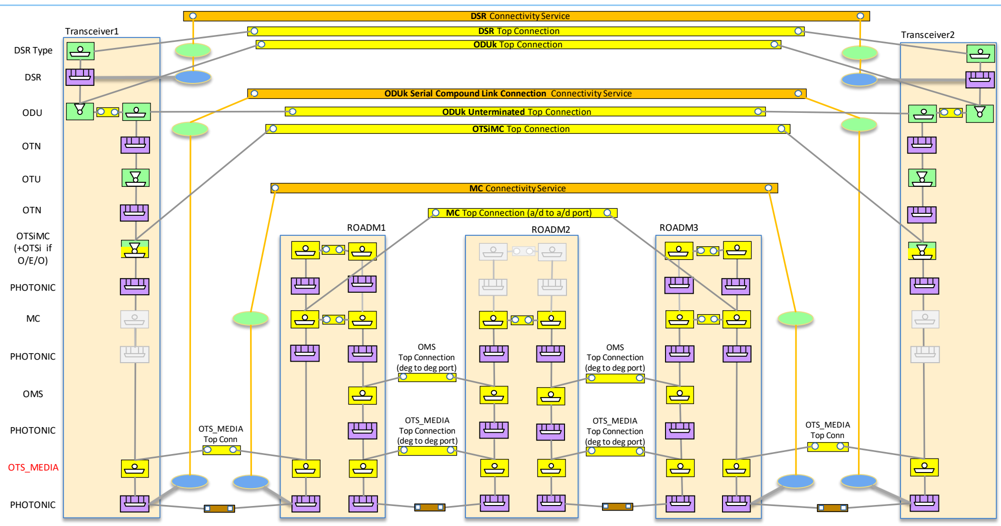
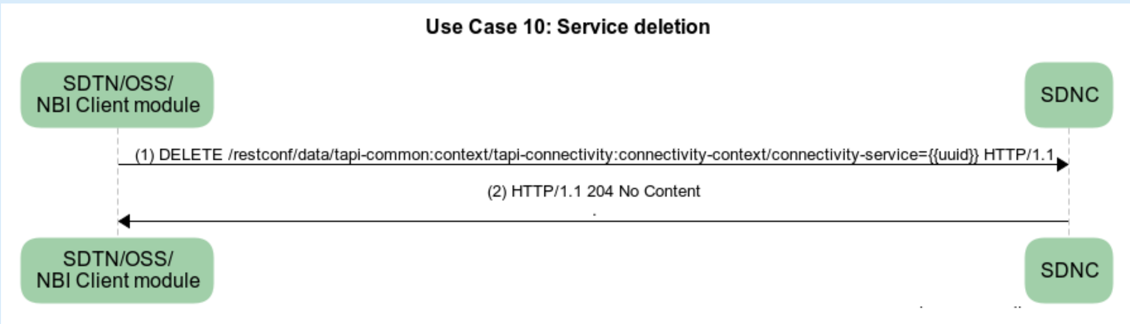
See the introduction to the Section 6.4 for the relevant parameters.

6.5 Maintenance

6.5.1 Use Case 10: Service deletion (applicable to all previous use cases)

Number	**UC10**
Name	**Service deletion (applicable to all previous use cases)**
Technologies involved	Optical
Process/Areas Involved	Planning and Operations
Brief description	This use case covers the deletion of a connectivity-service and specifies the rules by which the supporting connection(s) are also deleted. In this RIA this means the following considerations: 1. As detailed in Section 6.2.1, the provisioning of a connectivity service MAY trigger the instantiation of additional connectivity services, which MUST appear in connectivity context with a server allocated UUID [**server-allocated connectivity-service**] [*Note that the allocation by the server of connectivity services enables direct management, modification, and deletion*]. In consequence, a TAPI client is allowed to delete server allocated connectivity-services provided that such operation is consistent with the next considerations. 2. This RIA only considers server-allocated connectivity-services that have been allocated as a side-effect of a client driven connectivity service provisioning. 3. Because of the connectivity service(s) instantiation, a number of *supporting connections* [TAPI-CONN-MODEL-REQ-1] and the corresponding related NEPs and CEPs will have been created or configured. Further connectivity service(s) provisioning/deletion MAY modify such connections. 4. Connections which have been allocated by the server (were not created upon the provisioning of a connectivity service) cannot be deleted by a user operation [**pre-existing connections**]. For example, OMS/OTS connections are assumed pre-existing.

	5. DEFINITION: For a given CS supporting connections can be pre-existing or not. When a non-preexisting connection is supporting more than one Connectivity Service, we say those connectivity services have *shared-ownership* of the connection. If such connection is supporting only one Connectivity Service, we shay such connectivity service has *exclusive-ownership* of the connection [**connection ownership**]. The concept of ownership is related to connection deletion. 6. The deletion of a connectivity service (either the client provisioned ones or the server allocated ones) MAY trigger the deletion of any supporting server allocated connectivity services. [**chained deletion**] 7. Since it has been established that a server-allocated connectivity service is always a result of a provisioning process, a connectivity service lifetime is always ended with a TAPI-Client driven delete operation. In other words, the deletion of a CS is a result of a delete procedure and any connectivity service that has been allocated directly or indirectly by the server CANNOT be deleted by the server autonomously. We acknowledge that in scenarios not foreseen by this RIA, such requirements MAY not apply, and additional policies may be defined allowing the autonomous creation and deletion of server-allocated connectivity services [*Note that deletion of a server CS that is supporting client CS MUST fail, as detailed next*] 8. As per the definition in 6, the deletion of a connectivity service MUST cause the deletion of all supporting connections and associated server-allocated Connectivity Services that are exclusively supporting the connectivity service and are not *pre-existing connections*. This implies that there are no orphan connections if they were created upon the provisioning of a connectivity service [**no orphan connections**]. a. For example, the provisioning of a connectivity service ODU2-S1 MAY cause either 1) the instantiation of a top-level connection ODU2-C and a supporting connection ODU4-C or 2) the instantiation of a top-level connection ODU2-C, a supporting connection ODU4-C and a server allocated connectivity service ODU4-S2. In the second case, the deletion of ODU2-S1 MUST NOT cause the deletion of ODU4-C since its ownership is *shared by* ODU2-S1 and ODU4-S2 (ODU4-C is a supporting connection of both connectivity services). Let us note that it is also possible to delete ODU4-S2 prior to the deletion of ODU2-S1. In such case ODU4-C will exclusively support ODU2-S1 upon deletion of ODU4-S2. b. As a second example, consider the figure below. At time X, an ODUk Unterminated CS (and its top-connection) indicates that there is an infrastructure service, and the user may request additional client services using it. At time X + 1, the client establishes the DSR connectivity service, which triggers the instantiation of the ODUk (terminated) top-connection. Note that, following the RIA guidelines, it is possible to remove the *ODUk Serial Compound Link Connection Connectivity Service* and, consequently, its top-connection is removed, since it is not supporting the DSR CS (instead, the terminated ODUk top connection is) and there is no intermediate partitioning between top-connections (the unterminated top-connection is not included in the list of the terminated top-connnection's lower connections). Note that any cross-connection that was supporting the unterminated top-connection (e.g., 3R regeneration cases) MUST remain, since it is still supporting the ODUk top-connection.

	 9. A TAPI server, upon deletion of a Connectivity Service, SHALL make sure that no other connectivity service has dangling references (e.g., `peer-fwd-connectivity-service-end-point`, `protecting-connectivity-service-end-point`, `server-connectivity-service-end-point`, `coroute-inclusion`, `diversity-exclusion`). For example, it MUST be possible to delete connectivity service CS1 despite it was referred to during provisioning of the connectivity service CS2 (in its coroute-inclusion or diversity-exclusion) and the server MUST remove such reference (by deleting the coroute-inclusion or diversity-exclusion data node). [*Note: it is acknowledged that this behavior can also be accomplished by using "require-instance false" statement in the corresponding leafrefs to connectivity service uuids. This is for further study*] If the provided CS UUID does not exist, the server MUST return a "404 Not Found" status-line. The error-tag value "invalid-value" is returned in this case. If the DELETE request succeeds, a "204 No Content" status-line is returned.
Layers involved	DSR, DIGITAL_OTN, PHOTONIC_MEDIA
Type	Maintenance
Description & Workflow	The TAPI client MUST specify the ***tapi-connectivity:connectivity-service/uuid*** attribute in the RESTCONF DELETE request to identify the service to be removed.  Figure 6-79 UC-10: Service Deletion workflow.

6.5.2 Use Case 11a: Modification of service path

Number	**UC11a**
Name	**Modification of service path**
Technologies involved	Optical
Process/Areas Involved	Planning and Operations
Brief description	**Disclaimer: This use case is in a draft state, the final definition will be completed in a future release of this reference specification** This use case covers the modification of an existing connectivity-service path. The way TAPI is defined only allows service modification through connectivity-service modification, thus the implementation of this use case is based on the implicit modification of the existing connections composing an existing connectivity-service and not through explicit modification of the connection objects. The objective of this use case is to allow the TAPI client to be able to modify an existing connectivity-service route for several purposes, obviously here we won't motivate all, but some examples might be: • To optimize the network resources allocation. • To exclude a route's node or link to realize a maintenance operation. • To fix avoid a unique point of failure among other related services (SRGs). Thus, the ways to modify an existing path may be different depending on the specific motivation or intend. The TAPI connectivity-service allows the following explicit path's constrains definitions into the connectivity-service object which can be exploited in this use case to infer a path modification: `| +--rw coroute-inclusion tapi-common:uuid` `| +--rw diversity-exclusion* [connectivity-service-uuid]` `| +--rw include-path* tapi-common:uuid` `| +--rw exclude-path* tapi-common:uuid` `| +--rw include-link* tapi-common:uuid` `| +--rw exclude-link* tapi-common:uuid` `| +--rw include-node* tapi-common:uuid` `| +--rw exclude-node* tapi-common:uuid` All these constrains can be modified or add to an existing service. The implementation details shall follow the same guidelines described in UCs 3a, 3b, 3c. Moreover, the route-objective-function attribute can also be added or modified to an existing service to infer an implicit route selection by the TAPI server to accommodate service needs: `+--rw route-objective-function` The TAPI server behavior for accommodating different route-objective-functions is defined in UCs 3e, 3f. A pre-requisite for the implementation of this use case is that the

	administrative-state of the target connectivity-service is set of "UNLOCKED", in case of any other value for this attribute, the TAPI server MUST reject the TAPI client request. The modification of an existing service must be done through a HTTP PUT request over an existing connectivity-service object by specifying its unique universal identifier UUID attribute in the request. As per [RFC8040] and consistent with [RFC7231], if the PUT request creates a new resource, a "201 Created" status-line is returned. If an existing resource is modified, a "204 No Content" status-line is returned. The usage of HTTP PATCH is for further study.
Layers involved	DSR, DIGITAL_OTN, PHOTONIC_MEDIA
Type	Maintenance
Description & Workflow	The TAPI client MUST specify the **tapi-connectivity:connectivity-service/uuid** attribute in the RESTCONF PUT request to identify the service to be modified.

6.5.3 Use Case 11b: Modification of service nominal path to secondary (prot.) path for maintenance operations

Number	**UC11b**
Name	**Modification of service nominal path to secondary (protection) path for maintenance operations.**
Technologies involved	Optical
Process/Areas Involved	Planning and Operations
Brief description	**Disclaimer: This use case is in a draft state, the final definition will be completed in a future release of this reference specification** This use case covers the modification of an existing connectivity-service path. The way TAPI is defined only allows service modification through connectivity-service modification, thus the implementation of this use case is based on the implicit modification of the existing connections composing an existing connectivity-service and not through explicit modification of the connection objects. The objective of this use case is to allow the TAPI client to be able to modify an existing connectivity-service work and protection route roles. This modification implies a change on the switching conditions of the underlying connections implementing the **tapi-connectivity:connection/switch** objects which represent the control configuration. In order to perform a change between the work and protection connection roles, the TAPI client shall use the **tapi-connectivity:connectivity-service/end-point/protecting-connectivity-service-end-point** and **protection-role** parameters: `module: tapi-connectivity` `  augment /tapi-common:context:` `    +--rw connectivity-context` `       +--rw connectivity-service* [uuid]` `       |  +--rw end-point* [local-id]`

	`` | | +--rw protecting-connectivity-service-end-point | | | +--rw connectivity-service-uuid? -> /tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/uuid | | | +--rw connectivity-service-end-point-local-id? -> /tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/end-point/local-id | | +--rw protection-role? protection-role ``` The expected result is that TAPI client will send a modify TAPI request which will include a change between the roles of the working and protecting CSEPs of a given connectivity-service already provisioned in the network. As a response, the TAPI Server MUST perform the required changes into the underlying connections' switch control to route the traffic through the new "Working" connection route. A pre-requisite for the implementation of this use case is that the **administrative-state** of the target connectivity-service is set of "UNLOCKED", in case of any other value for this attribute, the TAPI server MUST reject the TAPI client request. The modification of an existing service must be done through a HTTP PUT request over an existing connectivity-service object by specifying its unique universal identifier UUID attribute in the request. The client MUST provide the complete connectivity service object in the PUT. A request message-body MUST be present, representing the new data resource, or the server MUST return "400 Bad Request" status-line. The error-tag value "invalid-value" is used in this case. Consistent with [RFC7231], if an existing resource is modified, a "204 No Content" status-line is returned.
Layers involved	DSR, DIGITAL_OTN, PHOTONIC_MEDIA
Type	Maintenance
Description & Workflow	The TAPI client MUST specify the **tapi-connectivity:connectivity-service/uuid** attribute in the RESTCONF PUT request to identify the service to be modified.

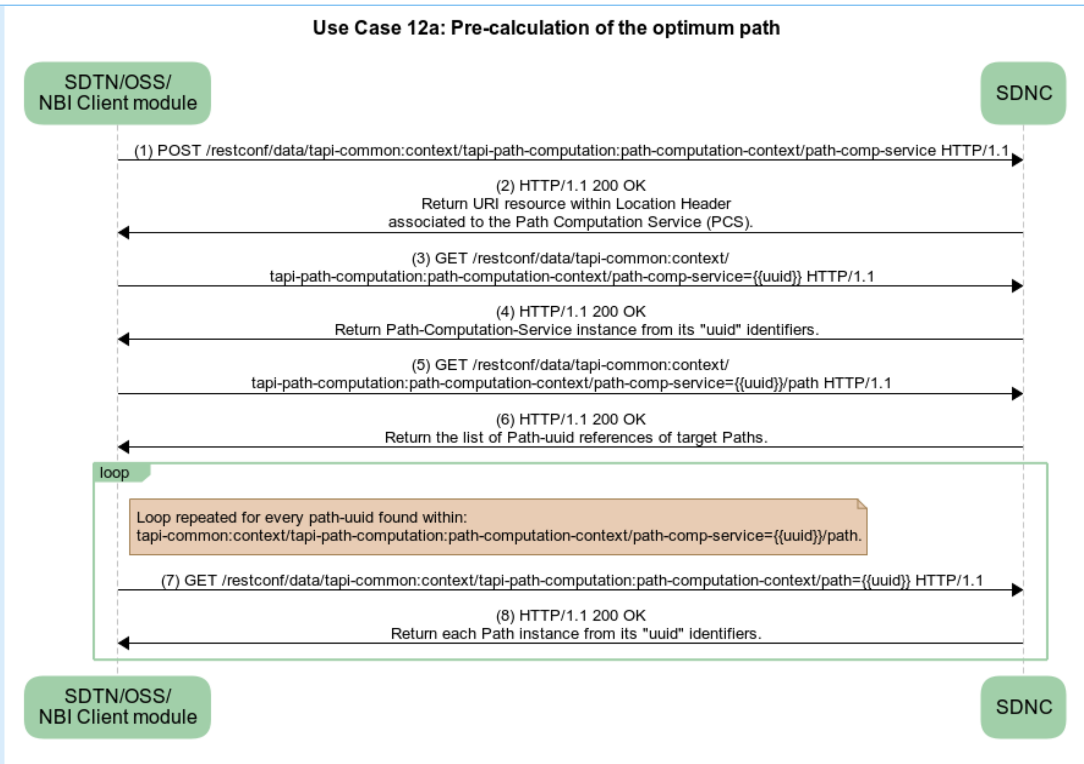
6.5.4 Use Case 11c: Setting SIP administrative state

This UC is for further consideration, including change of state of existing Connectivity Services referring to locked SIPs.

6.6 Planning

6.6.1 Use case 12a: Path Computation

Number	**UC12a**
Name	**Path Computation**
Technologies involved	Optical

Process/Areas Involved	Planning and Operations
Brief description	**Disclaimer: This use case is in a draft state, the final definition will be completed in a future release of this reference specification** This use case covers requesting a *path computation service*, which causes the computation of one or more TAPI paths, to be used as a routing constraint for connectivity services provisioning. The path computation service is instantiated upon request of the client and is requested between two path computation endpoints from a given protocol and layer qualifier (i.e., DSR, ODU, OTU, OTSiMC, MC). The path computation service request MAY include routing policies (i.e., min. hops, min. latency) and additional constrains (the same applicable to the creation of services i.e., use cases 3). In TAPI, paths are a sequence of links. Generally, the resulting paths MAY include regeneration (3R). NOTE: The policy affecting the instantation of link objects upon the instantiation of connections is not specified in this RIA.
Layers involved	DSR, DIGITAL_OTN, PHOTONIC_MEDIA
Type	Planning
Description & Workflow	 Figure 6-80 UC-12a: Pre-calculation of the optimum path workflow. Note: Step (5) assumes the server supports a GET operation on a list node. Alternatively, it can be of the form :


```
GET ..../path-comp-service={{uuid}}?fields=path(path-uuid)
```

6.6.1.1 Relevant Parameters

Table 70: Path-computation-context parameters.

path-computation-context				
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
path-comp-service	List of {**path-comp-service**}	RW	M	• Provided by *tapi-client*
path	List of {**path**}	RO	M	• Provided by *tapi-server*

Table 71: path-comp-serv object's parameters.

path-comp-serv				
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
end-point	List of {**path-service-end-point**}	RW	M	• Provided by *tapi-client*
routing-constraint	{ **routing-constraint** }	RW	M	• Provided by *tapi-client* • For details, see Table 74
topology-constraint	{List **topology-constraint** }	RW	M	• Provided by *tapi-client* • For details, see Table 73
objective-function	{**objective-function**}	RW	M	• Provided by tapi-client • For details, see Table 75
optimization-constraint	{**optimization-constraint**}	RW	O	• Provided by tapi-client • For details, see Table 76
direction	BIDIRECTIONAL or UNIDIRECTIONAL	RW	M	• Provided by tapi-client
layer-protocol-name	Applicable LPN			•
uuid	As per RFC4122	RW	M	• Provided by *tapi-client*
path	List of path uuid references	RO	M	• Provided by tapi-server

Table 72: Path-service endpoint (PSEP) object's parameters.

path-service-end-point (PSEP)				
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
local-id	"[0-9a-zA-Z_]{32}"	RW	M	• Provided by *tapi-client*
layer-protocol-name	Applicable LPN	RW	M	• Provided by *tapi-client*
layer-protocol-qualifier	Applicable LPQ	RW	M	• Provided by *tapi-client* • All children identities MUST be supported depending on hardware capabilities.
direction	One of ["BIDIRECTIONAL", "INPUT", "OUTPUT"]	RW	O	• Provided by *tapi-client*
role	One of ["SYMMETRIC", "ROOT", "LEAF", "TRUNK" or "UNKNOWN"]	RW	O	• Provided by *tapi-client* • Support only P2P and SYMMETRIC
capacity	"total-size": {value: unit} • "value": "[0-9]{8}", • "unit": see note	RW	O	• Provided by *tapi-client* • Unit depends on layer.

service-interface-point	*"/tapi-common:context/service-interface-point/uuid*"	RW	M	• Provided by *tapi-client*

Table 73: Topology constraint object's parameters.

topology-constraint				
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
include-topology	LeafList of topology uuids	RW	O	• This is a loose constraint - that is it is unordered and could be a partial list
avoid-topology	LeafList of topology uuids	RW	O	• This is a loose constraint - that is it is unordered and could be a partial list
include-path	LeafList of path uuids	RW	M	• Provided by *tapi-client* • This is a loose constraint - that is it is unordered and could be a partial list. • The uuid MUST refer to a valid { **tapi-path-computation:path}** object present within the tapi-server datastore.
exclude-path	LeafList of path uuids	RW	M	• Provided by *tapi-client* • This is a loose constraint - that is it is unordered and could be a partial list. • The uuid MUST refer to a valid {**tapi-path-computation:path}** object present within the tapi-server datastore
include-node	LeafList of node uuids	RW	M	• Provided by *tapi-client* • This is a loose constraint - that is it is unordered and could be a partial list. • The uuid MUST refer to a valid {**tapi-topology:node**} object present within the tapi-server datastore
exclude-node	LeafList of node uuids	RW	M	• Provided by *tapi-client* • Reference to an existing node-id already present in the TAPI server context MUST be valid. • The uuid MUST refer to a valid {**tapi-topology:node**} object present within the tapi-server datastore
include-link	LeafList of link uuids	RW	M	• Provided by *tapi-client* • This is a loose constraint - that is it is unordered and could be a partial list • The uuid MUST refer to a valid {**tapi-topology:link**} object present within the tapi-server datastore
exclude-link	LeafList of link uuids	RW	M	• Provided by *tapi-client* • This is a loose constraint - that is it is unordered and could be a partial list • The uuid MUST refer to a valid {**tapi-topology:link**} object present within the tapi-server datastore
preferred-transport-layer	One of [DIGITAL_OTN, PHOTONIC_MEDIA]	RW	M	• Provided by *tapi-client*

Table 74: Routing constraint object's parameters.

routing-constraint				
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
cost-characteristic	Includes{ cost-name, cost-value, cost-algorithm } • "cost-name": "string", • "cost-value": "string", • "cost-algorithm": "string",	RW	O	• Provided by *tapi-client*
latency-characteristic	Includes { traffic-property-name, fixed-latency-characteristic, queuing-latency-characteristic, jitter-characteristic, wander-characteristic } • "traffic-property-name": "string", • "fixed-latency-characteristic": "string", • "queing-latency-characteristic": "string", • "jitter-characteristic":"string" • "wander-characteristic":"string"	RW	O	• Provided by *tapi-client*
risk-diversity-characteristic	Includes { risk-characteristic-name, risk-identifier-list} • risk-characteristic-name • risk-identifier-list	RW	O	• Provided by *tapi-client*
diversity-policy	{SRLG, SRNG, SNG,NODE, LINK}	RW	O	• Provided by *tapi-client*
route-objective-function	One of ["MIN_WORK_ROUTE_HOP", "MIN_WORK_ROUTE_COST", "MIN_WORK_ROUTE_LATENCY", "MIN_SUM_OF_WORK_AND_PROTECTION_ROUTE_HOP", "MIN_SUM_OF_WORK_AND_PROTECTION_ROUTE_COST", "MIN_SUM_OF_WORK_A ND_PROTECTION_ROUTE_LATENCY", "LOAD_BALANCE_MAX_UNUSED_CAPACITY"	RW	M	• Provided by tapi-client
route-direction	One of ["BIDIRECTIONAL", "INPUT", "OUTPUT"]	RW	M	• Provided by *tapi-client*
is-exclusive	Boolean	RW	O	• Provided by *tapi-client*

Table 75: Objective function object's parameters.

objective-function				
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
bandwidth-optimization	One of ["MINIMIZE", "MAXIMIZE", "ALLOW", "DISALLOW", "DONT_CARE"]	RW	O	• Provided by *tapi-client*
concurrent-paths	One of ["MINIMIZE", "MAXIMIZE", "ALLOW", "DISALLOW", "DONT_CARE"]	RW	O	• Provided by *tapi-client*

cost-optimization	One of ["MINIMIZE", "MAXIMIZE", "ALLOW", "DISALLOW", "DONT_CARE"]	RW	O	• Provided by *tapi-client*
link-utilization	One of ["MINIMIZE", "MAXIMIZE", "ALLOW", "DISALLOW", "DONT_CARE"]	RW	O	• Provided by *tapi-client*
resource-sharing	One of ["MINIMIZE", "MAXIMIZE", "ALLOW", "DISALLOW", "DONT_CARE"]	RW	O	• Provided by *tapi-client*
local-id	"[0-9a-zA-Z_]{32}"	RW	M	• Provided by *tapi-client*
name	"value-name": "OBJ_FUNCTION" "value": " [0-9a-zA-Z_]{64}"	RW	M	• Provided by *tapi-client*

Table 76: Optimization-constraint object's parameters.

optimization-constraint				
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
traffic-interruption	One of {"ALLOW", "DISALLOW" }	RW	M	• Provided by *tapi-client*
local-id	"[0-9a-zA-Z_]{32}"	RW	M	• Provided by *tapi-client*
name	"value-name": "OPT_CONSTRAINT_NAME" "value": " [0-9a-zA-Z_]{64}"	RW	M	• Provided by *tapi-client*

6.6.2 Use case 12b: Simultaneous pre-calculation of two disjoint paths

Number	**UC12b**
Name	**Simultaneous pre-calculation of two disjoint paths**
Technologies involved	Optical
Process/Areas Involved	Planning and Operations
Brief description	**Disclaimer: This use case is in a draft state, the final definition will be completed in a future release of this reference specification** This UC extends 12a to support simultaneous computation of 2 or more paths.
Layers involved	DSR, DIGITAL_OTN, PHOTONIC_MEDIA
Type	Planning
Description & Workflow	Case 1: same endpoints Case 2: different endpoints The current approach is to request two paths sequentially and impose a "exclude-path" constrain to the second path-request by including a reference to the previously calculated, thus assuring the second path is disjoint from the previous one.

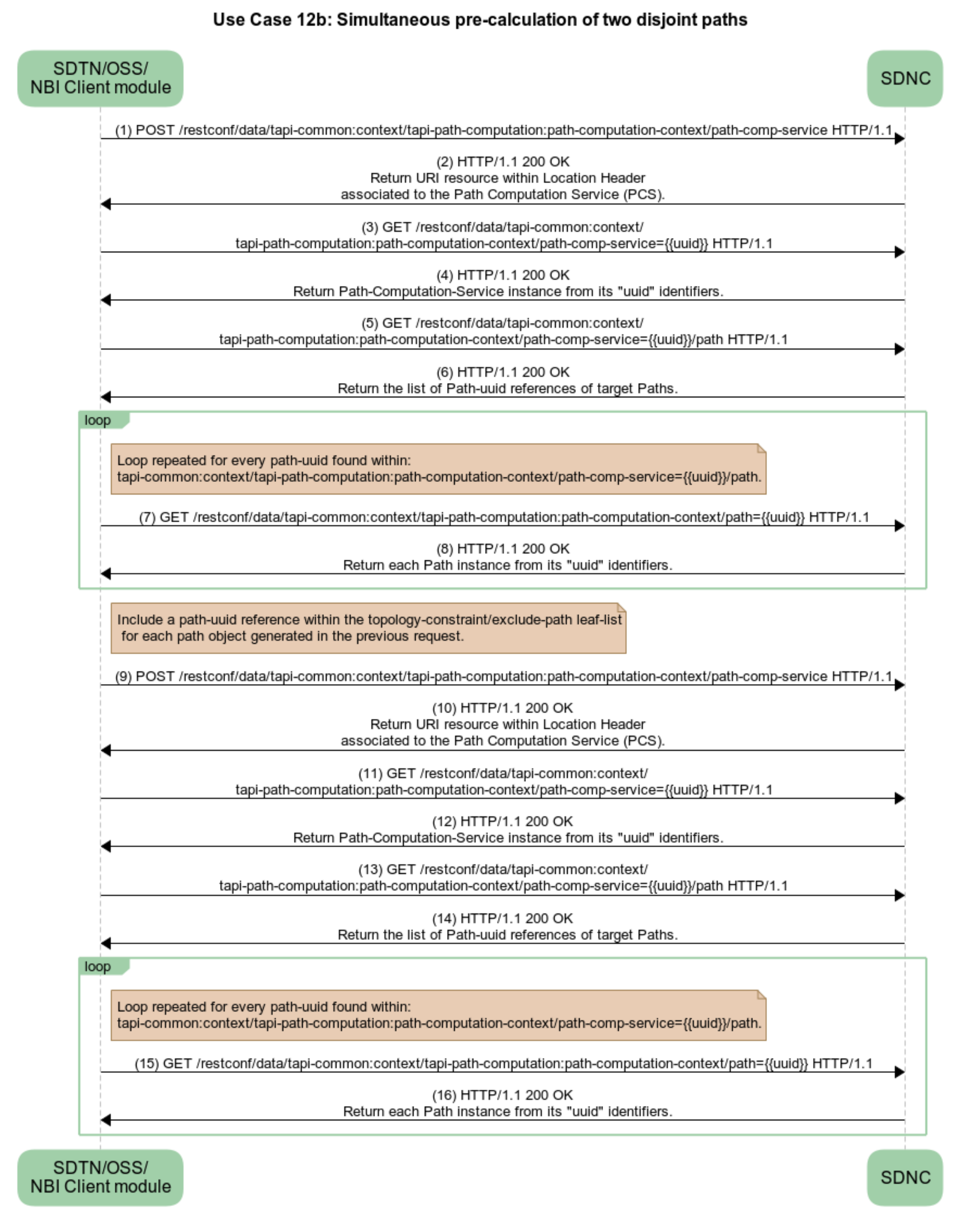


Figure 6-81 UC-12b: Simultaneous pre-calculation of two disjoint paths

6.6.3 Use case 12c: Multiple simultaneous path computation (Bulk request processing)

Number	**UC12c**
Name	**Multiple simultaneous path computation (Bulk request processing)**
Technologies involved	Optical
Process/Areas Involved	Planning and Operations
Brief description	**Disclaimer: This use case is in a draft state, the final definition will be completed in a future release of this reference specification** The multiple simultaneous path computation use case enables the computation of several paths, in such a way that the resulting paths are optimal with regards to the outcome when the path computation is performed in a sequential way. This UC extends 12a to support multiple simultaneous computation. It relies on sending multiple path computation requests one after the other and waiting for the total path computation of all the involved requests.
Layers involved	DSR, DIGITAL_OTN, PHOTONIC_MEDIA
Type	Planning
Description & Workflow	This solution involves the client sending multiple POST messages, as shown in UC12a, but the server MUST not address the path computation until all the POSTs within the logical group of requests have been received. A group is identified as using a convention in the naming scheme. If the client tries to retrieve the path(s) corresponding to the path computation service, before the completion of the logical group of requests, the operation MUST fail.

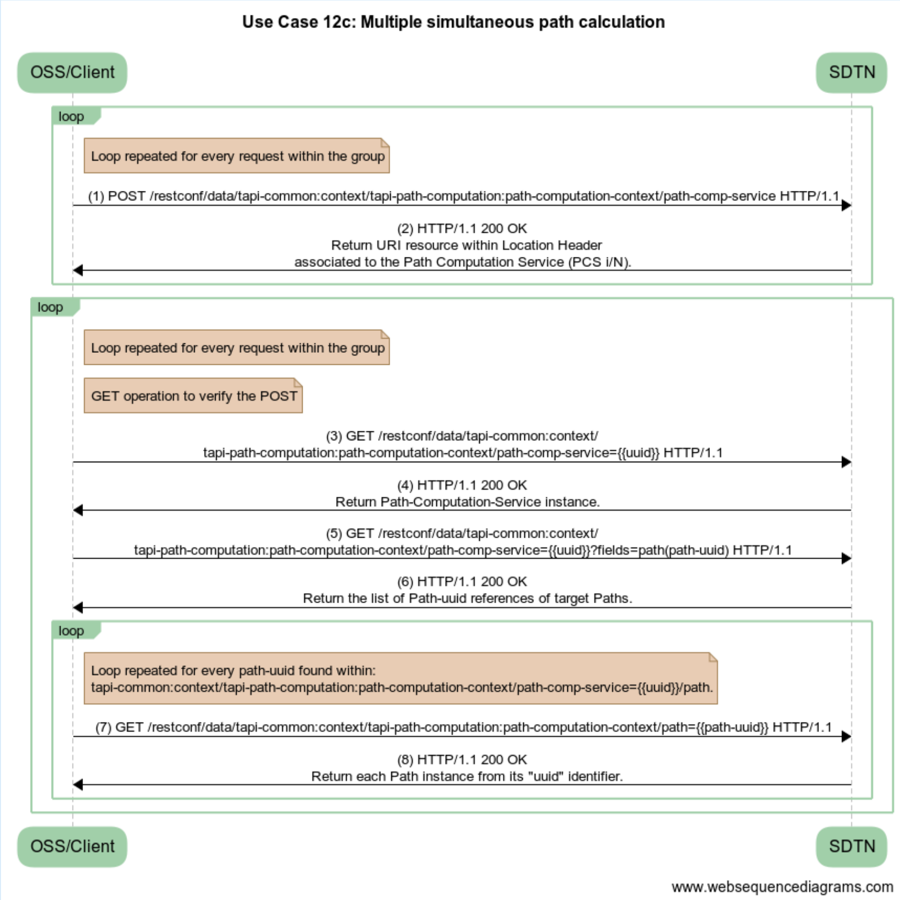


Figure 7 18 UC-12c: Multiple simultaneous path computation

The applicable yang tree is as follows:

```
module: tapi-path-computation
  augment /tapi-common:context:
    +--rw path-computation-context
       +--rw path-comp-service* [uuid]
       |  +--ro path* [path-uuid]
       ...
       |  +--rw end-point* [local-id]
       |  |  +--rw service-interface-point
       |  |  |  +--rw service-interface-point-uuid
...
       |  +--rw routing-constraint
...
       |  +--rw topology-constraint
...
       |  +--rw objective-function
...
       |  +--rw uuid                       uuid
       |  +--rw name* [value-name]
       |     +--rw value-name    string
       |     +--rw value?        string
```


Table 77: Use of value names for bulk processing.

Data Node	/tapi-common:context/tapi-connectivity:connectivity-context/tapi-connectivity:connectivity-service/name			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
value-name/value	"path-request-grup-uuid" and uuid for the group	RW	M	Provided by *tapi-client*.
value-name/value	"request-local-id" encoded as "1/N"	RW	M	Provided by *tapi-client*.

6.6.4 Use case 12d: Physical Impairment Data retrieval for OTSi path planning and validation

Number	UC 12d
Name	**Physical Impairment Data retrieval for OTSi path planning and validation**
Technologies involved	Photonic
Process/Areas Involved	Planning and Operations
Brief description	This UC involves retrieving physical layer impairments data from a TAPI server in order to (potentially) rely on third party tools for path computation and / or validation. This UC includes in particular: - Retrieve the Transceiver profiles. - Retrieve OMS/OTS parameters. - Retrieve ROADM paths profiles. - Retrieve Amplification profiles. - Retrieve Fiber profiles.
Layers involved	PHOTONIC_MEDIA
Type	Notifications and Alarms
Description & Workflow	This UC is an extension of UC0a, UC0b, UC0c since it involves: - Performing GET operation(s) on the list of profiles from the TAPI context - Performing GET operation(s) on NEPs to retrieve applicable profiles - Performing GET operation(s) on CEPs to retrieve applicable profiles

6.6.4.1 Transceiver Impairment data

The Transceiver Impairments are modeled by the *TransceiverProfile* object, which is used to represent:

- The capability of a given Transceiver by means of a list of Transceiver Profile instances.
- The state of a given Transceiver.
- The provisioning of a given Transceiver, as part of the provisioning of the transponder-to-transponder connectivity service (not included in this UC, it is part of e.g., UC2a by using *connectivity-service/end-point/profile* or *connectivity-service/end-point/source-profile* depending on whether the CS is bidirectional or unidirectional).

There are three types of *TransceiverProfile:* the *Standard*, *Organizational* and *Explicit* ones. Figure 6-82 shows that the NEP, which (potentially) supports CEP(s) at OTSiMC layer, may include the list of supported Transceiver Profiles.

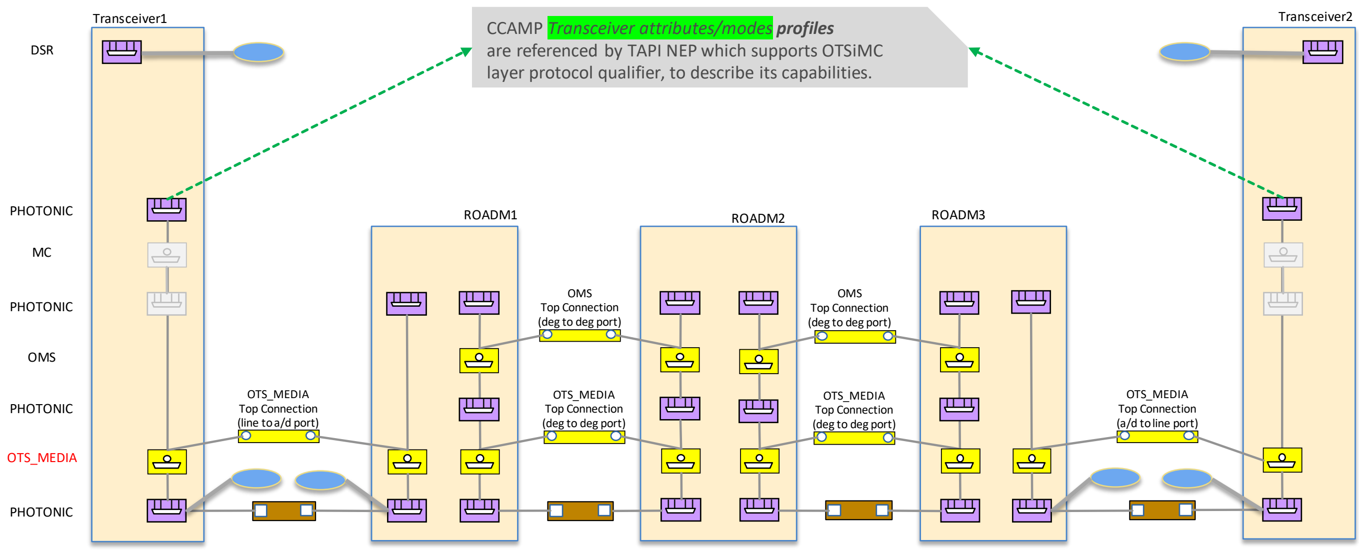


Figure 6-82 Transceiver Profile, capability

6.6.4.1.1 Transceiver Profile retrieval

Once the connectivity service has been provisioned, it is possible to retrieve the Transceiver Profile instance if it is referenced by the OTSiMC CEPs as state information (see Figure 6-83).

GET NEP

GET CEP

6.6.4.1.2 Transceiver Configuration via profile selection

Note: this is not required by the UC, but it is here for completeness.

Additionally, Figure 6-83 shows that a Transceiver Profile instance can be referenced by DSR/ODU/OTU CSEPs at provisioning time of the transponder-to-transponder connectivity service. Note that OTSiA direct provisioning is not considered in this version of this RIA but can be used as Layer Protocol Constraint. Likewise, the direct provisioning of OTSiMC Connectivity Service (CS with layer protocol qualifier being OTSiMC) with transponder-to-transponder is left for further study.

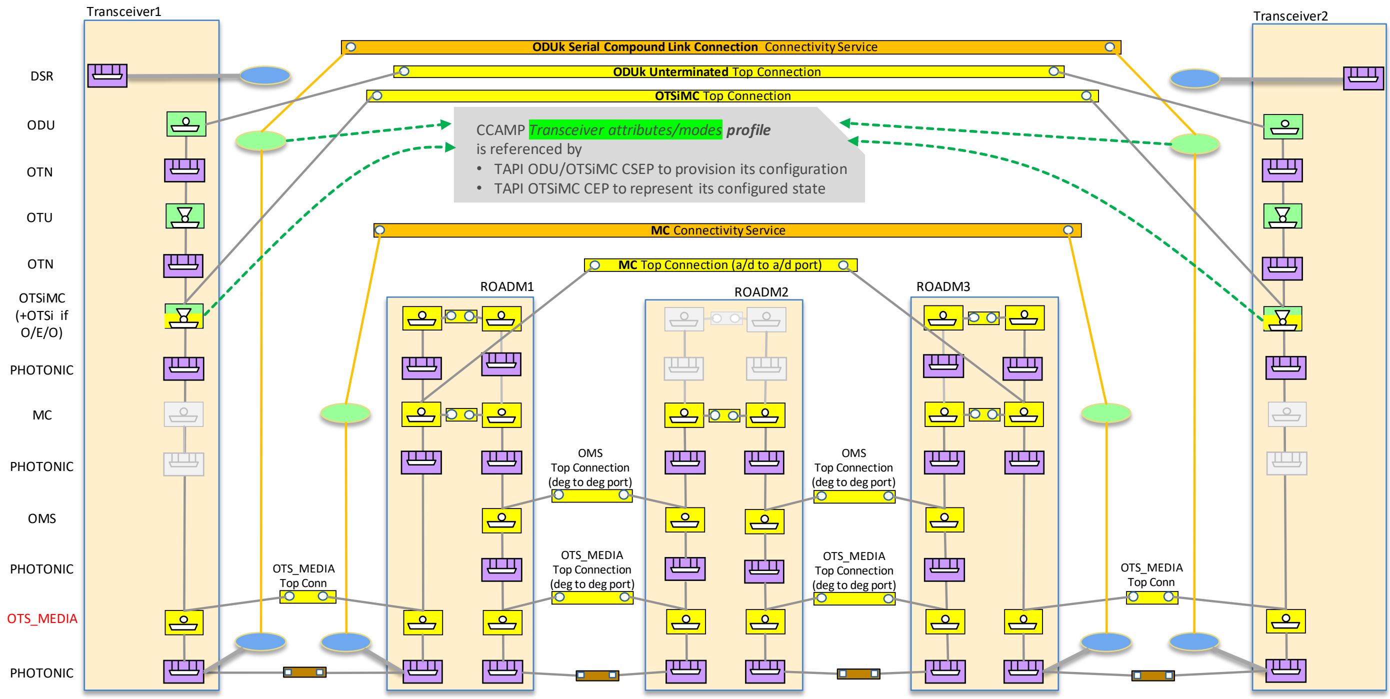


Figure 6-83 Transceiver Profile, configuration and state

6.6.4.2 Optical Multiplex Section Impairments

The OMS Impairments are defined by the *OmsGeneralOpticalParams* object(s), which is (are) included within the OMS CEPs (/tapi-common:context/tapi-topology:topology-context/topology/node/owned-node-edge-point/tapi-connectivity:cep-list/connection-end-point/tapi-photonic-media:oms-connection-end-point-spec/oms-general-optical-params) as shown in Figure 6-84. See Table 43 for details regarding the number of instances and their directionality.

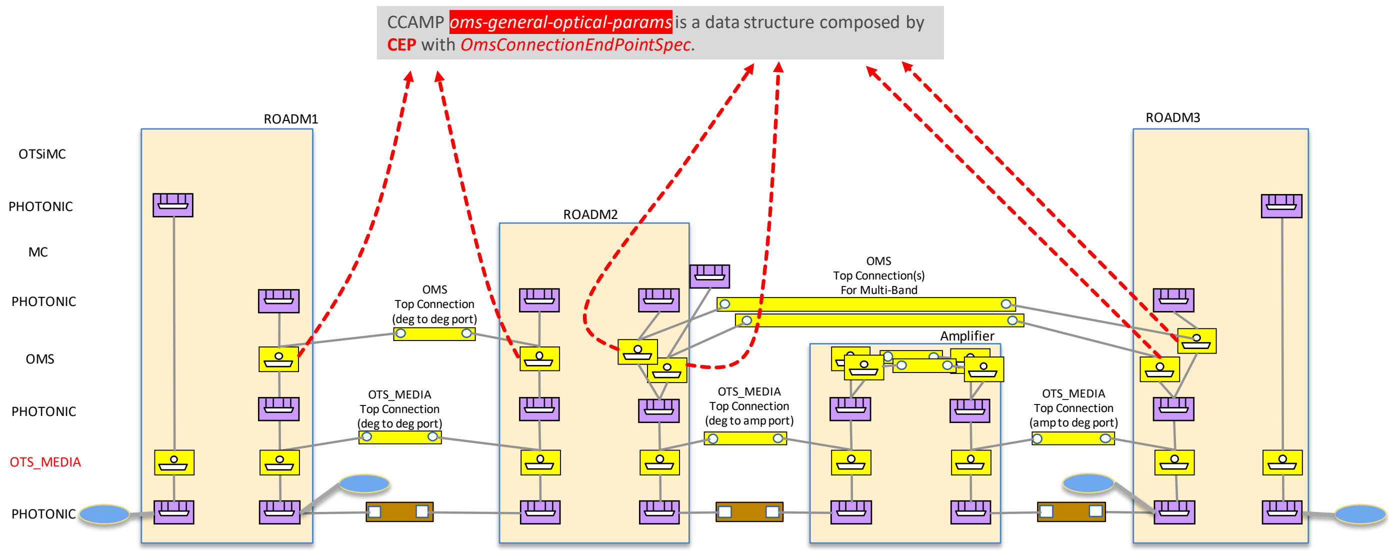


Figure 6-84 OMS Impairments

GET CEP

6.6.4.3 Optical Transmission Section Impairments

The OTS Impairments are defined by the *OtsImpairments* object(s), included within the OTS CEPs (*...tapi-topology:topology/node/owned-node-edge-point/tapi-connectivity:cep-list/connection-end-point/tapi-photonic-media:ots-media-connection-end-point-spec/ots-impairments*) as shown in Figure 6-85.

The *ots-impairments* is a list of max 2 entries (depending on the directionality of the OTS_MEDIA CEP). For a given direction (e.g., ingress-direction false), the object is a sequence of *ImpairmentRouteEntries*, each entry composed of *OtsFiberSpanImpairments* and *OtsConcentratedLoss* entries.

The model includes also the *FiberProfile* object, which could be referenced by *AbstractStrand* object. Further releases of this specification will clarify the relationship between fiber profile and Impairment Route Entries, to be detailed in future version of this document. Please cfr. Table 44 for details.

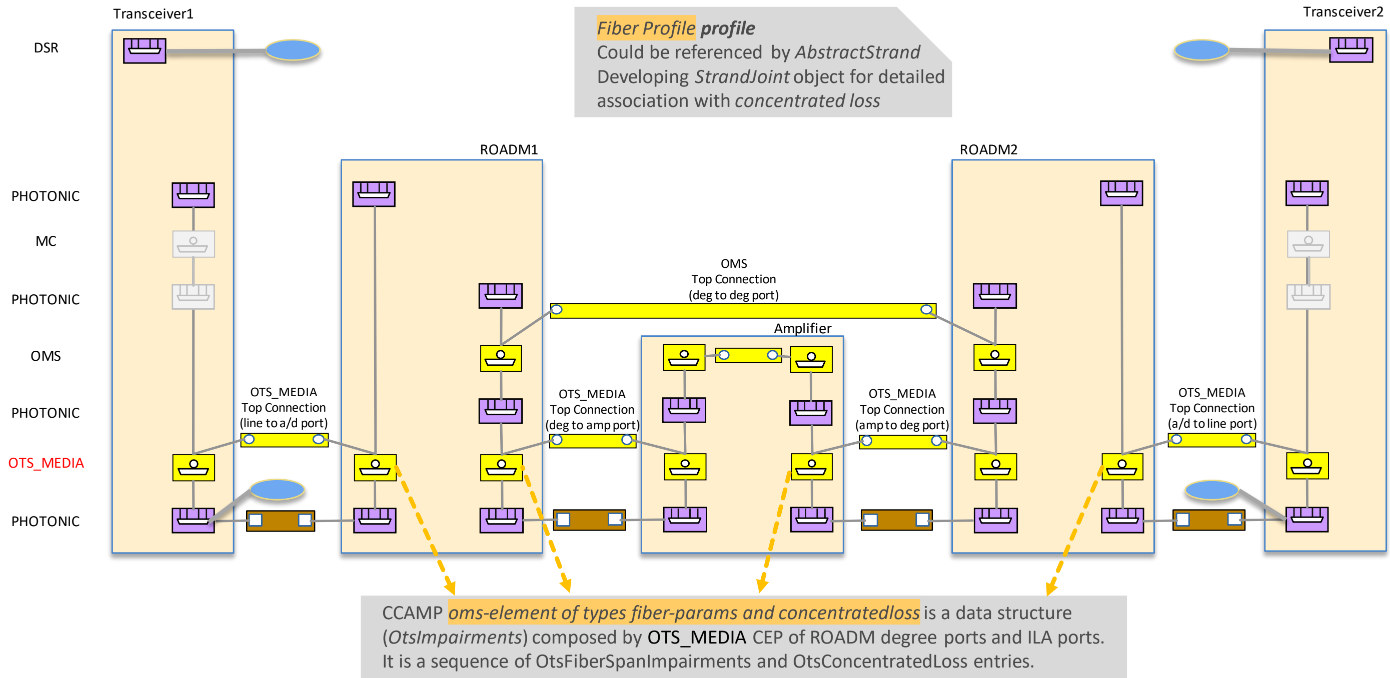


Figure 6-85 OTS Impairments

6.6.4.4 Amplification Impairments

The Amplification Impairments are defined by the *Amplification and AmplificationProfile* objects. As far as TAPI does not foresee a functional oriented model (that is, there are not e.g. "transponder" or "amplification" objects), then the amplification related data are associated to the OMS CEP which better approximates the output of the amplification function. Figure 6-86 shows an example of the amplification objects referenced by the OMS CEPs.

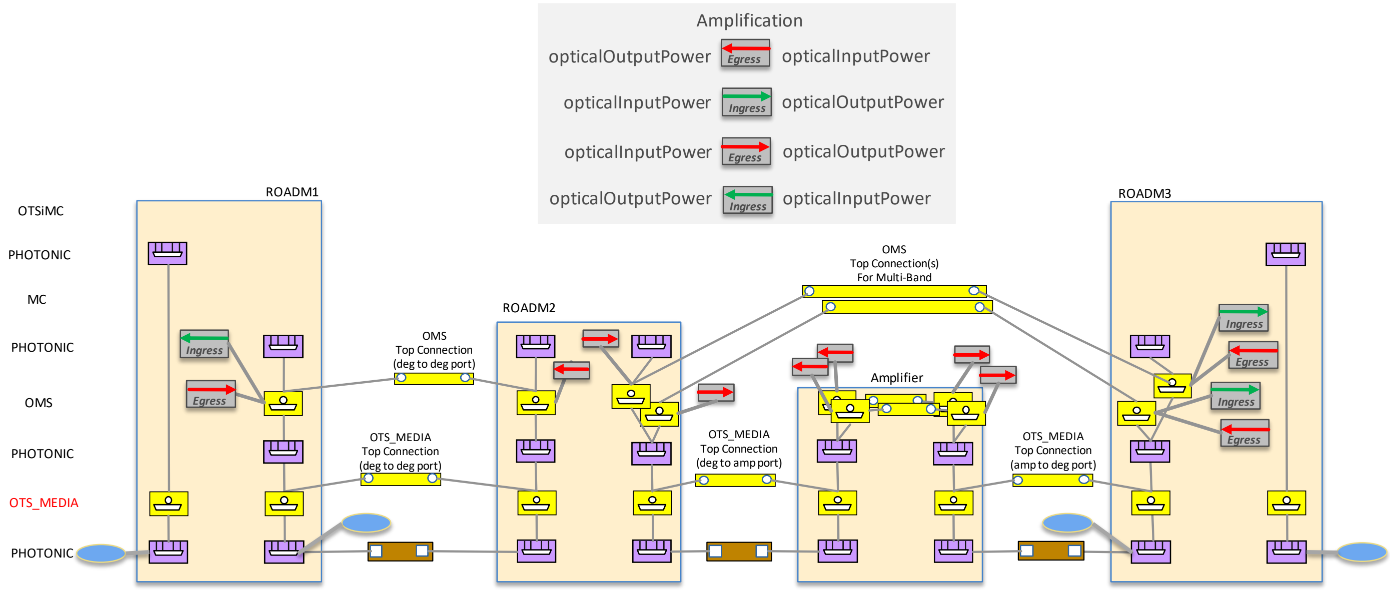


Figure 6-86 Amplification Impairments

More amplification functions can be composed by the same OMS CEP in case of:

1. Separate amplification per spectrum portions and the OMS CEP instance includes all these spectrum portions.
2. The same bidirectional CEP instance better approximates the output of two amplification functions (in the two directions, e.g. booster and preamplifier).
3. The same CEP instance better approximates many amplification stages.

See Figure 6-87.

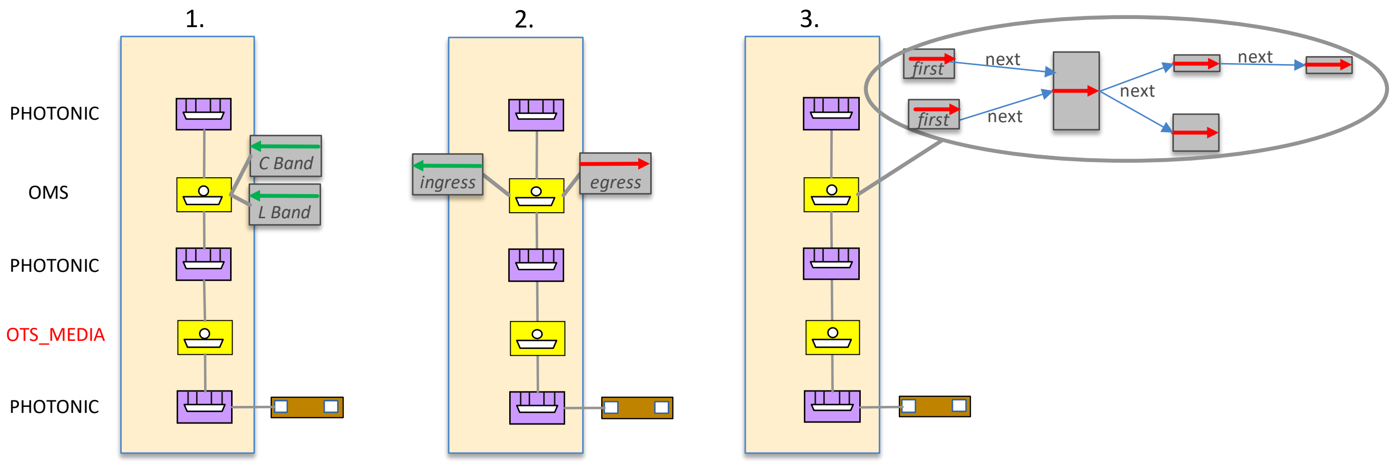


Figure 6-87 More Amplification functions per CEP instance

6.6.4.5 Connectivity Impairments

- A connectivity impairment profile specifies impairments associated to potential connectivity between (the CEPs instantiated on *referenced)* NEPs (A and Z) of a single node. NEPs are grouped into groups using node's node-rule-group. In other words, a group G of NEPs is defined using one **node-rule-group**. Note that, in turn, the NEP node-rule-group list includes all the node-rule-groups the NEP is referred to by.

- The model must support specifying connectivity impairment profiles:

i) between members of a single group (e.g., all degree ports),
ii) between members of different groups (typically two groups e.g., add-drop port group and degree port group).

Moreover, the model should support specifying default connectivity impairment profiles *without explicitly encoding NEP group(s)*. In this case, it is left to the client to deduce the applicability of a given impairment profile. For example, a profile name value (or label) may encode the semantic and involved node edge points. This method is limited in terms of flexibility.

Impairments without defined NEP groups

- The node **profile** list refers to one or more connectivity impairment profiles. Such profiles contain a name value pair with the value-name=”LABEL” and with value a string which is known to the TAPI client. For example, an implementation may add a connectivity impairment profile to the node with name-pair value-name=“LABEL” and value=”add-path”.
- This method is NOT RECOMMENDED and limited to symmetric and simple models.

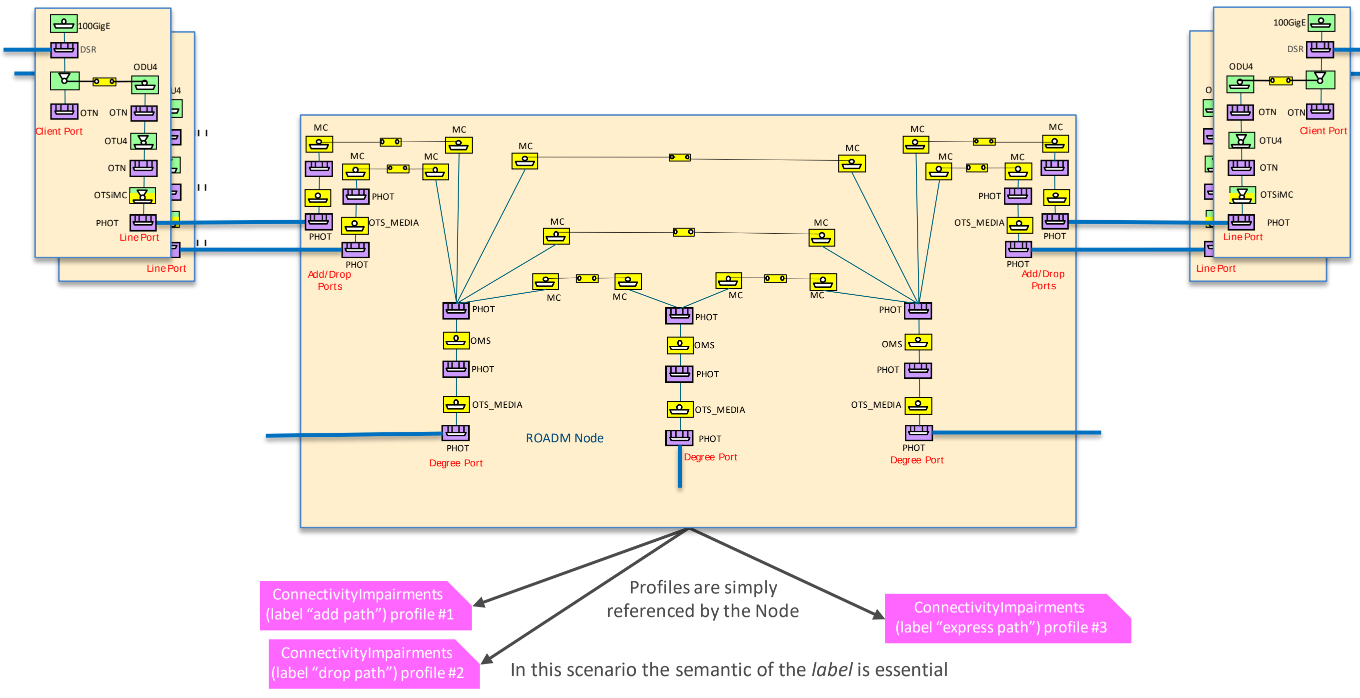


Figure 6-88 Connectivity Impairments – No Node Rule Group

Impairments between NEPs of the same group (e.g., degree ports of a ROADM)

- A group G of NEPs is defined using one **node-rule-group**.
- The node-rule-group’s *node-edge-point* list includes the references to the NEPs in the group.
- This version of the RIA only considers a node-rule-group with one rule of type IMPAIRMENT.
- The node-rule-group’s *rule* has *rule-type* IMPAIRMENT and *cep-direction* MUST be BIDIRECTIONAL or not present (Note that to specify asymmetric A-Z and Z-A profiles the approach with two groups must be used).
- The node-rule-group’s *rule* refers to the applicable connectivity impairment profile (*rule/profile/profile-uuid)*

Impairments between NEPs of different groups (e.g., add-drop ports from/towards degree ports of a ROADM)

- Two or more groups (G1, G2,...) of NEPs are defined using as many **node-rule-groups** needed.
- The *node-edge-point* list of each node-rule-group includes the referenced NEPs.
- The *rule* of each node-rule-group has *rule-type* GROUPING and *cep-direction* may be SINK, SOURCE or BIDIRECTIONAL depending on the applicability/symmetry of the profile.
- The *rule* of each node-rule-group does not refer to any impairment profile.
- One or more node’s *inter-rule-group(s)* associates two groups (e.g., G1 and G2), by using the inter-rule-group *associated-node-rule-group* attribute (e.g., refers to G1 and G2)
- The *rule* of each inter-rule-group has *rule-type* IMPAIRMENT and no *cep-direction*.
- The *rule* of each inter-rule-group refers to the applicable connectivity impairment profile (*rule/profile/profile-uuid)*

Note that a group may also have an impairment rule which applies to two members of such group (that is, *Impairments between NEPs of the same group* and *Impairments between NEPs of different groups* methods can coexist).

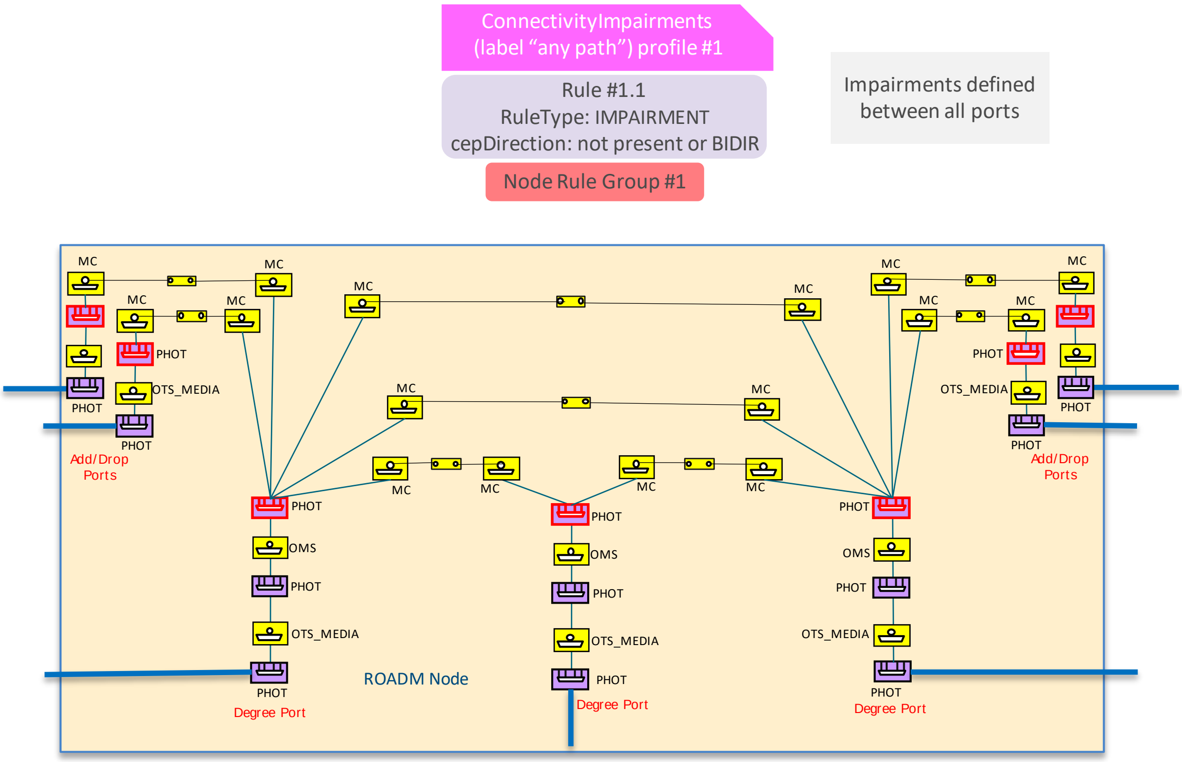


Figure 6-89 Connectivity Impairments are homogeneous for all potential connectivities

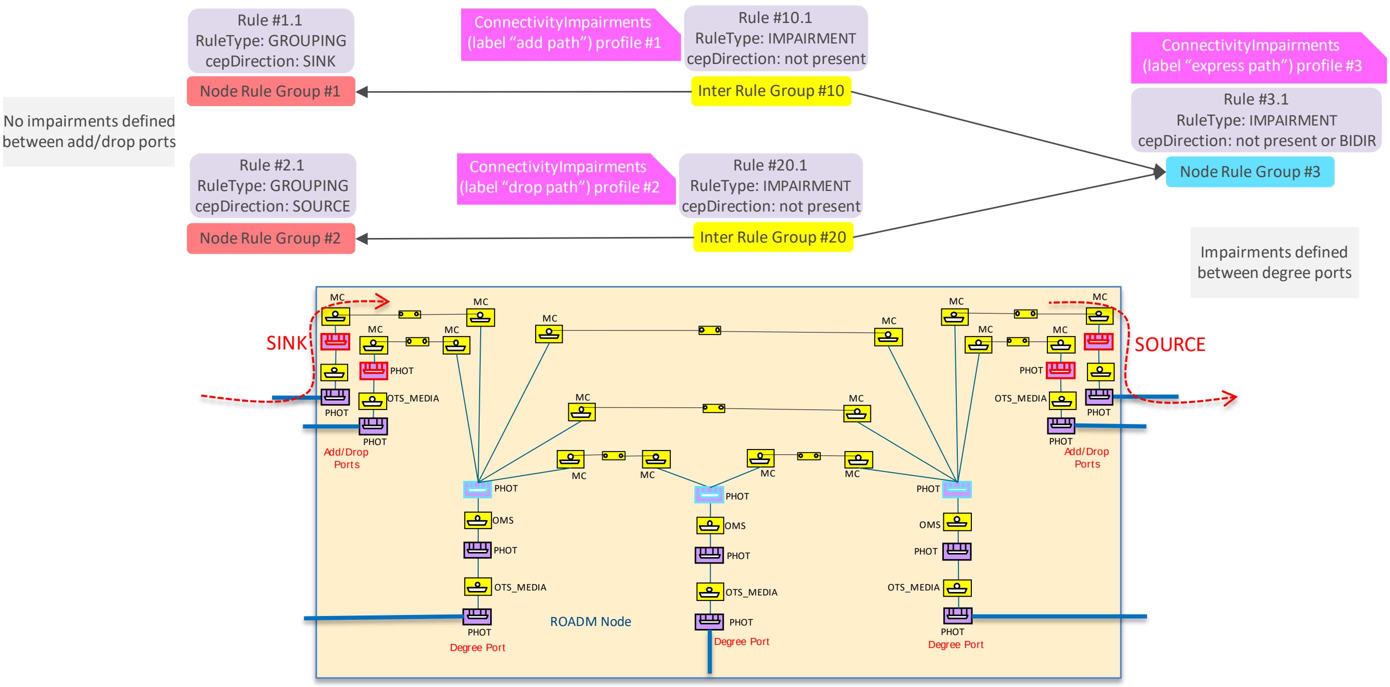


Figure 6-90 Conn. Impairments per *add*, *drop* and *express* conns, homogeneous between *add / drop* and *express*

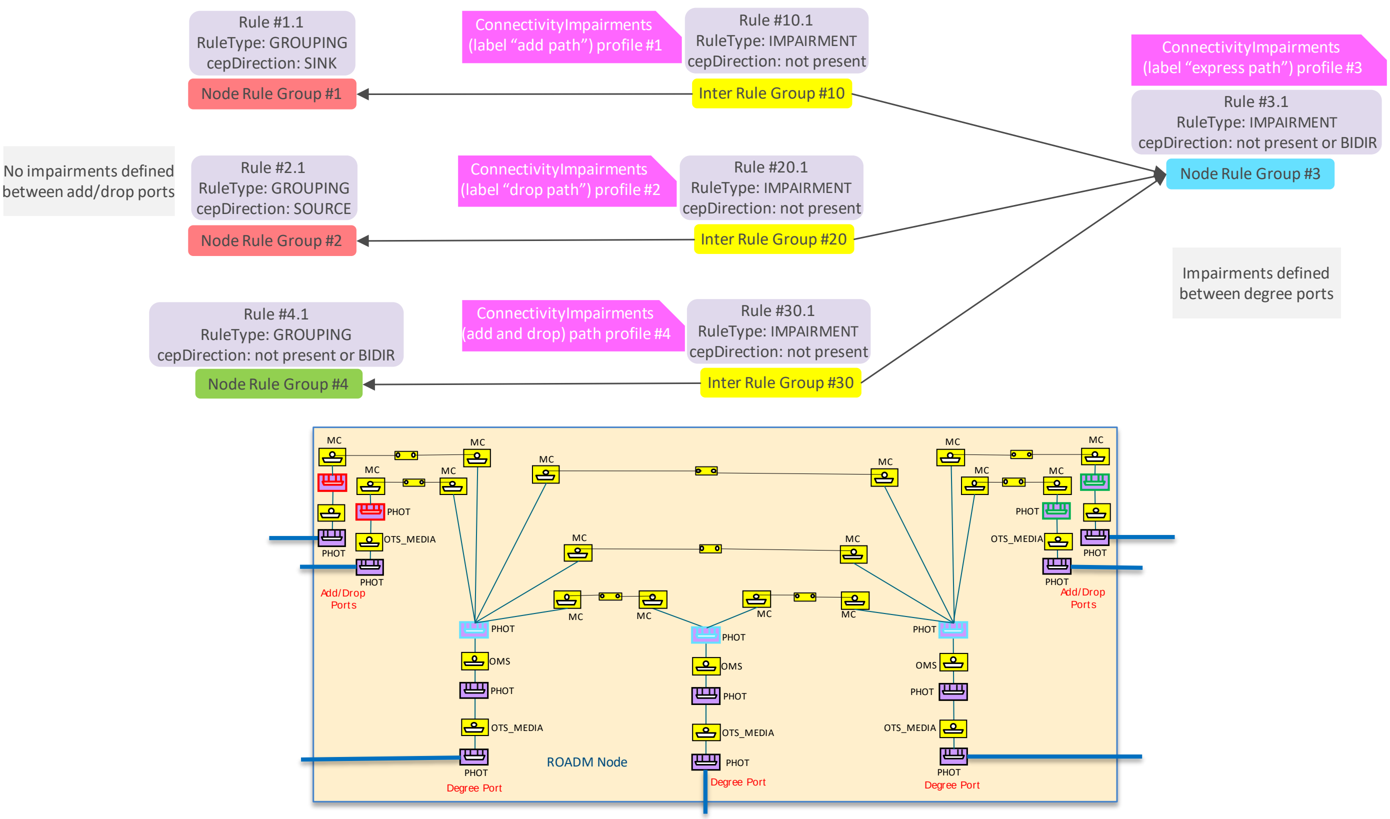


Figure 6-91 Conn. Impairments per *add*, *drop* and *express* conns, not homogeneous between *add / drop* and *express*

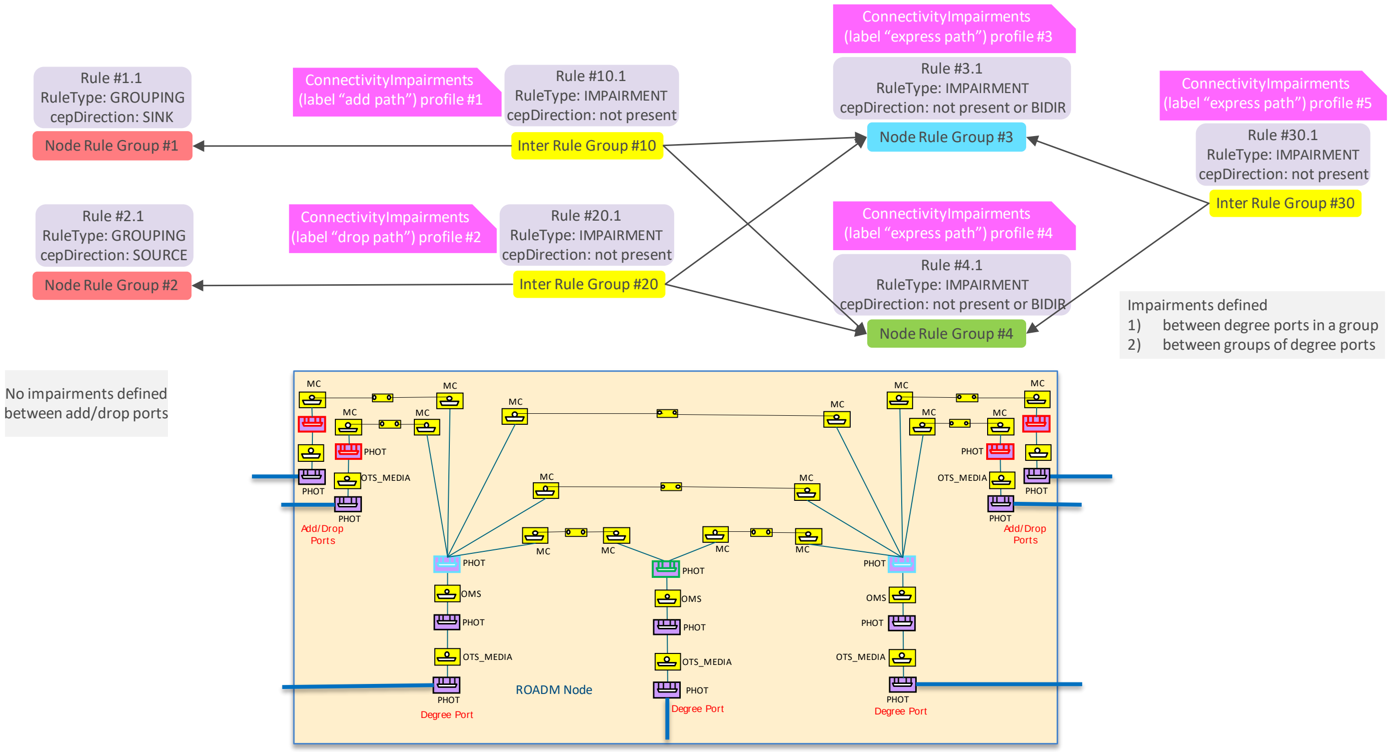


Figure 6-92 Conn. Impairments specified per *add*, *drop* and *express* conns, not homogeneous between *express*

6.7 Notifications and alarms.

As noted in Section 2.7, TAPI Streaming as defined in [ONF TR-548] MAY be used in addition to RESTCONF Notifications. Where TAPI Streaming is used the solution should comply with the Use Cases and structures set out in [ONF TR-548].

6.7.1 Use case 13a: Subscription to Notification service

<table>
<tr><td>Number</td><td>UC 13a</td></tr>
<tr><td>Name</td><td>Subscription to Notification service</td></tr>
<tr><td>Technologies involved</td><td>All</td></tr>
<tr><td>Process/Areas Involved</td><td>Planning and Operations</td></tr>
<tr><td>Brief description</td><td>This UC covers RESTCONF stream subscription, as described in Section 2.7.1.5. This means that the server MUST support a client performing a GET operation to a given RESTCONF stream, once the stream location has been properly discovered, with potentially a filter query parameter. The result of a GET operation to a stream (subscription) creates a subscription channel used for the flow of notifications.

The UC MUST cover the default tapi-notification stream and MAY cover RESTCONF subscription to additional streams. In this version of the RIA, the creation of additional streams is only supported via the creation of TAPI NotificationSubscriptionServices, as specified in Section 2.7.1.4. This creation is limited to the specification of filters as shown in the YANG tree fragment:
<pre>
module: tapi-notification
  augment /tapi-common:context:
    +--rw notification-context
       +--rw notif-subscription* [uuid]
       | +--rw subscription-filter
       | | +--rw requested-notification-types*   notification-type
       | | +--rw requested-object-types*   object-type
       | | +--rw requested-layer-protocols*   tapi-common:layer-protocol-name
       | | +--rw requested-object-identifier*   tapi-common:uuid
       | | +--rw include-content?   boolean
       | | +--rw local-id?   string
       | | +--rw name* [value-name]
       | | +--rw value-name   string
       | | +--rw value?   string
</pre>
Note that the creation of addional streams for filtering MAY be emulated (similar behavior can be achieved) by the proper RESTCONF filter applied to the default tapi-notification stream.

Notification Filtering methods (can be combined):</td></tr>
</table>

<table>
<tr><th>TAPI based (creation of a "filtered stream")</th><th>RESTCONF based (subscription)</th></tr>
<tr><td>Creation of a filtered stream (in addition to the existing default one)</td><td>Creation of a channel (upon subscription)</td></tr>
<tr><td></td><td></td></tr>
<tr><td>notif-subscription/subscription-filter subtree</td><td>Filter query parameters</td></tr>
<tr><td>Filtering parameters:
<code>requested-notification-types,</code>
<code>requested-object-types,</code>
<code>requested-layer-protocols,</code>
<code>requested-object-identifier lists</code></td><td><filter-expression> (which may include, but not limited to):
<pre>
notification
 +--ro notification-type
 +--ro target-object-type
 +--ro target-object-identifier
</pre>
OR
<pre>
event-notification
 +--ro target-object-type
 +--ro target-object-identifier
 +--ro target-local-object-type
 +--ro target-local-object-identifier
</pre></td></tr>
<tr><td>Upon a succesful POST, a new stream appears in the list of RESTCONF streams.

The notification-subscription contains read-only data, whose stream address includes the URI of the new stream:
<pre>
| +--ro notification-channel
      | | +--ro stream-address?   string
      | | +--ro next-sequence-no?   uint64
      | | +--ro local-id?   string
      | | +--ro name* [value-name]
      | | +--ro value-name   string
      | | +--ro value?   string
</pre></td><td>Can be applied to the default tapi-notification stream.</td></tr>
<tr><td>POST method on the tapi-notification:notification-context including the notif-subscription obect.</td><td>GET method on the /stream/<stream-name> where stream name is either "tapi-notification" or a uuid of a TAPI created filtered stream.</td></tr>
<tr><td>Requires RESTCONF subscription</td><td>May not require to interact with TAPI notification context.</td></tr>
</table>

Either by creating a new stream or by applying a RESTCONF filter, the server MUST support the filtering of notifications by a combination of:

- (target-)object-type (i.e., Connectivity-Service, Connection…),
- networking layer,
- Detected condition: */tapi-notification:event-notification/tapi-fm:detected-condition/tapi-fm:detected-condition-name*
- Perceived severity: */tapi-notification:event-notification/tapi-fm:detected-condition/tapi-fm:detector-info/tapi-fm:perceived-severity*

- (event-) notification-types, supporting NOTIFICATION_TYPE_ {
  OBJECT_CREATION,
  ATTRIBUTE_VALUE_CHANGE,
  OBJECT_DELETION,
  FM_ALARM_EVENT,
  FM_THRESHOLD_CROSSING_ALERT
  }

and MAY allow filtering:

- by object-identifier (i.e., uuid)

Implementations MUST support client applications subscribing to the default tapi-notification stream (or additionally created ones) with *different filtering characteristics* thus resulting in different *subscriptions channels*. All NOTIFICATIONs emitted by the TAPI server *through a dedicated subscribtion channel* MUST be tagged with sequence number (monitonically increasing) and a timestamp.

For *notification*

```
/tapi-notification:notification:
         +--ro sequence-number?            uint64
         +--ro event-time-stamp?           tapi-common:date-and-time
```

and for *event-notification*

```
/tapi-notification:event-notification:
         +--ro sequence-number?            uint64
         +--ro event-time-stamp?           tapi-common:date-and-time
```

For the RESTCONF filter, the TAPI server MUST implement the defined filtering mechanism following the [XPath] format. In the following, some possible filters are shown. **Please note the scope of the filtering mechanism is not restricted to the examples proposed.**

Without loss of generality, for the examples please assume all notifications are defined within the custom "*tapi-notification*" stream. For TAPI created additional streams the prefix "/streams/tapi-notification" may vary depending on the stream access/location.

Example 1 filter (both *notification* and *notification-type*):

```
/tapi-notification:notification/notification-
type='NOTIFICATION_TYPE_OBJECT_CREATION'
```

```
GET /streams/tapi-notification?filter=%2Ftapi-
notification%3Anotification%2Fnotification-
type%3D'NOTIFICATION_TYPE_OBJECT_CREATION'
```

```
/tapi-notification:event-notification/event-notification-
type='NOTIFICATION_TYPE_OBJECT_CREATION'
```


<table>
<tr><td></td><td>

```
GET /streams/tapi-notification?filter=%2Ftapi-notification%3Aevent-notification%2Fevent-notification-type%3D'NOTIFICATION_TYPE_OBJECT_CREATION'
```

Example 2 filter:

`(/tapi-notification:notification/notification-type='NOTIFICATION_TYPE_ATTRIBUTE_VALUE_CHANGE' and /tapi-notification:notification/target-object-type='NODE')`

```
GET /streams/tapi-notification?filter=%2Ftapi-notification%3Anotification%2Fnotification-type%3D'NOTIFICATION_TYPE_ATTRIBUTE_VALUE_CHANGE'%20and%20%2Ftarget-object-type%3D'NODE'
```

Example 3 filter:

`/tapi-notification:event-notification/target-object-name['INVENTORY_ID']/value[contains(.,'/ne=MadridNorte')]`

```
GET /streams/tapi-notification?filter=%2Ftapi-notification%3Aevent-notification%2Ftarget-object-name['INVENTORY_ID']%2Fvalue[contains(.%2C'/ne=MadridNorte')]
```

</td></tr>
<tr><td>Layers involved</td><td>DSR, DIGITAL_OTN, PHOTONIC_MEDIA</td></tr>
<tr><td>Type</td><td>Notifications and Alarms</td></tr>
<tr><td>Description & Workflow</td><td>

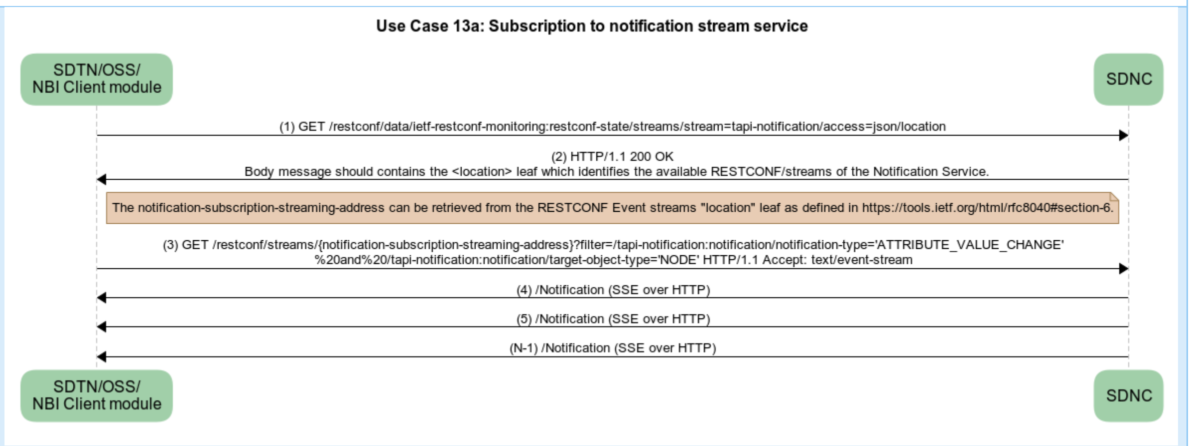


Figure 6-93 UC-13a: Subscription to notification stream service

</td></tr>
</table>

6.7.2 Use case 13b: Subscription to Notification Service for Alarm Events.

Number	**UC 13b**
Name	**Subscription to Notification Service for Alarm Events.**

Technologies involved	All
Process/Areas Involved	Planning and Operations
Brief description	The UC covers the subscription to asynchronous notifications concerning Alarm events. It is based on UC13a where the filtering approaches described MUST support filtering by: - *notification-type* (for notification objects) or *event-notification-type* (for event-notification objects) including NOTIFICATION_TYPE_FM_ALARM_EVENT. - target-object-type (i.e., Connectivity-Service, Connection…), - by networking layer, by target-object-name or by perceived-severity among others. In addition to filtering by common notification fields, implementations MUST allow filtering to select the relevant and add filters based on any mandatory field of the *tapi-fm:alarm-info* as detailed in Section 3.2.8 as well as based on any mandatory field of the *tapi-fm:detected-condition* in which *tapi-fm:detected-condition-name* is any identity based on ALR (alarm). Without loss of generality, for the examples please assume all notifications are defined within the custom "tapi-notification" stream. Examples are provided for *notification*, without excluding the equivalent ones for *event-notification.* Example 1: `Filter =` `/tapi-notification:notification/notification-type='NOTIFICATION_TYPE_FM_ALARM_EVENT'` `GET /streams/tapi-notification?filter=%2Ftapi-notification%3Anotification%2Fnotification-type%3D'NOTIFICATION_TYPE_FM_ALARM_EVENT'` Example 2: `filter = (` `/tapi-notification:notification/notification-type='NOTIFICATION_TYPE_FM_ALARM_EVENT'` `and` `/tapi-notification:notification/target-object-type='EQUIPMENT_OBJECT_TYPE_EQUIPMENT'` `)` `GET /streams/tapi-notification?filter=%2Ftapi-notification%3Anotification%2Fnotification-type%3D'ALARM_EVENT'%20and%20%2Ftarget-object-type%3D'EQUIPMENT_OBJECT_TYPE_EQUIPMENT'` Example 3: `filter = (` `/tapi-notification:notification/notification-type='NOTIFICATION_TYPE_FM_ALARM_EVENT'` `and` `/tapi-notification:notification/tapi-fm:alarm-info/perceived-severity-type='CRITICAL'` `)`

	`GET /streams/tapi-notification?filter=%2Ftapi-notification%3Anotification%2Fnotification-type%3D'NOTIFICATION_TYPE_FM_ALARM_EVENT'%20and%20%2Ftapi-fm%3Aalarm-info%2Fperceived-severity-type%3D'CRITICAL'`
Layers involved	DSR, DIGITAL_OTN, PHOTONIC_MEDIA
Type	Notifications and Alarms
Description & Workflow	This UC is implemented following the same workflow described in "Description & Workflow" of UC13a

6.7.3 Use case 13c: Subscription to Notification Service for Threshold Crossing Alert (TCA).

Number	**UC 13c**
Name	**Subscription to Notification Service for Threshold Crossing Alert (TCA).**
Technologies involved	All
Process/Areas Involved	Planning and Operations
Brief description	The UC covers the subscription to asynchronous notifications concerning TCA events. It is based on UC13a where the filtering approaches described MUST support filtering by: - *notification-type* (for notification objects) or *event-notification-type* (for event-notification objects) including NOTIFICATION_TYPE_FM_THRESHOLD_CROSSING_ALERT. - target-object-type (i.e., Connectivity-Service, Connection…), - by networking layer, by target-object-name or by perceived-severity among others. Additionally, the user may add filters based on any mandatory field of the *tapi-fm:tca-info* as detailed in Section 3.2.8 as well as based on any mandatory field of the *tapi-fm:detected-condition* in which *tapi-fm:detected-condition-name* is any identity based on PM (performance monitoring), including, for example PM_BBE, PM_DELAY or PM_FEC_CORRECTED_ERROR. Without loss of generality, for the examples please assume all notifications are defined within the custom "tapi-notification" stream. Examples are provided for *notification*, without excluding the equivalent ones for *event-notification.* *Note: URL encoding see, for example, UC 13b* Example 1 `filter =` `/tapi-notification:notification/notification-type='NOTIFICATION_TYPE_FM_THRESHOLD_CROSSING_ALERT'` Example 2:

	`filter = (` `/tapi-notification:notification/notification-type='NOTIFICATION_TYPE_FM_THRESHOLD_CROSSING_ALERT'` `and` `/tapi-notification:notification/tapi-fm:tca-info/perceived-tca-severity = 'PERCEIVED_TCA_SEVERITY_CLEAR'` `)`
Layers involved	DSR, DIGITAL_OTN, PHOTONIC_MEDIA
Type	Notifications and Alarms
Description & Workflow	This UC is implemented following the same workflow described in “Description & Workflow” of UC13a

6.7.4 Use case 14a: Subscription and Notification of insertion and removal of Topology Objects

Number	**UC 14a**
Name	**Subscription and Notification of insertion and removal of Topology Objects**
Technologies involved	All
Process/Areas Involved	Planning and Operations
Brief description	The UC covers the emission of events exposing the creation/deletion of Topology object-types such as topology, link, node and node-edge-point (i.e., a TOPOLOGY object when a network element is introduced or removed). This UC includes UC13a where implementations MUST support the subscription including a combination of: - *notification-type* (for *notification* objects) or *event-notification-type* (for *event-notification* objects) including o NOTIFICATION_TYPE_OBJECT_CREATION, o NOTIFICATION_TYPE_OBJECT_DELETION - *target-object-type* including identities based on TOPOLOGY_OBJECT_TYPE o TOPOLOGY_OBJECT_TYPE_TOPOLOGY o TOPOLOGY_OBJECT_TYPE_NODE o TOPOLOGY_OBJECT_TYPE_LINK o TOPOLOGY_OBJECT_TYPE_NODE_EDGE_POINT
Layers involved	DSR, DIGITAL_OTN, PHOTONIC_MEDIA
Type	Notifications and Alarms
Description & Workflow	This UC follows the same workflow as UC13a.

6.7.5 Use case 14b: Subscription and Notification of insertion and removal of Connectivity Objects

Number	**UC 14b**
Name	**Subscription and Notification of insertion and removal of Connectivity Objects**
Technologies involved	All
Process/Areas Involved	Planning and Operations
Brief description	The UC covers the emission of events exposing the creation/deletion of Connectivity Services. This UC includes UC13a where implementations MUST support the subscription including a combination of: - *notification-type* (for *notification* objects) or *event-notification-type* (for *event-notification* objects) including o NOTIFICATION_TYPE_OBJECT_CREATION, o NOTIFICATION_TYPE_OBJECT_DELETION - *target-object-type* including identities based on CONNECTIVITY_OBJECT_TYPE, including: o CONNECTIVITY_OBJECT_TYPE_CONNECTIVITY_SERVICE o CONNECTIVITY _OBJECT_TYPE_CONNECTIVITY_SERVICE_END_POINT o CONNECTIVITY _OBJECT_TYPE_CONNECTION o CONNECTIVITY _OBJECT_TYPE_CONNECTION_END_POINT o CONNECTIVITY _OBJECT_TYPE_CONNECTION_ROUTE o CONNECTIVITY _OBJECT_TYPE_CONNECTION_SWITCH - *target-object-type* being OBJECT_TYPE_SERVICE_INTERFACE_POINT
Layers involved	DSR, DIGITAL_OTN, PHOTONIC_MEDIA
Type	Notifications and Alarms
Description & Workflow	This UC follows the same workflow as UC13a.

6.7.6 Use case 14c: Subscription and Notification of insertion and removal of Path Computation Objects

Number	**UC 14c**
Name	**Subscription and Notification of insertion and removal of Path Computation Objects**
Technologies involved	All
Process/Areas Involved	Planning and Operations

Brief description	The UC covers the emission of events exposing the creation/deletion of Connectivity Services. This UC includes UC13a where implementations MUST support the subscription including a combination of: - *notification-type* (for *notification* objects) or *event-notification-type* (for *event-notification* objects) including o NOTIFICATION_TYPE_OBJECT_CREATION, o NOTIFICATION_TYPE_OBJECT_DELETION - *target-object-type* including identities based on PATH_COMPUTATION_OBJECT_TYPE, including: o PATH_COMPUTATION_OBJECT_TYPE _PATH_COPUTATION_SERVICE o PATH_COMPUTATION_OBJECT_TYPE_PATH_COMP_PATH_SERVICE_END_POINT o PATH_COMPUTATION_OBJECT_TYPE _PATH
Layers involved	DSR, DIGITAL_OTN, PHOTONIC_MEDIA
Type	Notifications and Alarms
Description & Workflow	This UC follows the same workflow as UC13a.

6.7.7 Use case 14d: Subscription and Notification of Creation/Deletion of OAM data

Number	**UC 14d**
Name	**Subscription and Notification of Creation/Deletion of OAM data**
Technologies involved	All
Process/Areas Involved	Notification / OAM
Brief description	The UC covers the emission of events exposing the creation/deletion of Connectivity Services. This UC includes UC13a where implementations MUST support the subscription including a combination of: - *notification-type* (for *notification* objects) or *event-notification-type* (for *event-notification* objects) including o NOTIFICATION_TYPE_OBJECT_CREATION, o NOTIFICATION_TYPE_OBJECT_DELETION - *target-object-type* including identities based on OAM_OBJECT_TYPE, including: o OAM_OBJECT_TYPE_OAM_SERVICE o OAM_OBJECT_TYPE_OAM_SERVICE_POINT o OAM_OBJECT_TYPE_MEG, _MIP, _MEP o OAM_OBJECT_TYPE_OAM_JOB o OAM_OBJECT_TYPE_OAM_PROFILE o OAM_OBJECT_TYPE_CURRENT_DATA o OAM_OBJECT_TYPE_HISTORY_DATA o OAM_OBJECT_TYPE_PM_DATA

Layers involved	DSR, DIGITAL_OTN, PHOTONIC_MEDIA
Type	Notifications and Alarms
Description & Workflow	This UC follows the same workflow as UC13a.

6.7.8 Use case 15a: Notification of status change on existing Topology Objects

Number	**UC 15a**
Name	**Notification of status change on existing Topology Objects**
Technologies involved	All
Process/Areas Involved	Planning and Operations
Brief description	The Notification system MUST emit events exposing the attribute changes of Topology object-types such topology, link, node and node-edge-points. The server MUST report a TOPOLOGY object change notification when a network element is modified due to a network condition or user modification. The server MAY include the reason in the source-indicator of the *notification or event-notification* object. This UC includes UC13a where implementations MUST support the subscription including a combination of: - *notification-type* (for *notification* objects) or *event-notification-type* (for *event-notification* objects) o NOTIFICATION_TYPE_ATTRIBUTE_VALUE_CHANGE - *target-object-type* including identities based on TOPOLOGY_OBJECT_TYPE o TOPOLOGY_OBJECT_TYPE_TOPOLOGY o TOPOLOGY_OBJECT_TYPE_NODE o TOPOLOGY_OBJECT_TYPE_LINK o TOPOLOGY_OBJECT_TYPE_NODE_EDGE_POINT The server MUST include the changed-attributes parameter in the notification.
Layers involved	DSR, DIGITAL_OTN, PHOTONIC_MEDIA
Type	Notifications and Alarms
Description & Workflow	This UC follows the same workflow as UC13a.

6.7.9 Use case 15b: Notification of status change on existing Connectivity Objects

Number	**UC 15b**

Name	**Notification of status change on existing Connectivity Objects**
Technologies involved	All
Process/Areas Involved	Planning and Operations
Brief description	The Notification system MUST emit events exposing the attribute changes of Connectivity object-types such connectivity-services, connections and connection-end-points and service-interface-points. The server MUST report a connectivity object change notification when such object is modified due to a network condition or user modification. The server MAY include the reason in the source-indicator. This UC includes UC13a where implementations MUST support the subscription including a combination of: - *notification-type* (for *notification* objects) or *event-notification-type* (for *event-notification* objects) including ○ NOTIFICATION_TYPE_ATTRIBUTE_VALUE_CHANGE - *target-object-type* including identities based on CONNECTIVITY_OBJECT_TYPE, including: ○ CONNECTIVITY_OBJECT_TYPE_CONNECTIVITY_SERVICE ○ CONNECTIVITY _OBJECT_TYPE_CONNECTIVITY_SERVICE_END_POINT ○ CONNECTIVITY _OBJECT_TYPE_CONNECTION ○ CONNECTIVITY _OBJECT_TYPE_CONNECTION_END_POINT - *target-object-type* being OBJECT_TYPE_SERVICE_INTERFACE_POINT The server MUST include the changed-attributes parameter in the notification.
Layers involved	DSR, DIGITAL_OTN, PHOTONIC_MEDIA
Type	Notifications and Alarms
Description & Workflow	This UC follows the same workflow as UC13a.

6.7.10 Use case 15c: Notification of status change on the switching conditions of an existing Connection

Number	**UC 15c**
Name	**Notification of status change on the switching conditions of an existing Connection.**
Technologies involved	All
Process/Areas Involved	Planning and Operations
Brief description	The Notification system MUST emit events exposing the attribute changes of Connection sub-object-types such ROUTE and SWITCH.

	The server MUST report a connectivity object change notification when such object is modified due to a network condition or user modification. The server MAY include the reason in the source-indicator. This UC includes UC13a where implementations MUST support the subscription including a combination of: - *notification-type* (for *notification* objects) or *event-notification-type* (for *event-notification* objects) including o NOTIFICATION_TYPE_ATTRIBUTE_VALUE_CHANGE - *target-object-type* including identities based on CONNECTIVITY_OBJECT_TYPE, including: o CONNECTIVITY _OBJECT_TYPE_ROUTE o CONNECTIVITY _OBJECT_TYPE_SWITCH The server MUST include the changed-attributes parameter in the notification
Layers involved	DSR, DIGITAL_OTN, PHOTONIC_MEDIA
Type	Notifications and Alarms
Description & Workflow	This UC follows the same workflow as UC13a.

6.7.11 Use case 15d: Notification of status change on the OAM data

Number	**UC 15d**
Name	**Notification of status change on OAM data**
Technologies involved	All
Process/Areas Involved	Planning and Operations
Brief description	The UC covers the emission of events exposing the creation/deletion of Connectivity Services. This UC includes UC13a where implementations MUST support the subscription including a combination of: - *notification-type* (for *notification* objects) or *event-notification-type* (for *event-notification* objects) including NOTIFICATION_TYPE_ATTRIBUTE_VALUE_CHANGE o - *target-object-type* including identities based on OAM_OBJECT_TYPE, including: o OAM_OBJECT_TYPE_OAM_SERVICE o OAM_OBJECT_TYPE_OAM_SERVICE_POINT o OAM_OBJECT_TYPE_MEG, _MIP, _MEP o OAM_OBJECT_TYPE_OAM_JOB o OAM_OBJECT_TYPE_OAM_PROFILE o OAM_OBJECT_TYPE_CURRENT_DATA o OAM_OBJECT_TYPE_HISTORY_DATA (to be discussed) o OAM_OBJECT_TYPE_PM_DATA

Layers involved	DSR, DIGITAL_OTN, PHOTONIC_MEDIA
Type	Notifications and Alarms – OAM
Description & Workflow	This UC follows the same workflow as UC13a.

6.7.12 Use case 16a: Notification of Alarm events

Number	**UC16a**
Name	**Notification of Alarm events**
Technologies involved	All
Process/Areas Involved	Planning and Operations
Brief description	The Notification system MUST emit events related to alarms. This UC includes the subscription in UC13b. This UC involves the parameters included in either **tapi-fm:alarm-info** (deprecated) or in detected-condition with **tapi-fm:detected-condition-name** is any identity based on ALR (alarm).
Layers involved	DSR, DIGITAL_OTN, PHOTONIC_MEDIA
Type	Planning and Operations
Description & Workflow	This Use Case relies on in the workflow defined in UC13b.

6.7.12.1 Relevant parameters

Table 78: UC16a Alarm information (tapi-fm:alarm-info) Relevant Parameters

Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Information Recorded**
See Table 5: Alarm information (alarm-info) Relevant Parameters				

Table 79: UC16a Alarm information (detected condition) Relevant Parameters

Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Information Recorded**

See Table 7: detected-condition object definition

6.7.13 Use case 16b: Notification of Threshold Crossing Alert (TCA) events

Number	**UC16b**
Name	**Notification of Threshold Crossing Alert (TCA) events**
Technologies involved	All
Process/Areas Involved	Planning and Operations
Brief description	The Notification system MUST emit events related to alarms. This UC includes the subscription in UC13c This UC involves the parameters included in either tapi-fm:alarm-info (deprecated) or in detected-condition with tapi-fm:detected-condition-name is any identity based on PM (performance monitoring)
Layers involved	DSR, DIGITAL_OTN, PHOTONIC_MEDIA
Type	Planning and Operations
Description & Workflow	This Use Case relies on the workflow defined in UC13c.

6.7.13.1 Relevant parameters

Table 80: UC16b TCA information (tapi-fm:tca-info) Relevant Parameters

Attribute	Allowed Values/Format	Mod	Sup	Information Recorded
See Table 6: Threshold Crossing Alert information (tapi-fm:tca-info) Relevant Parameters				

Table 81: UC16b TCA information (detected condition) Relevant Parameters

Attribute	Allowed Values/Format	Mod	Sup	Information Recorded
See Table 7: detected-condition object definition				

6.8 Performance and OAM.

TAPI OAM enables to perform SLA compliance of a TAPI Connectivity Service (CS). TAPI OAM provides the representation of Generation/Termination, Processing and Forwarding of OAM overhead constructs for the purpose of Fault Detection, Fault Propagation and Performance Monitoring.

TAPI OAM enables the retrieval of performance counter values and enables the configuration, start, and stop functions related to Detect & Monitoring, Performance collection and Maintenance Tests. The discovery of this information is intended to be requested periodically and/or on-demand basis, proactively from the TAPI client role, to synchronize the OAM context information.

6.8.1 OAM Provisioning Scenarios

Two provisioning scenarios are considered for OAM Services: a lightweight "embedded" approach where the OAM properties are specified as part of the CS provisioning and an "independent" approach which provides additional features/flexibility, and which involves the explicit creation of an OAM Service and related OAM Service Point(s).

In the embedded case,

- No MEGs (MIP/MEPs) are instantiated, since the parameters are included in the CEP instances (for example, NCM in ODU connections).
- OAM-related information is present in the connectivity-context and, more precisely in:
- i) the CSEPs (*tapi-oam:connectivity-oam-job and tapi-oam:connectivity-oam-service. Within connectivity-oam-service*
    - *connectivity-oam-service-point list*
    - *otn-oam-mep-service-point with odu-mep, odu-tcm-mep and otu-mep*
    - *otn-oam-mip-service-point with odu-mip and odu-tcm-mip*

- ii) in CEPs, such as ODU/OTU OAM including
    - *tapi-digital-otn:odu-connection-end-point-spec/odu-term-and-adapter/odu-mep*
    - *tapi-digital-otn:otu-connection-end-point-spec/otu-ttp-pac/otu-mep*.

In the independent case,

- A MEG, and its MEPs/MIPs MAY be instantiated upon creation of an OAM Service. For example, the system may reuse OAM information already present in CEPs instead.
- If a MEG is instantiated, the involved CEPs MUST have a reference to the supported MEPs and MIPs (via their tapi-oam:mep-mip-list CEP augmentation)
    - Example of MEG instantiation: ODU Tandem Connection Monitoring (TCM).
    - Example of no MEG instantiation: an OAM Service created to monitor optical power or a loopback service directly on photonic media CEPs, since the OAM parameters are included in the CEP instances.

In all cases,

- CEPs MAY also have active monitoring points that have not been provisioned by the client. In other words, additional PM parameters MAY be part of the CEP object without explicit configuration (e.g., ODU NIM modelled through tapi-digital-otn:odu-connection-end-point-spec/odu-ctp/odu-mip or tapi-photonic-media:otsi-mc-connection-end-point-spec with the measured optical power within the power-measurement-pac).

- OAM Jobs (*tapi-oam:oam-context:oam-job)* MAY be created which holds PM data. In such case, they MAY exist even the corresponding connectivity service and connections have been deleted. In other words, measurements may be available after the connectivity deletion, with OAM Job in "concluded" state. Implementations SHOULD document this behaviour along with rules that apply to job deletion (e.g., client deletion, policy/time based, etc.)

In this sense, regarding the objects lifecycle and PM data retrieval, the following rules apply:

In the independent case,

- OAM Service Points, MEPs and MIPs cannot exist without the CS/CSEPs and related Connection(s)/CEPs.
- OAM Jobs do not refer to any CSEP (In the embedded case the jobs point to the CSEP).

Figure 6-94 shows the main OAM scenarios considered in this RIA. NCM stands for Network Connection Monitoring, TCM for Tandem Connection Monitoring.

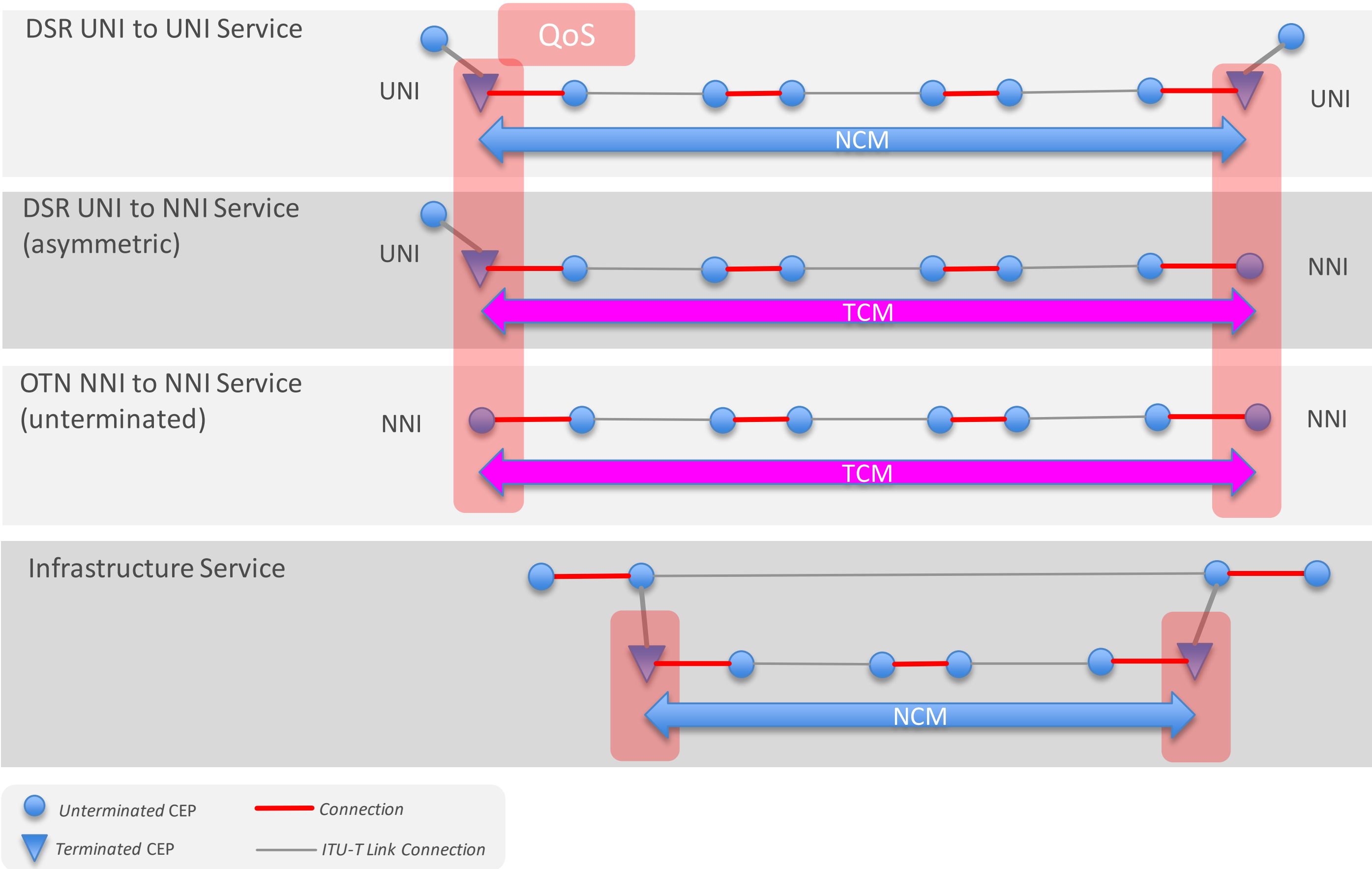


Figure 6-94 OAM Scenarios

Figure 6-95 and Figure 6-96 show the configuration steps in case of *embedded* mode, *DSR UNI to UNI Service scenario* and *Infrastructure Service scenario*, monitoring functions are two MEPs.

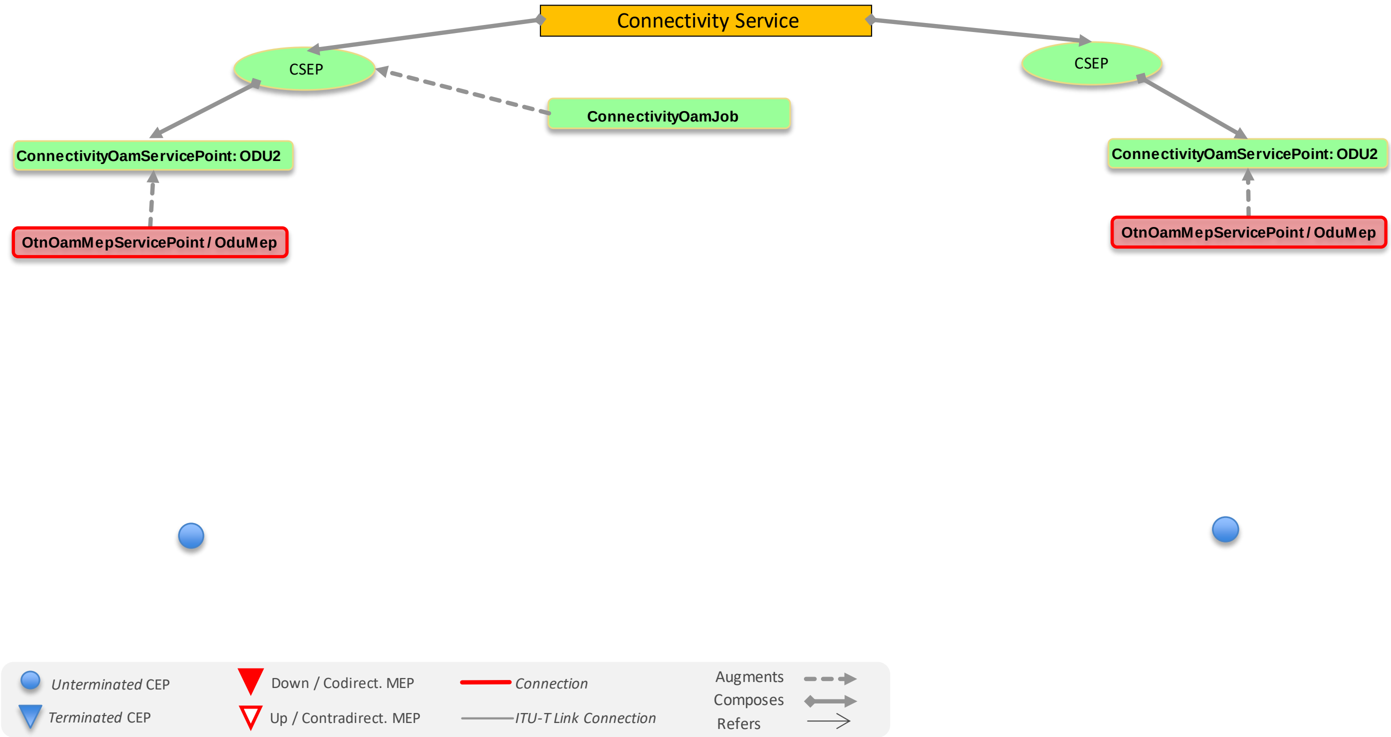


Figure 6-95 OAM provisioning, Client Controller creates the CS with the CSEPs including OAM configuration

Figure 6-96 shows:

- The creation or activation of MEP parameters of involved CEPs, according to the ConnectivityOamServicePoint augments of the CSEPs. Note that the NCM MEPs are composed by the CEPs, there is not a distinct MEP object instance.
- The creation of OAM Job instance and related Current and History Data instances according to the ConnectivityOamJob augment of the CSEP.

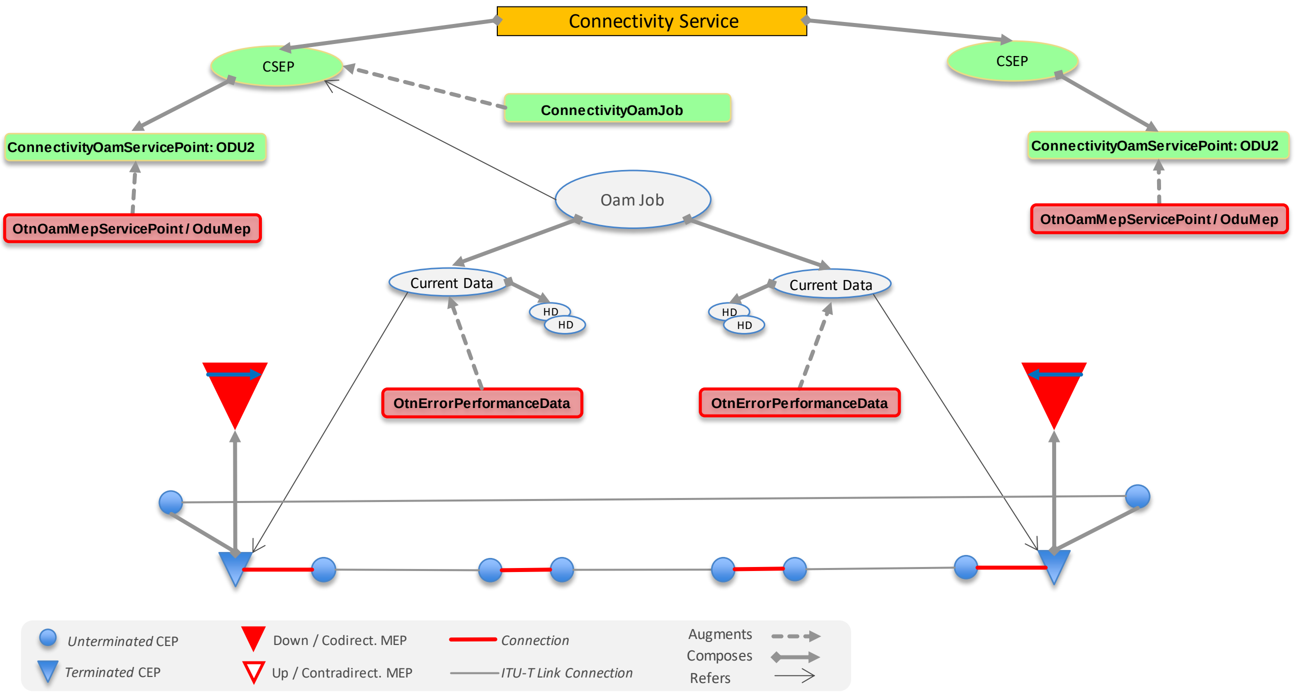


Figure 6-96 OAM provisioning, Server Controller creates OAM Job, Current and History Data instances

Figure 6-97 shows the *DSR UNI to NNI (asymmetric)* scenario, provisioned through *embedded* mode. Monitoring functions are one MEP and two MIPs.

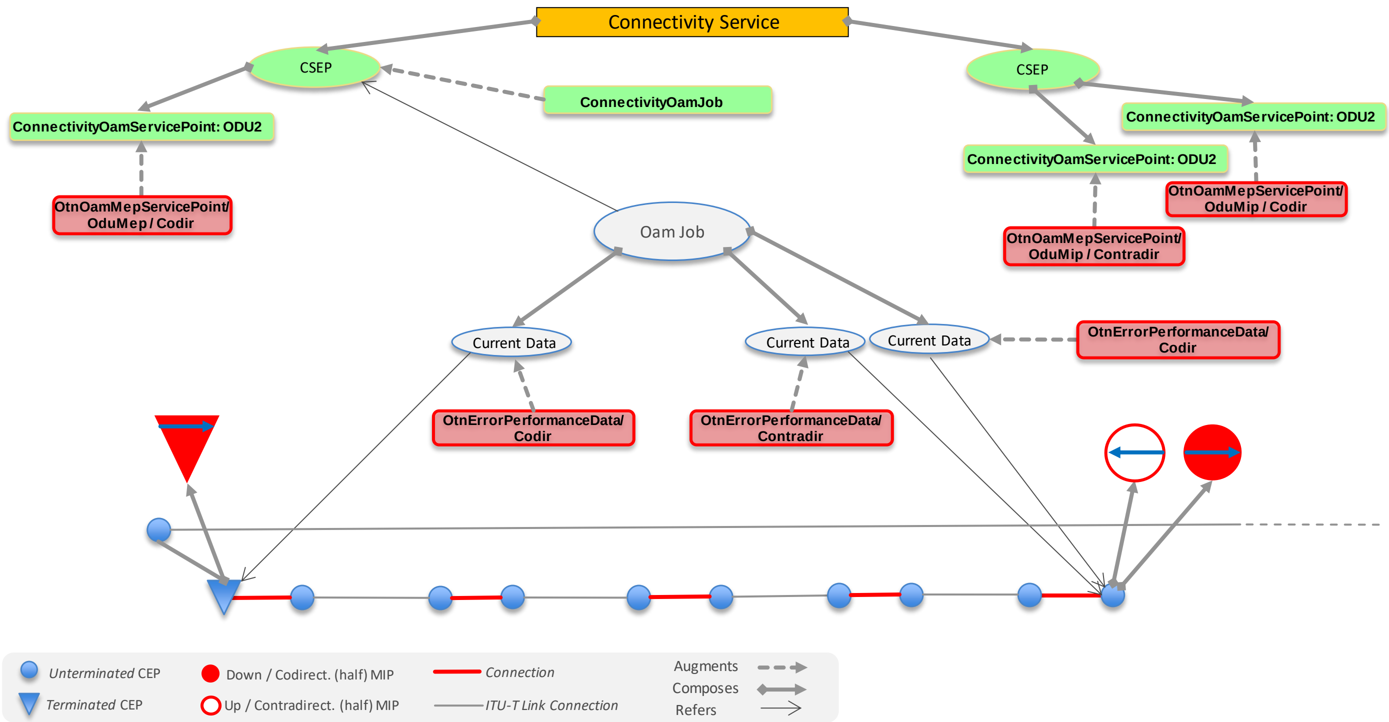


Figure 6-97 OAM provisioning, DSR UNI to NNI (asymmetric)

Figure 6-98 shows the *OTN NNI to NNI (unterminated) scenario*, provisioned through *embedded* mode. Monitoring functions are four MIPs.

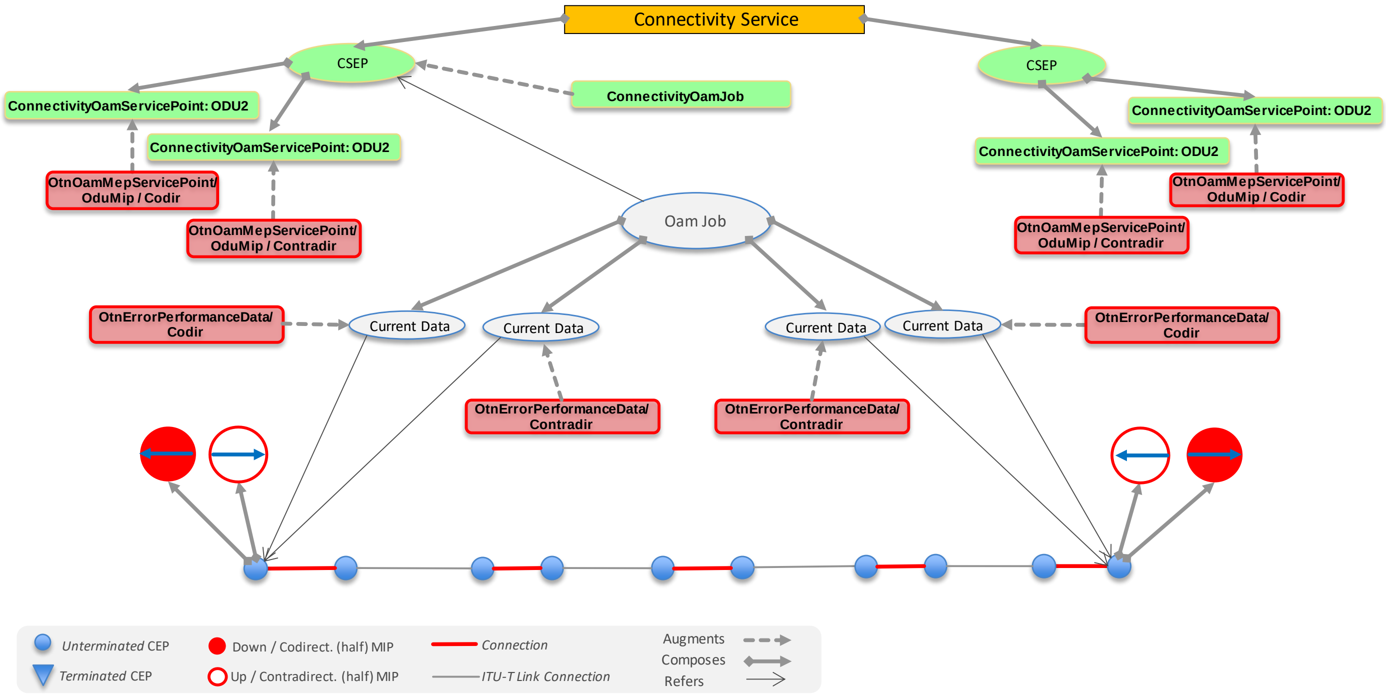


Figure 6-98 OAM provisioning, OTN NNI to NNI (unterminated)

Figure 6-99, Figure 6-100, Figure 6-101, Figure 6-102 show the configuration steps in case of *independent* mode, *OTN NNI to NNI (unterminated) scenario*, monitoring functions are four TCM MEPs. Note that in the Figures, the Connectivity Service has previously been provisioned (pre-existing in the independent mode).

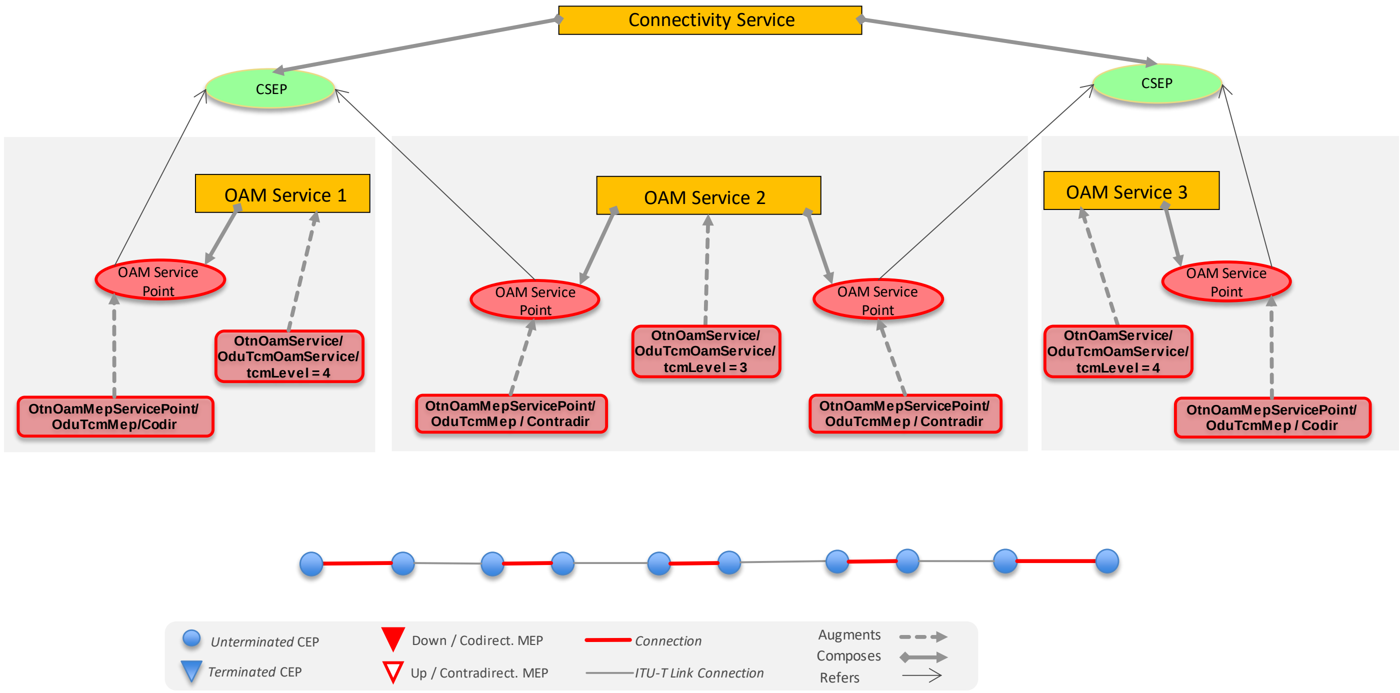


Figure 6-99 OAM provisioning, Client Controller creates the OAM Service and its End Points, OTN NNI to NNI

Figure 6-100 shows the creation, by the server, of TCM MEG and MEP instances according to OAM Service Point provisioning.

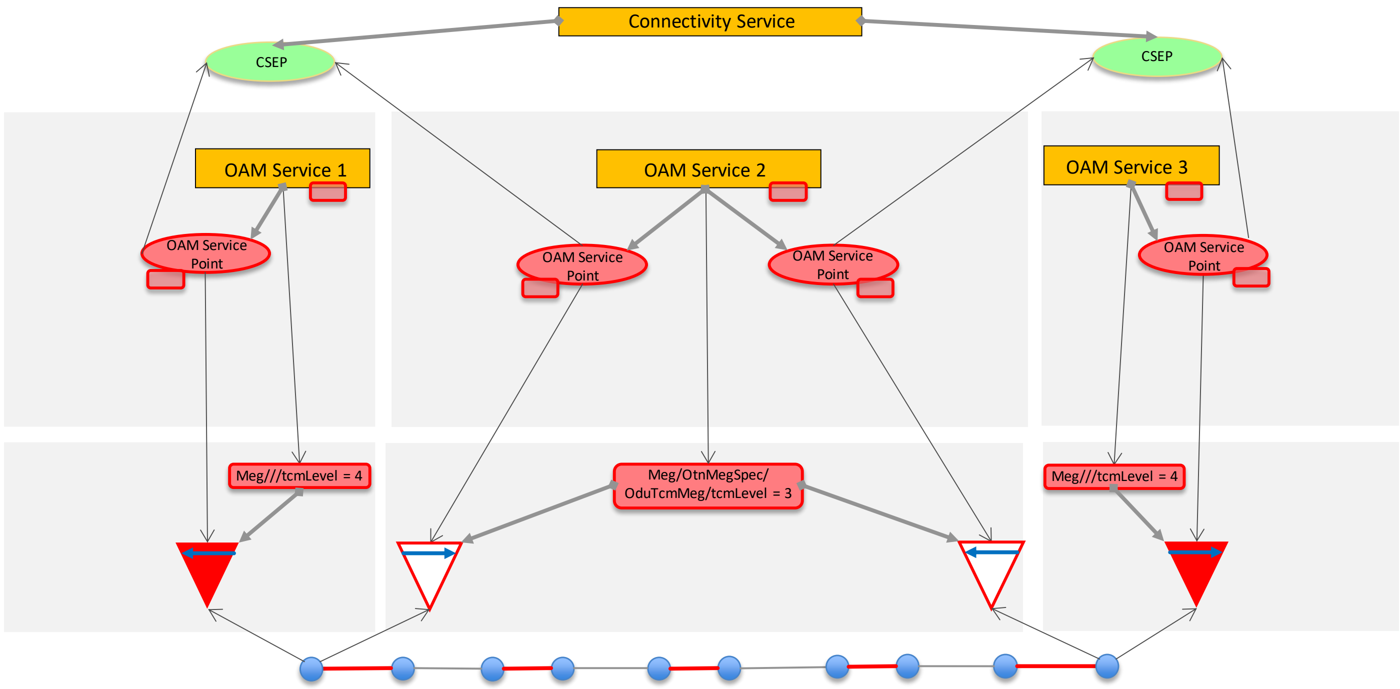


Figure 6-100 OAM provisioning, Server Controller creates the TCM MEG and MEP instances

Figure 6-101 shows the provisioning of OAM Job instances by the client controller.

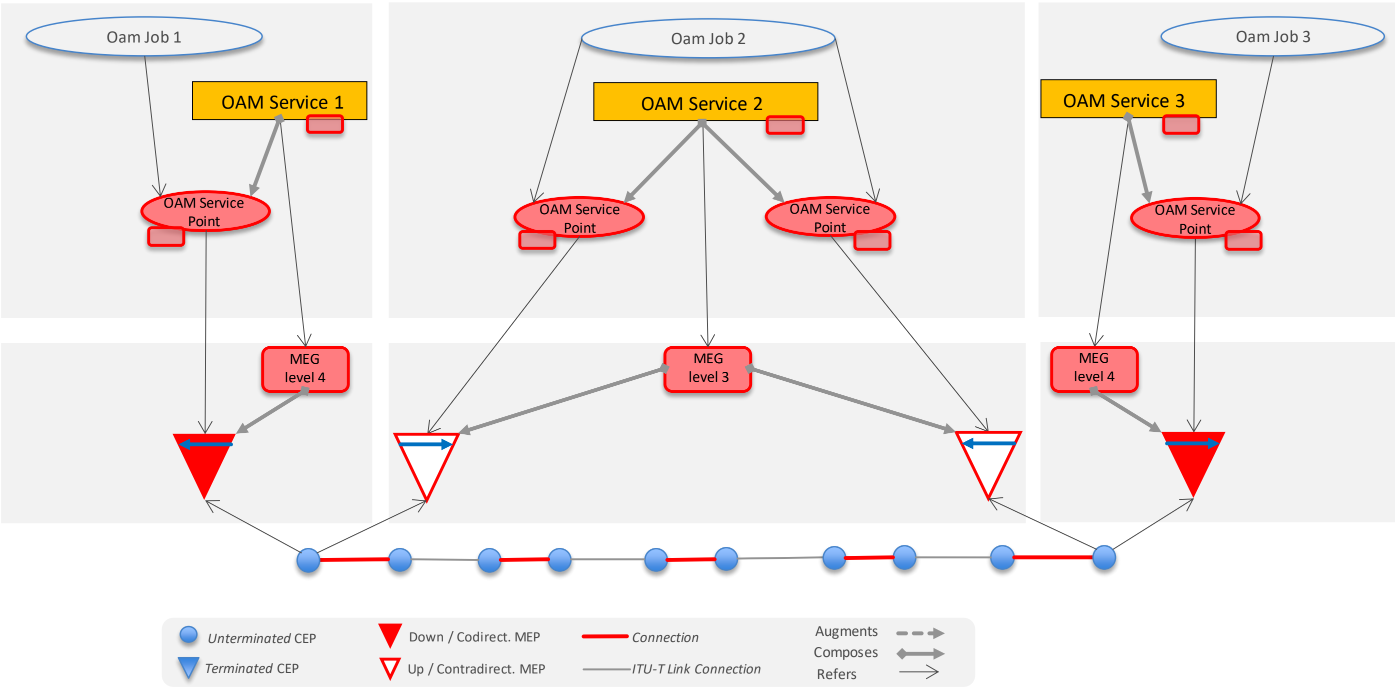


Figure 6-101 OAM Provisioning, Client Controller creates the OAM Jobs

Figure 6-102 shows the creation of Current and History Data instances according to OAM Job provisioning.

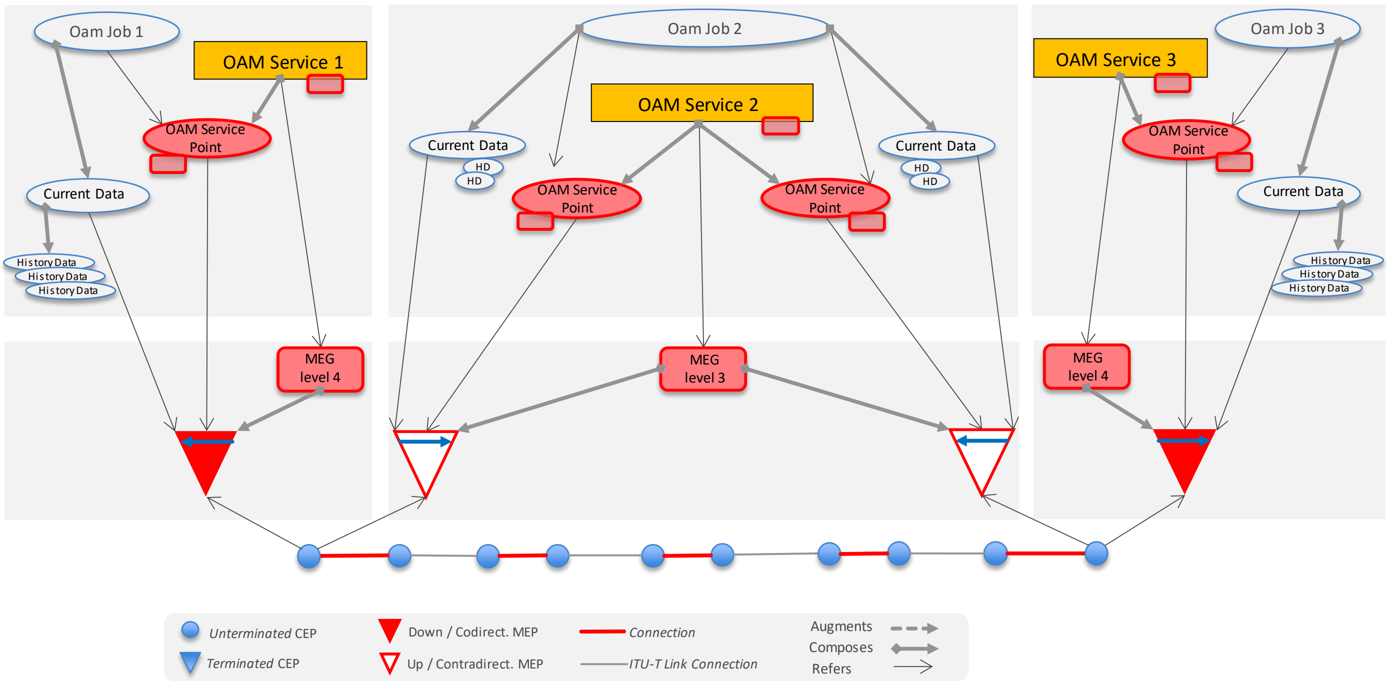


Figure 6-102 OAM provisioning, Server Controller creates Current and History Data instances

Figure 6-103 shows the *independent* provisioning in case of *DSR UNI to NNI (asymmetric) scenario*, monitoring functions are three TCM MEPs. CS is already existing.

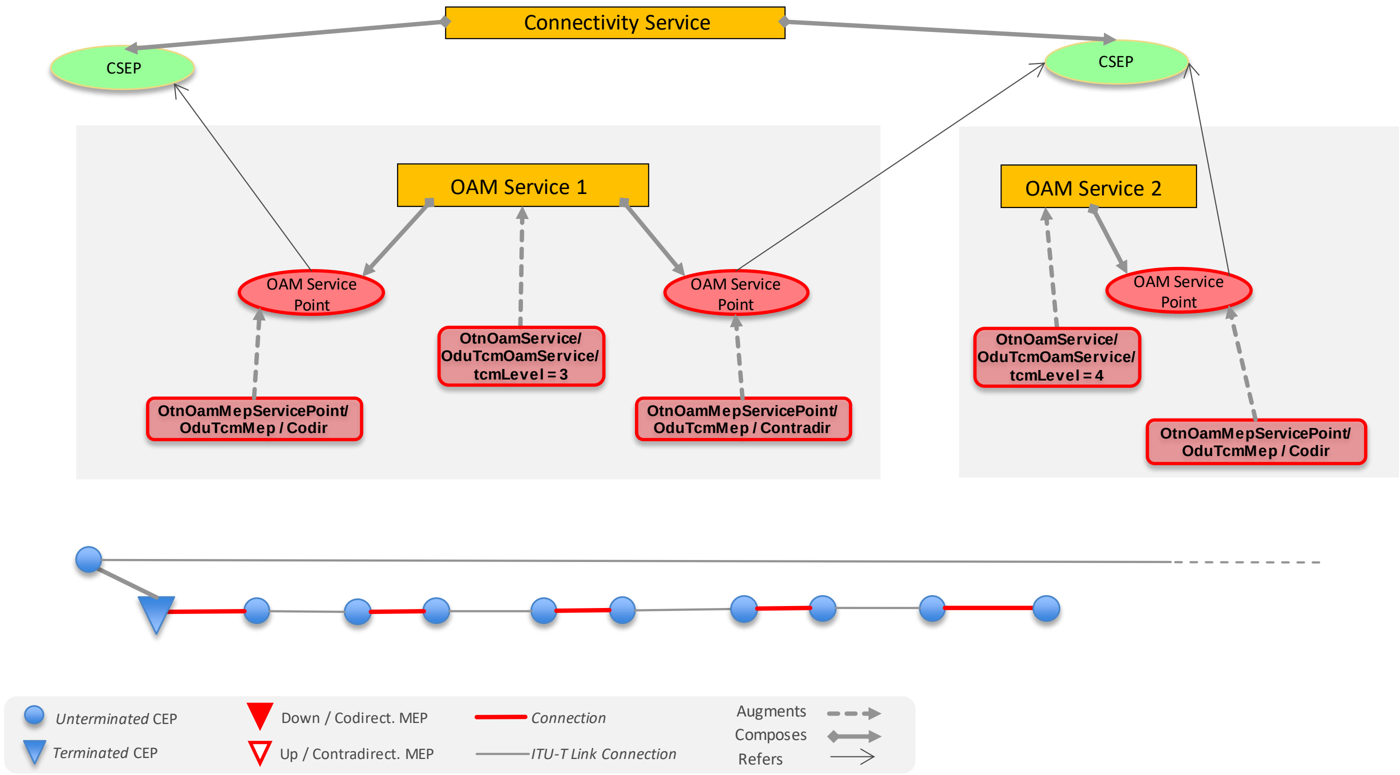


Figure 6-103 OAM provisioning, Client Controller creates the OAM Service and its End Points, DSR UNI to NNI

Figure 6-104 shows the result of Figure 6-103 provisioning.

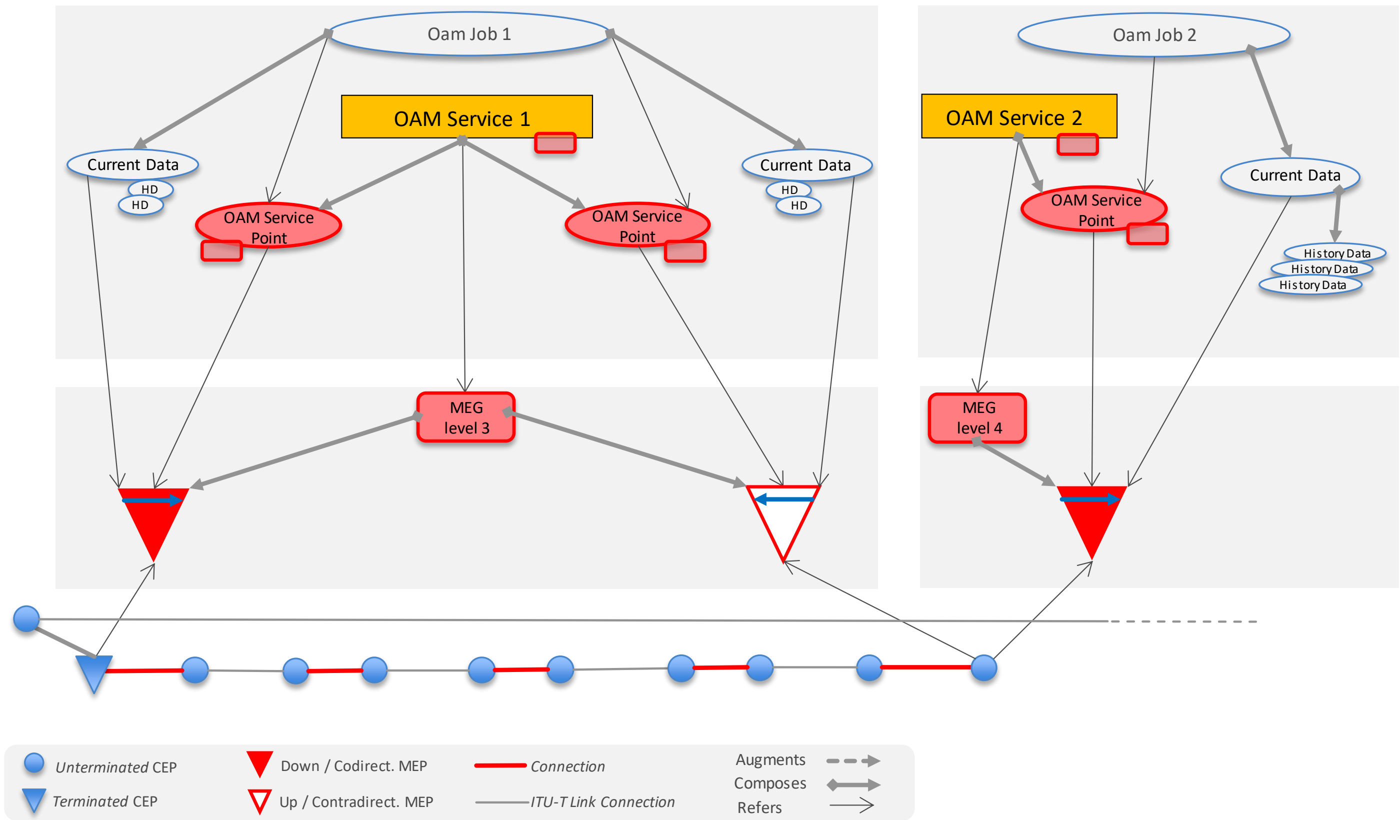


Figure 6-104 OAM provisioning, DSR UNI to NNI (asymmetric) scenario, result

These scenarios will be referred to in the use cases below.

6.8.2 OAM Profile

As mentioned, TAPI 2.4.0 introduces the generic concept of Profile (modelled as tapi-common:context/profile={uuid}) which is, in some cases, augmented by the OAM module (tapi-common:context/profile={{uuid}}/tapi-oam:oam-profile). An OAM Profile contains a list of Performance Monitoring (PM) data. A PM Parameter includes a PM metric and, where applicable, its use in the definition of a threshold. The pm-parameter-name identifies the PM metric (such as BBE, SES, UAS or DELAY).

Table 82: OAM **Profile**

OamProfile	**/tapi-common:context/profile={uuid}/tapi-oam:oam-profile**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
pm-data	List of { PmData } objects indexed by their local-id	RW	M	An OAM profile MUST have at least one PM Data instance.

Table 83: OAM **PM Data**

PmData	**/tapi-common:context/profile={uuid}/tapi-oam:oam-profile/pm-data[local-id]**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
local-id	String. Identifies the PM data within the profile	RW	M	• Local identifier of the PmData instance
name	Set of name value pairs.	RW	O	• Additional names for the PmData
applicable-job-type	A job type (identity with base OAM_JOB_TYPE)	RW	O	• Leaf-list of job types, to specify which jobs can refer to the specific OAM Profile
granularity-period	As defined in tapi-common:time-period: "value": value of the time period (uint64) "unit": one of YEARS, MONTHS, DAYS, HOURS, MINUTES, SECONDS, MILLISECONDS, MICROSECONDS, NANOSECONDS or PICOSECONS	RW	C	• Provided by TAPI client. • The granularity period or measurement interval time. • This attribute contains the discrete non overlapping periods of time during which measurements are available in the current data. At the end of the period a history data is created with the PM metric value. • Defines the integration period for thresholds. *NOTE: if granularity-period is not present, it means a single, one-shot, measurement collected in the current data and no history data is created.*

is-transient	Boolean. A threshold crossing alert (TCA) is transient when stateless, i.e., an explicit alarm clear notification is not foreseen. With stateless reporting, a TCA is generated in each Measurement Interval in which the threshold is crossed. With stateful reporting, a SET TCA is generated in the first Measurement Interval in which the threshold is crossed, and a CLEAR TCA is subsequently generated at the end of the first Measurement Interval in which the threshold is not crossed. Note: In ITU-T G.7710 terminology, stateless TCA reporting corresponds to a transient condition, and stateful TCA reporting corresponds to a standing condition.	RW	C	• MUST be used when the profile is used for threshold crossing AND there is not CLEAR threshold define.
pm-parameter	List of PM Parameters, keyed by their pm-parameter-name	RW	M	• List of Parameters that compose this profile and, if applicable, the threshold configuration. • The PM Data list of PM parameters MUST include at least one PM Parameter.

Table 84: OAM **PmParameter** definition

OAM PM Parameter		
Attribute	**Allowed Values/Format**	**Notes**
pm-parameter-name	tapi-common:pm	Key of the list element
threshold-config	List of Threshold configurations (threshold parameters)	If the profile does not include threshold configuration, this attribute MUST NOT be present.

Table 85: OAM Threshold Configuration definition

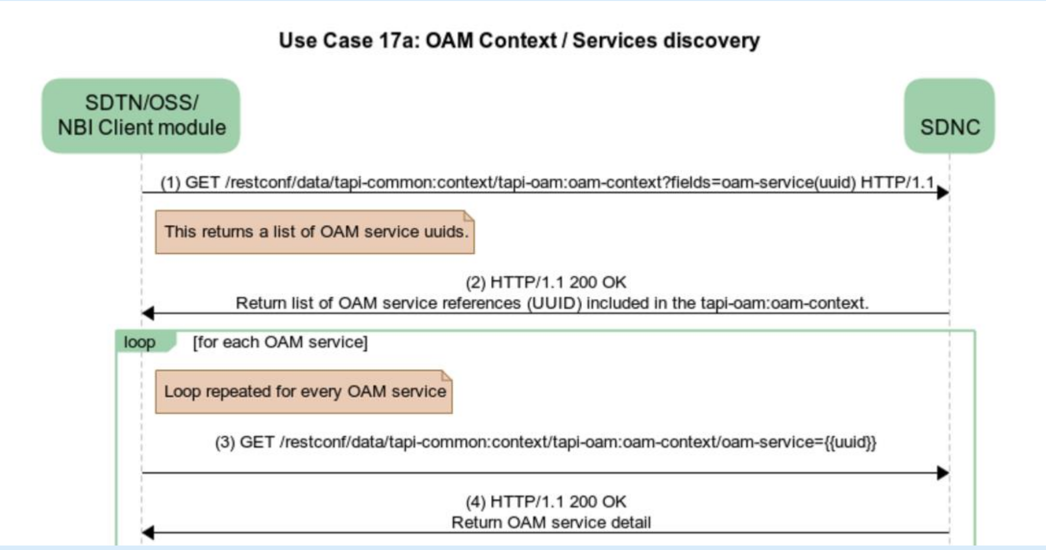
OAM Threshold Config		
Attribute	**Allowed Values/Format**	**Notes**
threshold-location	One of { NOT_APPLICABLE, NEAR_END, FAR_END, BIDIRECTIONAL, FORWARD, BACKWARD }	Specifies whether it is "Near End detection", "Far end detection.", "Composition of near and far end detections", or as per MEF 35.1 and MEF 83
threshold-type	Any identity that extends the THRESHOLD_TYPE base One of { THRESHOLD_TYPE_UPPER, THRESHOLD_TYPE_LOWER, THRESHOLD_TYPE_TIDEMARK, THRESHOLD_TYPE_POSITIVE_DELTA, THRESHOLD_TYPE_NEGATIVE_DELTA }	Defines the type of threshold that applies to the configuration.
pm-parameter-value	Includes "pm-parameter-value" : decimal64 "pm-parameter-unit" : string	Defines the parameter value and its unit. Specific use cases below may constraint the usage of the different attributes Units encoded as strings are capital letters e.g., "MILLISECONDS".

		In cases without a given unit, the pm-parameter-unit field MUST not appear.
clear-threshold	Boolean. If true, means that the value refers to a "CLEAR" of the threshold type	

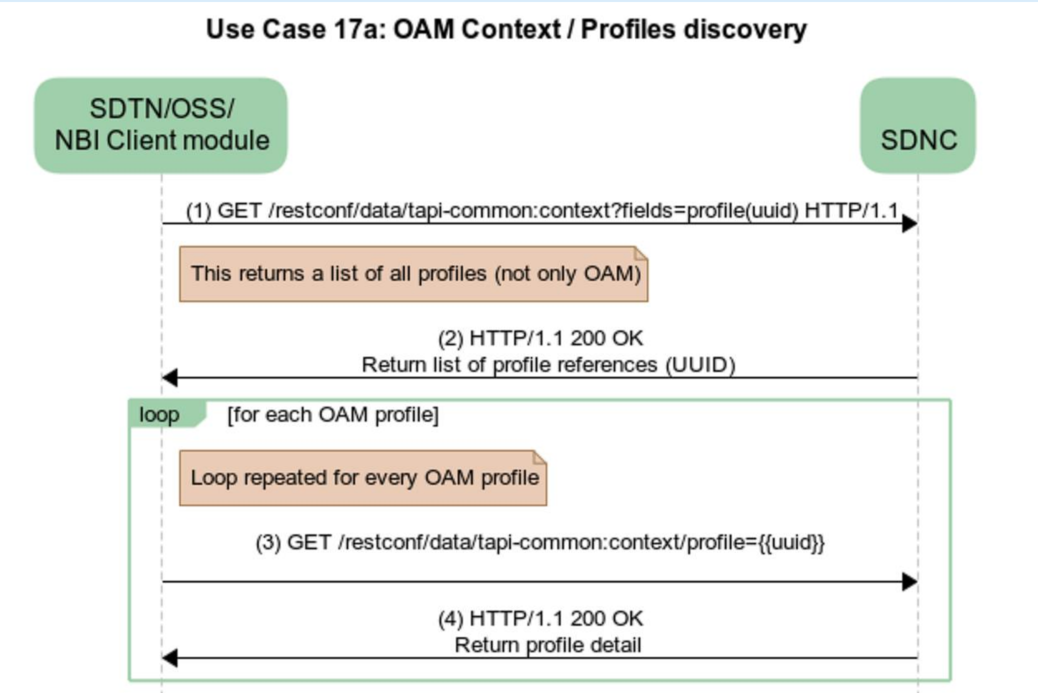
6.8.3 Use case 17a: OAM Profile and Context discovery

Number	**UC17a**
Name	**OAM Context discovery**
Technologies involved	All
Process/Areas Involved	OAM
Brief description	This use case consists of retrieving all information available from the TAPI server (SDN-C) regarding OAM Services and maintenance-entity-group (MEG) end-points. This use case is intended to be performed by any NBI client controller, module or application which intends to discover OAM Services and OAM Capabilities of a given network which is controlled by an SDN-C. In particular, the use case covers: i) retrieving the OAM services and endpoints; ii) retrieving the OAM jobs; iii) retrieving the OAM profiles from the TAPI context and iv) discovering the list of MEGs from the context, including the MEPs and MIPs (from a high-level perspective). NOTE: As previously, OAM information is also present in the connectivity-context. In all cases, CEPs MAY also have active monitoring points that have not been provisioned by the client, and PM parameters MAY be part of the CEP object without explicit configuration.
Layers involved	DSR, DIGITAL_OTN, PHOTONIC_MEDIA
Type	OAM
Description & Workflow	The first part of the workflow is the discovery of the OAM services. For this, the client performs a GET operation on the OAM context asking for the oam-service objects listing the uuids (1) and

retrieves the list of OAM service uuids (2). The client may later iterate each OAM service (3) and the server returns the data associated to the OAM service.



Profile discovery is as follows: the client performs a GET operation on the main TAPI context to retrieve the list of profile uuids (1), and the server returns the context including only the profiles uuids. Note that it is not possible for a client to GET OAM profiles only using a direct RESTCONF call. The client should filter based on the presence of the OAM augment. It is expected that profiles will be retrieved based on uuids present in other parts of the OAM and Connectivity context.



The case of OAM job is very similar to OAM services. The client requests the OAM job uuids (1)(2) and for each job, the client may retrieve the job data (3)(4).

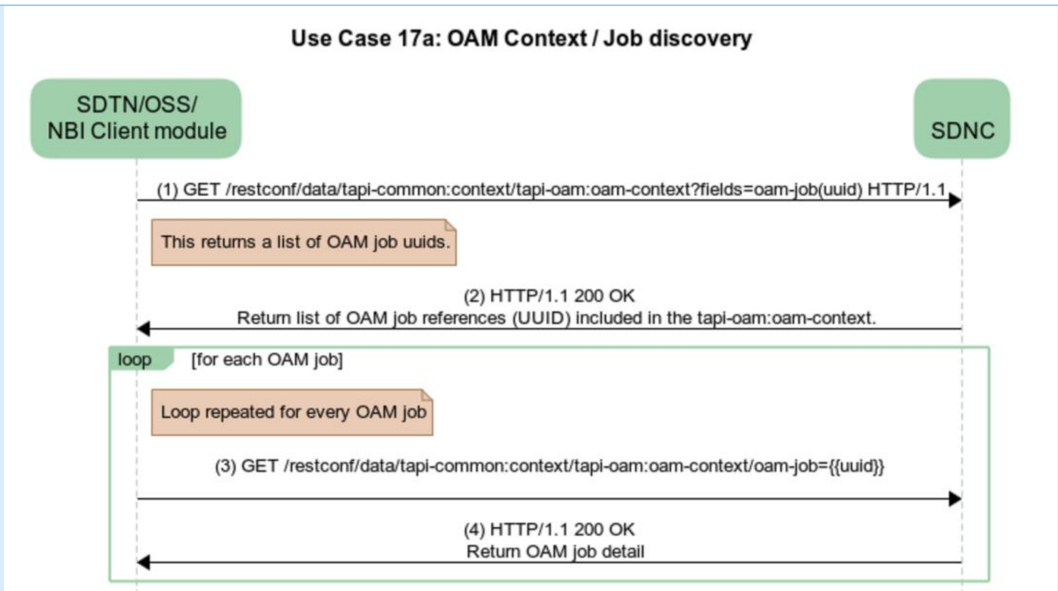


Figure 6-105 UC-17a: OAM Context discovery

To retrieve the list of MEGs:

The client (1) retrieves a list of MEG “uuid” (employing the “fields” query parameter), and the. This operation iteratively retrieves the “uuid” of each MEG object present in the OAM context (2).

For each MEG, the client (3) retrieves the list of maintenance-entity-group end-point (MEP) “local-id” within a MEG (by its “uuid”) and uses the “fields” query parameter. Similarly, the client (5) retrieves the list of maintenance-entity-group intermediate-point (MIP) “local-id” within a MEG (by its “uuid”) and employs the “fields” query parameter to obtain the desired information only.

It is then possible (7-8) to iteratively retrieve each MEP object details by its “local-id” and within a MEG by its “uuid” and each MIP (9,10).

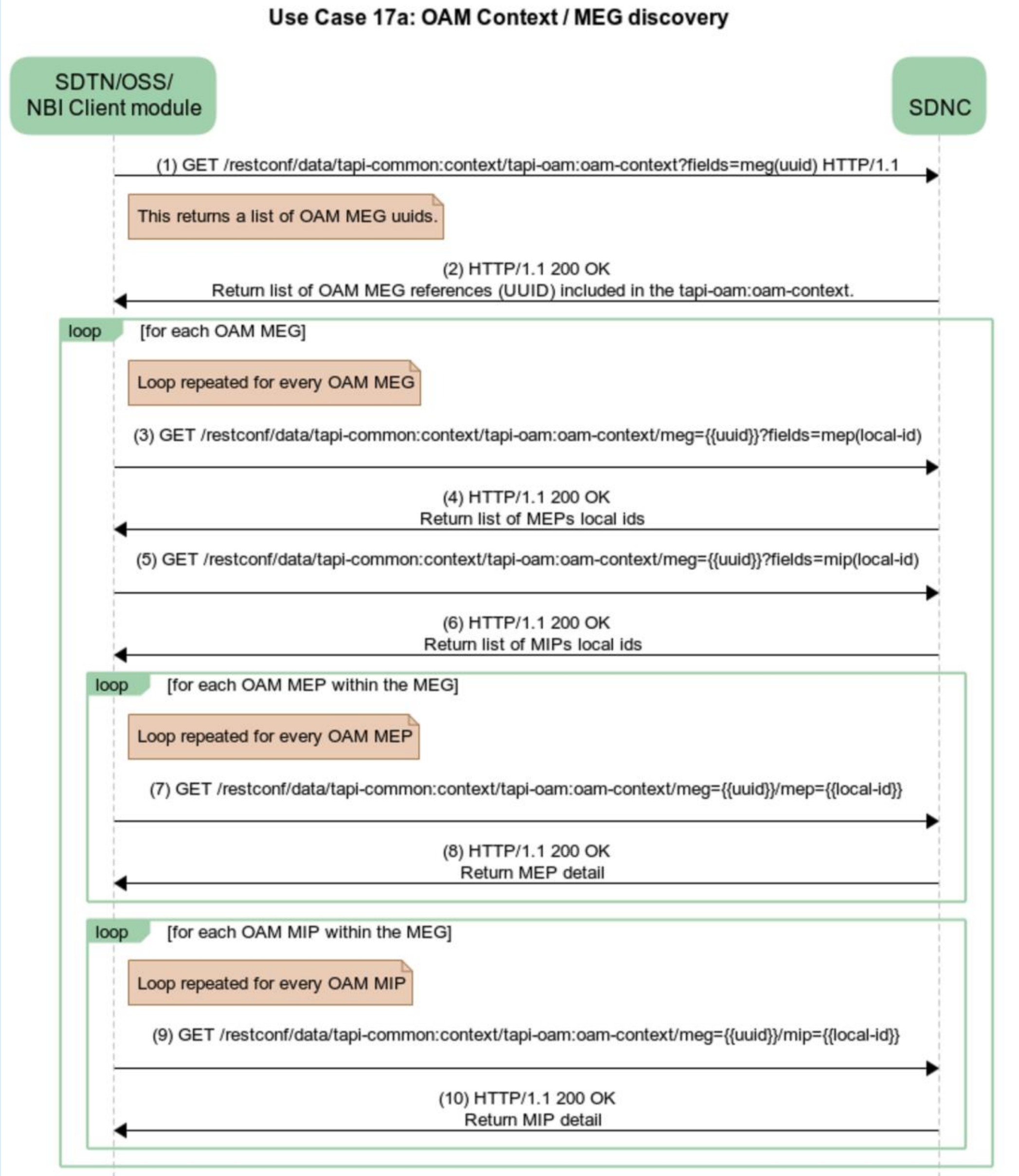


Figure 6-106 UC-17a: OAM MEG discovery

6.8.3.1 Relevant parameters

Table 86: OAM Service object definition

OamService	**/tapi-common:context/tapi-oam:context/oam-service/**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
oam-service-point	List of {**end-point**}, indexed by their local-id	RW	M	• Provided by *tapi-server* There MUST be at least one OAM Service Point.

meg	MEG uuid ref to /tapi-common:context/tapi-oam:oam-context/meg/uuid	RO	M	• Provided by *tapi-server* Once the OAM service has been created, this attribute MUST point to the allocated OAM MEG of the OAM context.
uuid	uuid of the OAM service	RW	M	• As per RFC 4122
name	List of value-name pairs	RW	M	• Provided by *tapi-server*
tapi-digital-otn:otn-oam-service/odu-tcm-oam-service/tcm-level	uint64 Specifies the TCM level for this OAM Service	RW	C	• Provided by *tapi-server* This attribute MUST be present in the case of ODU TCM Services

Table 87: OamServicePoint object definition

OamServicePoint	**/tapi-common:context/tapi-oam:context/oam-service/oam-service-point**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
service-interface-point	Service Interface Point ref (sip-ref) service-interface-point-uuid	RW	C	• Provided by *tapi-client.* These attributes are exclusive. • At least one MUST be present. • Specifies the OAM Service Points of the OAM service, providing the relation with the Connectivity model.
connectivity-service-end-point	Connectivity Service End Point ref	RW	C	
connection-end-point	CEP ref	RW	C	
layer-protocol-name	"DIGITAL_OTN" or "PHOTONIC_MEDIA"	RW	O	
layer-protocol-qualifier	Valid layer protocol qualifier	RW	O	
mep	Maintenance Entity group end Point ref mep-ref (meg uuid and mep local-id)	RO	C	• Provided by *tapi-server* • It is instantiated by the server and refers to the MEP as appropriate (see Section 6.8.1)
mip	Maintenance entity group Intermediate Point ref mip-ref (meg uuid and mip local-id)	RO	C	• Provided by *tapi-server* • It is instantiated by the server and refers to the MIP as appropriate (see Section 6.8.1) • *For a given MEG, MIPs may be present or not.*
is-mip	Boolean	RW	M	• Provided by *tapi-client*
local-id	string	RW	M	• Provided by *tapi-client*

name	List of {value-name: value}	RW	M	• Provided by *tapi-client*
tapi-digital-otn:otn-oam-mep-service-point	odu-mep **odu-tcm-mep** otu-mep	RW	C	• Provided by *tapi-client* • *NOTE: From a configuration perspective this RIA only considers TCM, other objects MAY be present.* *See UC17e for the configuration of this*
tapi-digital-otn:otn-oam-mip-service-point	odu-mip **odu-tcm-mip** otu-mip	RW	C	• Provided by *tapi-client* • *NOTE: From a configuration perspective this RIA only considers TCM, other objects MAY be present.* • *See UC17e for the configuration of this*

Table 88: **OAM Job** object definition

Note that in the context of discovery all the attributes shall be considered as RO.

oam-job	**/tapi-common:context/tapi-oam:context/oam-job**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
oam-job-type	Any entity the derives from OAM_JOB_TYPE	RW	M	• The type of the job when it was created.
oam-job-state	Any entity the derives from OAM_JOB_STATE	RO	M	• State of the job (active, not active or concluded).
oam-service-point	List of OAM Service Points ÇRefs, each being a pair { *oam-service-uuid, oam-service-point-local-id* } used to associate the job to one or more OAM service point.	RW	C	• These attributes are exclusive. NOTES: • If the job is associated to an OAM Service (and its Service points) the oam-service-point list MUST be non-empty. • If the job is associated to one or more CEPs, then the connection-end-point list MUST be non-empty (see UC17d.1) • If the job is created by the server upon request of a connectivity service (embedded provisioning scenario, UC 17b) the job connectivity service MUST point to such CSEP uuid
connection-end-point	List of CEP references, used to associate the job to such CEP instances.	RW	C	
connectivity-service-end-point	Reference to a Connectivity Service End Point used to associate the OAM Job (created by the embedded provisioning scenario).	RW	C	
profile	Reference to a profile (augmented with OAM capabilities) that contains the metric(s) and threshold(s) data for this job.	RW	C	• ***profile and pm-data are exclusive.*** • *A job is either created referring to an existing OAM profile OR with a list of PM data with the PM parameters for the job.* • *profile* is the reference to the OAM profile if a profile was used when creating the job (either directly or via an embedded OAM service)
pm-data	List of {PM Data}	RW	C	

				• *pm-data* contains a list of PM Data, Each PM data in turn a list of parameters threshold configuration.
current-data	List of { **current-data** } indexed by local-id The CurrentData instances in the scope of the OamJob.	RO	M	
schedule	Time range, i.e., { "start-time": date-and-time "end-time": date-and-time }	RW	O	• Provided by *tapi-server*. Defines the period where this job is active. If this is not specified, the schedule corresponds to the job object lifetime.
creation-time	TAPI tapi-common:date-and-time	RO	M	• Provided by *tapi-server*. Specifies the time point where the job is instantiated.
uuid	As per RFC4122	RW	M	• The uuid may be allocated by the server if the creation of the job is the result of NCM provisioning (UC 17b.1, 17b.2)
name	OAM job list of name value pairs.	RW	O	• Provided by *tapi-server*
results	String that specifies alternative means to retrieve PM data (e.g., a filename)	RO	O	• For further study.

A MEG is fundamentally a global object within the OAM context that encompasses a list of MEPs and MIPs.

Table 89: **MEG** object definition

MEG	**/tapi-common:context/tapi-oam:context/meg**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
uuid	As per RFC4122	RO	M	• Provided by *tapi-server*
name	List of {value-name, value}	RO	M	• Provided by *tapi-server*
mip	List of {**mip**}	RO	C	• Provided by *tapi-server* • *Depends on the Use Case*
mep	List of {**mep**}	RO	C	• Provided by *tapi-server* • *Depends on the Use Case*
tapi-digital-otn:otn-meg-spec/odu-tcm-meg/tcm-level	uint64	RO	C	• Provided by *tapi-server* • *Specifies the TCM level for this MEG*

Table 90: **MEP** object definition

MEP	**/tapi-common:context/tapi-oam:context/meg/mep**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
layer-protocol-name	"DIGITAL-OTN"	RO	M	• Provided by *tapi-server*
layer-protocol-qualifier	A valid protocol qualifier	RO	M	• Provided by *tapi-server*
local-id	string	RO	M	• Provided by *tapi-server*
name	list of {value-name, value}	RO	M	• Provided by *tapi-server*

tapi-digital-otn:otn-mep-spec	Includes { odu-mep otu-mep odu-tcm-mep }	RO	C	• This attribute contains the ODU MEP
tapi-digital-otn:otn-mep-spec/odu-mep	txti: string otn-oam-common { ex-dapi ex-sapi deg-thr tim-det-mode tim-act-disabled deg-m } odu-mep-status { acti tcm-fields-in-use [] }	RO	C	• ODU MEP parameters. • Note: this RIA only considers the independent provisioning scenario (which instantiates MIP and MEP) for TCM. "ex-dapi": The Expected Destination Access Point Identifier (ExDAPI), provisioned by the managing system, to be compared with the TTI accepted at the overhead position of the sink for the purpose of checking the integrity of connectivity "ex-sapi": The Expected Source Access Point Identifier (ExSAPI), provisioned by the managing system, to be compared with the TTI accepted at the overhead position of the sink for the purpose of checking the integrity of connectivity "deg-thr" the threshold level for declaring a performance monitoring (PM) Second to be bad. The value of the threshold can be provisioned in terms of number of errored blocks or in terms of percentage of errored blocks. For percentage-based specification, in order to support provision of less than 1%, the specification consists of two fields. The first field indicates the granularity of percentage. For examples, in 1%, in 0.1%, or in 0.01%, etc. The second field indicates the multiple of the granularity. For number of errored block based, the value is a positive integer. See UC 17b for details
tapi-digital-otn:otn-mep-spec/otu-mep	txti: string otn-oam-common { ex-dapi ex-sapi deg-thr tim-det-mode tim-act-disabled deg-m } otu-mep-status { acti : string } fec-monitoring: boolean fec-corrected-error-threshold: uint64	RO	C	• OTU MEP parameters • Note: this RIA only considers the independent provisioning scenario (which instantiates MIP and MEP) for TCM. See UC 17b for details
tapi-digital-otn:otn-mep-spec/odu-tcm-mep	codirectional tcm-level position-sequence tcm-extension tcm-mode admin-state-source admin-state-sink txti: otn-oam-common { ex-dapi ex-sapi	RO	M	• ODU TCM MEP parameters Note: this RIA only considers the independent provisioning scenario (which instantiates MIP and MEP) for TCM.

	deg-thr tim-det-mode tim-act-disabled deg-m } otu-tcm-mep-status { tcm-field ac-status-source ac-status-sink operational-state acti }			

Table 91: **MIP** object definition

MIP	**/tapi-common:context/tapi-oam:context/meg/mip**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
layer-protocol-name	"DIGITAL_OTN"	RO	M	• Provided by *tapi-server*
layer-protocol-qualifier	A valid protocol qualifier	RO	M	• Provided by *tapi-server*
local-id	string	RO	M	• Provided by *tapi-server*
name	list of {value-name, value}	RO	M	• Provided by *tapi-server*
tapi-digital-otn:otn-mip-spec	Includes { odu-mip odu-tcm-mip }	RO	C	• ODU MIP parameters
tapi-digital-otn:otn-mip-spec/odu-mip	otn-oam-common { ex-dapi ex-sapi deg-thr tim-det-mode tim-act-disabled deg-m } codirectional odu-mip-status { acti tcm-fields-in-use [] odu-current-number-of-tributary-slots }	RO	C	• ODU MIP parameters. • Note: this RIA only considers the independent provisioning scenario (which instantiates MIP and MEP) for TCM.
tapi-digital-otn:otn-mip-spec/odu-tcm-mip	codirectional otn-oam-common { ex-dapi ex-sapi deg-thr tim-det-mode tim-act-disabled deg-m } otu-tcm-mip-status { tcm-field operational-state acti } position-sequence	RO	M	• ODU TCM MEP parameters • Note: this RIA only considers the independent provisioning scenario (which instantiates MIP and MEP) for TCM.

Table 92: Current Data instance of an OAM Job

current-data	**/tapi-common:context/tapi-oam:context/oam-job/current-data**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
				All attributes are provided by tapi server
local-id	string	RO	M	*Current data instances are local objects of a given Job*
name	list of {value-name, value}	RO	O	
period-start-time	date-and-time	RO	M	This attribute indicates the start time of the current monitoring interval / granularity period (or the single period in the case of one-shot measurements). The value is bound to the quarter of an hour in case of a 15-minute interval and bound to the hour in case of a 24-hour interval.
elapsed-time	time-interval with period: list of { value, unit}	RO	M	Q822: This attribute represents the difference between the current time and the start of the present interval
pm-data-pac/granularity-period pm-data-pac/suspect-interval-flag	time-interval, with period: list of { value, unit} boolean	RO	C	Parameters specific to Performance Monitoring functions.
mep	tapi-oam:mep-ref Maintenance Entity group end Point ref {/tapi-common:context/tapi-oam:oam-context/meg/uuid, /tapi-common:context/tapi-oam:oam-context/meg/mep/local-id}	RO	C	The current data refers, exclusively, to either a CEP, a MEP, or a MIP
mip	tapi-oam:mip-ref Maintenance entity group Intermediate Point ref {/tapi-common:context/tapi-oam:oam-context/meg/uuid, /tapi-common:context/tapi-oam:oam-context/meg/mip/local-id }	RO	C	
connection-end-point	tapi-connectivity:connection-end-point-ref	RO	C	
history-data	list of { history-data }	RO	C	See table below
digital-otn:otu-fec-performance-data	*OTU FEC Performance Data*	RO	C	Conditioned to the type of data. See Table 93
digital-otn:otn-error-performance-data	*OTN Error Performance Data*	RO	C	Conditioned to the type of data, See Table 94
digital-otn:odu-delay-performance-data	*ODU Error Performance Data*	RO	C	Conditioned to the type of data, See Table 95

tapi-photonic-media:optical-power-performance-data	*Optical Power Performance Data*	RO	C	Conditioned to the type of data, See Table 96 (added TAPI 2.4.1)

Table 93: **OTU FEC** Performance Data

OTU FEC Perf Data	**/tapi-common:context/tapi-oam:context/oam-job/current-data/digital-otn:otu-fec-perfomance-data**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
For all the following attributes, its presence is conditioned to the requested PM Data. For OTU FEC Perf. Data, this RIA only considers the PM_PARAMETER_NAME_FEC_CORRECTED_ERROR, so in such case, only fec-corrected-errors-count is *Mandatory* and the rest is optional.				
fec-corrected-errors-count	uint64	RO	M	
pre-fec-ber	decimal64	RO	O	Bit error rate before correction by FEC
post-fec-ber	decimal64	RO	O	Bit error rate after correction by FEC.
uncorrectable-bytes	uint64	RO	O	Bytes that could not be corrected by FEC
uncorrectable-bits	uint64	RO	O	Bits that could not be corrected by FEC
corrected-bytes	uint64	RO	O	Bytes corrected between those that were received corrupted

Table 94: **OTN Error** Performance Data

OTN Error Perf Data	**/tapi-common:context/tapi-oam:context/oam-job/current-data/digital-otn:otn-error-perfomance-data**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
For all the following attributes, its presence is conditioned to the requested PM Data.				
near-end-otn-counters	includes bbe, ses, uas as uint64	RO	C	
far-end-otn-counters	includes bbe, ses, uas as uint64	RO	C	
bidirectional-uas	uint64	RO	C	
codirectional	boolean	RO	C	
otn-cn-error-performance-data	List of OTN Error Perf. Data indexed by otn-cn-oh-index (near-end-odu-counter, etc.)	RO	C	

Table 95: **ODU Delay** Performance Data

ODU Error Perf Data	**/tapi-common:context/tapi-oam:context/oam-job/current-data/odu-error-perfomance-data**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
For all the following attributes, its presence is conditioned to the requested PM Data.				
delay-frame-count	uint64	RO	C	
delay-measure-success	boolean	RO	C	

Table 96: **Optical Power** Performance Data (TAPI 2.4.1)

Optical Power Perf Data	**/tapi-common:context/tapi-oam:context/oam-job/current-data/tapi-photonic-media:optical-power-perfomance-data**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
For all the following attributes, its presence is conditioned to the requested PM Data.				
power-measurement-pac	Includes { measured-input-power and measured-output-power } both with total-power and power-spectral-density	RO	C	• Provided by *tapi-server* *Depends on hw power monitoring capabilities*

Table 97: History data

history-data	**/tapi-common:context/tapi-oam:context/oam-job/current-data/history-data**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
All attributes are provided by tapi server				
local-id	string	RO	M	*History data instances are local objects*
name	list of {value-name, value}	RO	O	
period-start-time	date-and-time	RO	M	
period-end-time	date-and-time	RO	M	
pm-data-pac/granularity-period pm-data-pac/suspect-interval-flag	time-interval, with period: list of { value, unit} boolean	RO	M	Parameters specific to Performance Monitoring functions. granularity-period: the granularity period or measurement interval time
tapi-digital-otn:otu-fec-performance-data	As in Current Data	RO	C	Conditioned to the use case
tapi-digital-otn:otn-error-performance-data	As in Current Data	RO	C	Conditioned to the use case
tapi-digital-otn:odu-delay-performance-data	As in Current Data	RO	C	Conditioned to the use case
tapi-photonic-media:optical-power-performance-data	As in Current Data	RO	C	Conditioned to the use case TAPI 2.4.1

6.8.4 Use case 17b: OAM Provisioning using the embedded provisioning scenario (NCM)

6.8.4.1 Sub-Case 1: NCM Provisioning for DSR over ODU CS (for BBE, SES, UAS)

Number	**17b.1**

Name	NCM Provisioning for DSR over ODU CS (BBE, SES, UAS)
Technologies involved	DSR, DIGITAL_OTN
Process/Areas Involved	OAM
Brief description	The UC17b.1 describes the provisioning of a Network Connection Monitoring (NCM) using the provisioning of a DSR ***tapi-connectivity:connectivity-service*** instance between DSR SIPs. The UC involves a DSR over ODU connectivity service (e.g., between transponder client ports) such as a 10G over ODU2; 100G over ODU4 or x00G over ODUCN. This use case only covers symmetric and point to point connectivity services and enables the monitoring of the ODU top-connection.
Layers involved	DSR/DIGITAL_OTN
Type	OAM
Description & Workflow	This UC involves two parts: i) the provisioning of the DSR connectivity service where the OAM parameters are included in the Connectivity Service End Points (CSEPs). This UC addresses two aspects: 1) PM parameter monitoring (history of the BBE, SES, UAS) and 2) threshold crossing alert configuration. At least one CSEP MUST include a tapi-oam:**connectivity-oam-job data** which defines the parameters that will be monitored. The connectivity-oam-job within the CSEP can either use a reference to an existing OAM Profile (via profile/profile-uuid) or include the information regarding the PM Data directly in the pm-data. Note that: - the creation of the CS triggers the creation of the corresponding OAM job (tapi-oam:oam-context/oam-job), which can be later discovered following UC17a or Notifications/Streaming. - with this workflow, the OAM job uuid is allocated by the server and the OAM job contains a reference to the CSEP it (via **tapi-oam:oam-job/connectivity-service-end-point/connectivity-service-uuid and tapi-oam:oam-job/connectivity-service/end-point/connectivity-service-end-point-local-id**). - the created OAM job lifetime *is not* bound to the CS. If the CS is deleted, the job goes to the CONCLUDED state (and the connectivity-service-end-point reference is removed). Job deletion is out of scope (can be triggered by the client or upon policy). - there is no allocation of an OAM Service. MEGs and MIP/MEP are not instantiated in the OAM context. Only a Job current and history data can contain error performance data. The workflow is as follows: - Create the Connectivity Service using a POST (UC 1.X) (1)-(2) and including the embedded OAM data in the corresponding CSEP. The CSEP includes OAM Job Data (connectivity-oam-job) along with the embedded OAM Service (connectivity-oam-service/connectivity-oam-service-point, connectivity-oam-service/otn-oam-mep-service-point and connectivity-oam-service/otn-oam-mip-service-point) which shall be reflected into the corresponding CEPs.

- Retrieve the CEPs (if needed)

- Retrieve the Job data (see UC 17a)

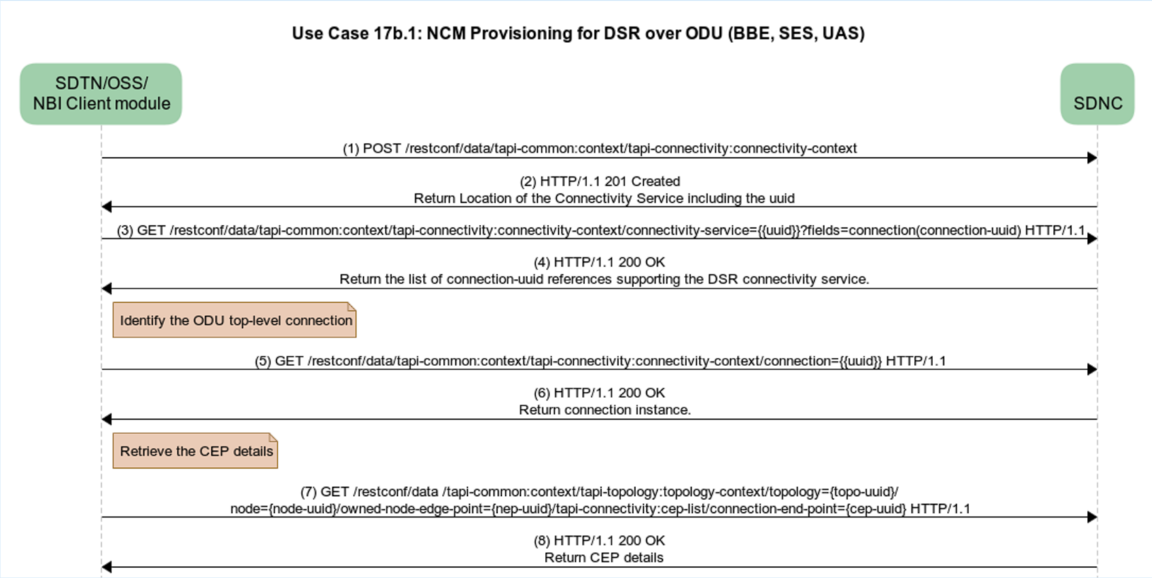


Figure 6-107 UC-17b.1: NCM DSR over ODU with BBE, SES, UAS

6.8.4.1.1 Relevant parameters

Table 98: Connectivity-service End Point (**CSEP**) OAM Job object definition (UC17b)

end-point/tapi-oam:connectivity-oam-job	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/end-point/tapi-oam:connectivity-oam-job**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
oam-job-type	ODU_OAM_JOB_TYPE_NCM	RW	M	• Provided by TAPI client
profile	Reference to a profile (via profile-uuid)	RW	C	• Provided by TAPI client • *This is used when the Job refers to an existing profile. It MUST NOT be used jointly with pm-data*
schedule	start-time and end-time			• Provided by TAPI client
pm-data	See PM Data Table 83			• Provided by TAPI client • *This is used when the job does not refer to an existing profile. It MUST NOT be used jointly with profile*

Table 99: Connectivity-service-end-point (**CSEP**) OAM Service Point definition (UC17b)

end-point/tapi-oam:connectivity-oam-service	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/end-point/tapi-oam:connectivity-oam-service**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
tapi-oam:connectivity-oam-service-point	List of Connectivity OAM service point, See next table	RW	M	This list MAY appear in both CSEPs of a given CS.
tapi-digital-otn:otn-oam-mep-service-point	odu-mep odu-tcm-mep (not used) otu-mep (not used)	RW	C	
tapi-digital-otn:otn-oam-mip-service-point	odu-mip odu-tcm-mip (not used)	RW	C	

Table 100: Connectivity-service-end-point (**CSEP**) OAM Service Point definition (UC17b)

end-point/tapi-oam:connectivity-oam-service/connectivity-oam-service-point	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/end-point/tapi-oam:connectivity-oam-service/connectivity-oam-service-point** Used to instantiate MEP and MIP composed on the CEPs.			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
is-mip	Boolean. True is the OAM Service Point is a MIP	RW	M	Provided by tapi-client. For UC17b this value is assumed false.
layer-protocol-name	DIGITAL_OTN	RW	M	The UC covers ODU NCM
layer-protocol-qualifier	ODUX	RW	M	Depends on the actual low order ODU (e.g., ODU2 for a 10G DSR service)
local-id	Local identifier	RW	M	
name	Name value pairs	RW	O	
tapi-digital-otn:otn-oam-mep-service-point/odu-mep	See Table 90 for a generic description (and below tables)	RW	C	Provided by tapi-client MUST NOT be present if is-mip is true.
tapi-digital-otn:otn-oam-mip-service-point/odu-mip	See Table 91 for a generic description (and below tables)	RW	C	Provided by tapi-client MUST NOT be present if is-mip is false. For UC17b this not present

Table 101: Connectivity-service-end-point (**CSEP**) OAM Service Point OTN/ODU MEP definition (UC17b)

odu-mep	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/end-point/tapi-oam:connectivity-oam-service/connectivity-oam-service-point**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
txti	String	RW	M	The Trail Trace Identifier (TTI) information, provisioned by the managing system at the termination source, to be placed in the TTI overhead position of the source of a trail for transmission (see ITU-T G.874)

				Allows the device to identify the TTI mismatch and raise the appropriate alarm.
otn-oam-common/ex-dapi, otn-oam-common/ex-sapi	Strings	RW	M	Expected SAPI/DAPI. Jointly with txti allows to identify the TTI mismatch.
otn-oam-common/deg-m	Integer number of seconds.	RW	M	Degrade threshold (deg-m) the threshold level for declaring a Degraded Signal defect (dDEG). A dDEG shall be declared if DegM consecutive bad PM Seconds are detected
otn-oam-common/tim-det-mode, otn-oam-common/tim-act-disabled	tim-det-mode: one of { DAPI, SAPI, BOTH, OFF } (enum) tim-act-disabled: boolean	RW	M	Det Mode indicates the mode of the Trace Identifier Mismatch (TIM) Detection function allowed values: OFF, SAPIonly, DAPIonly, SAPIandDAPI ACT Disabled provides the control capability for the managing system to enable or disable the Consequent Action function when detecting Trace Identifier Mismatch (TIM) at the trail termination sink
otn-oam-common/deg-thr	deg-thr-type: one of { PERCENTAGE, NUMBER_ERRORED_BLOCKS} Determines applicability of the next two parameters: deg-thr-value (uint64) percentage-granularity: one of { ONES, ONE_TENTHS, ONE_HUNDREDS, ONE_THOUSANDS }	RW	M	Configures the threshold level for declaring a performance monitoring (PM) Second to be bad. The value of the threshold can be provisioned in terms of number of errored blocks or in terms of percentage of errored blocks. For percentage-based specification, in order to support provision of less than 1%, the specification consists of two fields. The first field indicates the granularity of percentage. For examples, in 1%, in 0.1%, or in 0.01%, etc. The second field indicates the multiple of the granularity. For number of errored block based, the value is a positive integer. Example: 0.3% is value: 3 and percentage-granularity = “ONE_TENTHS”
odu-mep-status	acti: string tcm-fields-in-use list of uint64	RO	M	acti: The Trail Trace Identifier (TTI) information recovered (Accepted) from the TTI overhead position at the sink of a trail tcm-fields-in-use: This attribute indicates the used TCM fields of the ODU OH

Table 102: Connectivity-service-end-point (**CSEP**) OAM Service Point OTN/ODU MIP definition (UC17b)

odu-mip	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/end-point/tapi-oam:connectivity-oam-service/connectivity-oam-service-point**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
codirectional	boolean	RW	M	This attribute specifies the directionality of the ODU MIP with respect to the associated ODU CEP. The value of TRUE means that the (half MIP/sink part of the) ODU MIP receives the same signal direction as the sink part of the ODU CEP. The Source part behaves similarly. This attribute is meaningful only on objects instantiated under ODU CEP, and at least one among ODU CEP and the subordinate object is bidirectional.
otn-oam-common	As above	RW	M	
odu-mip-status	acti: string tcm-fields-in-use list of uint64 As above	RO	M	odu-current-number-of-tributary-slots applies only to ODUflex(GFP) connections. It represents the current number of tributary slots allocated to this ODUflex(GFP) connection in the HO-ODU server layer

	odu-current-number-of-tributary-slots			

Table 103: Connection-end-point (**CEP**) ODU object definition (UC17b)

tapi-digital-otn:odu-connection-end-point-spec	**/tapi-common:context/tapi-topology:topology-context/topology/node/owned-node-edge-point/tapi-connectivity:cep-list/connection-end-point/tapi-digital-otn:odu-connection-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
	See Table 39 along with Table 101 and Table 102	RO	M	Provided by tapi-server

For this UC the applicable PM Parameter are:

OAM PM Parameter		
Attribute	**Allowed Values/Format**	**Notes**
pm-parameter-name	One of ODU_PM_PARAMETER_NAME_BBE ODU_PM_PARAMETER_NAME_SES ODU_PM_PARAMETER_NAME_UAS	
threshold-config	List of Threshold configurations (threshold parameters)	

OAM Threshold Config		
Attribute	**Allowed Values/Format**	**Notes**
threshold-location	One of { NEAR_END, FAR_END, BIDIRECTIONAL }	Bidirectional is considered for the UAS
threshold-type	Any identity that extends the THRESHOLD_TYPE base One of { THRESHOLD_TYPE_UPPER, THRESHOLD_TYPE_LOWER, THRESHOLD_TYPE_UPPER_MAX, THRESHOLD_TYPE_UPPER_MIN, THRESHOLD_TYPE_LOWER_MAX, THRESHOLD_TYPE_LOWER_MIN, }	Defines the type of threshold that applies to the configuration.
pm-parameter-value	Includes "pm-parameter-value" : decimal64 "pm-parameter-unit" : string	Defines the parameter value and its unit. Specific use cases below may constraint the usage of the different attributes Units encoded as strings are capital letters e.g., "MILLISECONDS". In cases without a given unit, the pm-parameter-unit field MUST not appear.
clear-threshold	Boolean. If true, means that the value refers to a "CLEAR" of the threshold type	

6.8.4.2 Sub-Case 2: NCM Provisioning for DSR over ODU CS (DELAY)

Number	**17b.2**
Name	NCM Provisioning for DSR over ODU (DELAY)

Technologies involved	DSR, ODU
Process/Areas Involved	OAM
Brief description	The UC17b.1 describes the provisioning of a Network Connection Monitoring using the provisioning of a DSR ***tapi-connectivity:connectivity-service*** instance between DSR SIPs. The UC involves a DSR over ODU connectivity service (e.g., between transponder client ports) such as a10G over ODU2, 100G over ODU4 or x00G over ODUCN. This use case only covers symmetric and point to point connectivity services and enables the monitoring of the ODU top-connection.
Layers involved	DSR/ODU
Type	OAM
Description & Workflow	This Use Case is similar to UC 17b1, with the parameters specified below.

6.8.4.2.1 Relevant parameters

For this UC the applicable PM Parameter are:

OAM PM Parameter		
Attribute	**Allowed Values/Format**	**Notes**
pm-parameter-name	ODU_PM_PARAMETER_NAME_DELAY	
threshold-config	List of Threshold configurations (threshold parameters)	

OAM Threshold Config		
Attribute	**Allowed Values/Format**	**Notes**
threshold-location	NEAR_END	Bidirectional is considered for the UAS
threshold-type	Any identity that extends the THRESHOLD_TYPE base One of { THRESHOLD_TYPE_UPPER, THRESHOLD_TYPE_LOWER }	Defines the type of threshold that applies to the configuration.
pm-parameter-value	Includes "pm-parameter-value" : decimal64 "pm-parameter-unit" : “MILLISECONDS”	Defines the parameter value and its unit.
clear-threshold	Boolean. If true, means that the value refers to a "CLEAR" of the threshold type	

6.8.4.3 Sub-Case 3: NCM Provisioning for OTU (FEC Corrected Errors)

Number	**UC17c**

Name	**NCM Provisioning for FEC Corrected Errors**
Technologies involved	DIGITAL_OTN
Process/Areas Involved	OAM
Brief description	The UC consists in the configuration of the OAM to be able to retrieve the otu-fec-performance data. This data is available in the JOB current and history data as shown in UC17a. Notes: - For OTU FEC Perf. Data, this RIA only considers the PM_PARAMETER_NAME_FEC_CORRECTED_ERROR, so in such case, only fec-corrected-errors-count is *Mandatory*, and the rest is optional. - The usage of this UC for Pre-FEC BER and Post-FER BER monitoring and TCA is left for further study.
Layers involved	DIGITAL_OTN
Type	OAM
Description & Workflow	From a workflow perspective, this Use Case is similar to UC 17b1, with the parameters specified below.

6.8.4.3.1 Relevant parameters

For this UC the applicable PM Parameter are:

OAM PM Parameter		
Attribute	**Allowed Values/Format**	**Notes**
pm-parameter-name	PM_PARAMETER_NAME_FEC_CORRECTED_ERROR	
threshold-config	List of Threshold configurations (threshold parameters)	

OAM Threshold Config		
Attribute	**Allowed Values/Format**	**Notes**
threshold-location	NEAR_END	
threshold-type	THRESHOLD_TYPE_UPPER,	Defines the type of threshold that applies to the configuration.
pm-parameter-value	Includes "pm-parameter-value" : decimal64 "pm-parameter-unit" : string	Defines the parameter value and its unit. Specific use cases below may constraint the usage of the different attributes Units encoded as strings are capital letters e.g., "MILLISECONDS".

		In cases without a given unit, the pm-parameter-unit field MUST not appear.
clear-threshold	Boolean. If true, means that the value refers to a "CLEAR" of the threshold type	

6.8.5 Use case 17c: Configuration of an OAM profile

<table>
<tr><td>Number</td><td>17c</td></tr>
<tr><td>Name</td><td>Configuration of an OAM profile</td></tr>
<tr><td>Technologies involved</td><td>DIGITAL_OTN, PHOTONIC_MEDIA</td></tr>
<tr><td>Process/Areas Involved</td><td>OAM</td></tr>
<tr><td>Brief description</td><td>The UC17c targets the configuration of an OAM profile. An OAM Profile is a global class, stored within the TAPI server context and allows centralization of OAM provisioning aspects, e.g., the PM parameters and their threshold values.

The clients may create an OAM profile including its uuid and optional name value pairs. The OAM profile contains a list of PM threshold data which, in turn, contains a list of threshold-parameters. Once created, the OAM profile may be referred to when creating OAM Services or in the Embedded provisioning model.

Note that if an OAM Job is created as a result of a connectivity provisioning, the connectivity-oam-job object that is added to the connectivity service already allows to specify threshold-parameters. It is not needed to create a dedicated OAM profile. That said, the connectivity-oam-job MAY refer to an existing profile</td></tr>
<tr><td>Layers involved</td><td>DSR/DIGITAL_OTN/PHOTONIC_MEDIA</td></tr>
<tr><td>Type</td><td>OAM</td></tr>
<tr><td>Description & Workflow</td><td>This use case involves the creation of a OAM profile

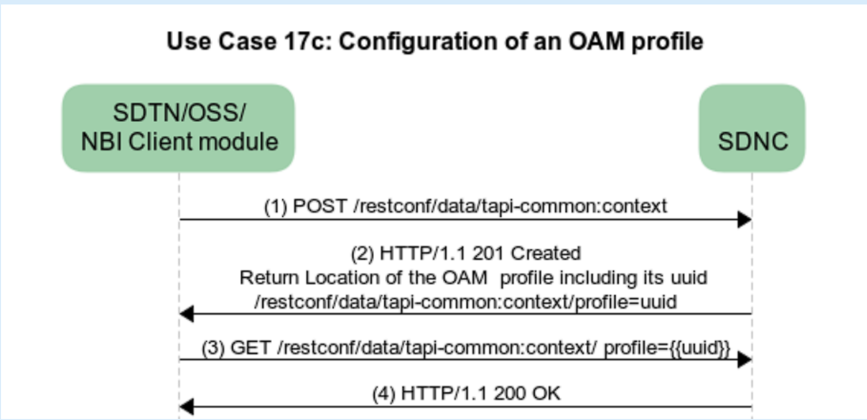


Figure 6-108 UC-17**c**: Creation and subsequent retrieval of an OAM Profile</td></tr>
</table>

The POST body object MUST include the uuid of the profile, as shown:

```
{
     "tapi-common:profile": [{
         "uuid": "6e0abcf9-037c-4b0a-b444-fe37a09f46ed",
         ...
         "tapi-oam:oam-profile" : {
              "pm-data" : [{...
              }]
         }
     }]
}
```

6.8.5.1 Relevant parameters

Table 104: OAM Profile object definition (UC17c)

oam-profile	**/tapi-common:context/tapi-oam:oam-context/oam-profile**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
uuid	As per RFC	RW	M	• Provided by TAPI client
name	Set of name value pairs.	RW	O	• Provided by TAPI client
pm-data	List of instances holding PM data information associated to the OAM profile.	RW	M	• Provided by TAPI client • Minimum number of elements is 1

Table 105: OAM PM Data object definition (UC17c)

pm-data	**/tapi-common:context/tapi-oam:oam-context/oam-profile={uuid}/pm-data={local-id}**
	See Table 83: OAM **PM Data** and Table 84: OAM **PmParameter** definition

6.8.6 Use case 17d: Provisioning of an OAM Job

Number	**17d**
Name	**Provisioning of an OAM Job**
Technologies involved	DIGITAL_OTN/PHOTONIC_MEDIA
Process/Areas Involved	OAM
Brief description	The UC17d targets the provisioning of an OAM Job.
Layers involved	DIGITAL_OTN/PHOTONIC_MEDIA
Type	OAM

Description & Workflow

The workflow relies on the client sending a POST message to the OAM context requesting the creation of an OAM job resource. The request includes the job uuid. The job may be bound to either: i) a previously created OAM service point, ii) a CEP, or a iii) Connectivity service.

The job MAY refer to an existing or previously created OAM Profile (UC17c).

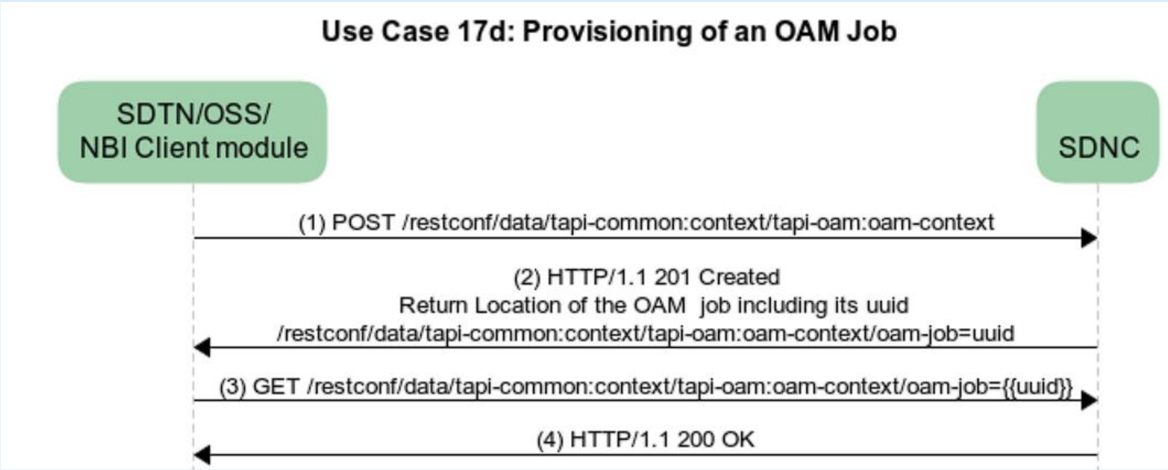


Figure 6-109 UC-17d: Creation and subsequent retrieval of an OAM Job

The POST body object MUST include the uuid of the job, as shown:

```
{
     "tapi-oam:job": [{
         "uuid": "6e0abcf9-037c-4b0a-b444-fe37a09f46ed",
         "oam-job-type" : ...
     }]
}
```

6.8.6.1 17d.1: OAM Loopback

Table 106: **OAM Job** object definition for OAM loopback

oam-job	**/tapi-common:context/tapi-oam:context/oam-job**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
oam-job-type	OAM_JOB_TYPE_LOOPBACK_FACILITY, OAM_JOB_TYPE_LOOPBACK_TERMINAL,	RW	M	• The type of the job when it was created.
connection-end-point	List of CEP references, used to associate the job to such CEP instances.	RW	C	• OAM Loopback applies to a CEP(s)
schedule	Time range, i.e., { "start-time": date-and-time "end-time": date-and-time }	RW	O	• Provided by *tapi-server*. Defines the period where this job is active. If this is not specified, the schedule corresponds to the job object lifetime.
uuid	As per RFC4122	RW	M	• The uuid may be allocated by the server if the creation of the job is the result of NCM provisioning (UC 17b.1, 17b.2)
name	OAM job list of name value pairs.	RW	O	• Provided by *tapi-server*

6.8.6.2 17d.2: Photonic Media Optical Power (draft)

Table 107: **OAM Job** object definition for optical power (complements UC17a)

oam-job	**/tapi-common:context/tapi-oam:context/oam-job**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
oam-job-type	OAM_JOB_TYPE_OPTICAL_POWER	RW	M	• The type of the job when it was created.
connection-end-point	List of CEP references, used to associate the job to such CEP instances.	RW	C	
profile	Reference to a profile (augmented with OAM capabilities) that contains the metric(s) and threshold(s) data for this job.	RW	C	• ***profile and pm-data are exclusive.*** • *A job is either created referring to an existing OAM profile OR with a list of PM data with the PM parameters for the job.* • *profile* is the reference to the OAM profile if a profile was used when creating the job (either directly or via an embedded OAM service) • *pm-data* contains a list of PM Data, Each PM data in turn a list of parameters threshold configuration. To be added (TAPI 2.4.1): • PM_OPTICAL_POWER • PM_PARAM_NAME_OPTICAL_POWER
pm-data	List of {PM Data}	RW	C	
schedule	Time range, i.e., { "start-time": date-and-time "end-time": date-and-time }	RW	O	• Provided by *tapi-server*. Defines the period where this job is active. If this is not specified, the schedule corresponds to the job object lifetime.
uuid	As per RFC4122	RW	M	• The uuid may be allocated by the server if the creation of the job is the result of NCM provisioning (UC 17b.1, 17b.2)
name	OAM job list of name value pairs.	RW	O	• Provided by *tapi-server*

6.8.7 Use case 17e: TCM Provisioning for ODU

Number	**17e**
Name	OAM Service TCM Provisioning
Technologies involved	ODU
Process/Areas Involved	OAM
Brief description	This UC addresses the TCM provisioning for ODU with the independent model. The ODU Connectivity Service has been previously established. This UC assumes that a dedicated OAM Service is provisioning referring to one or more CEPs. The CEP may be either a CEP of the top-level connection or any intermediate CEP.
Layers involved	ODU
Type	OAM
Description & Workflow	This UC involves: 1) The provisioning of the OAM service with one or more OAM Service Point (s) that refer to one or more existing CEP(s). For each OAM Service Point the client specifies whether *is-mip* and the tapi-digital-otn:otn-oam-mip-service-point or the tapi-digital-otn:otn-oam-mep-service-point accordingly. 2) After the successful provisioning of the OAM service, the server instantiates one MEG with its MEPs and MIPs (see UC17a) 3) The Server adds the reference to the corresponding MEP or MIP (within the MEG scope) in the tapi-oam:mep or tapi-oam:mip to the OAM Service Point accordingly (read-only containers) (see UC17a) 4) The client MAY retrieve the CEP(s) and consequently obtain a list to the relevant MEP/MIPs. This UC does not preclude the creation of additional OAM jobs and/or profiles.

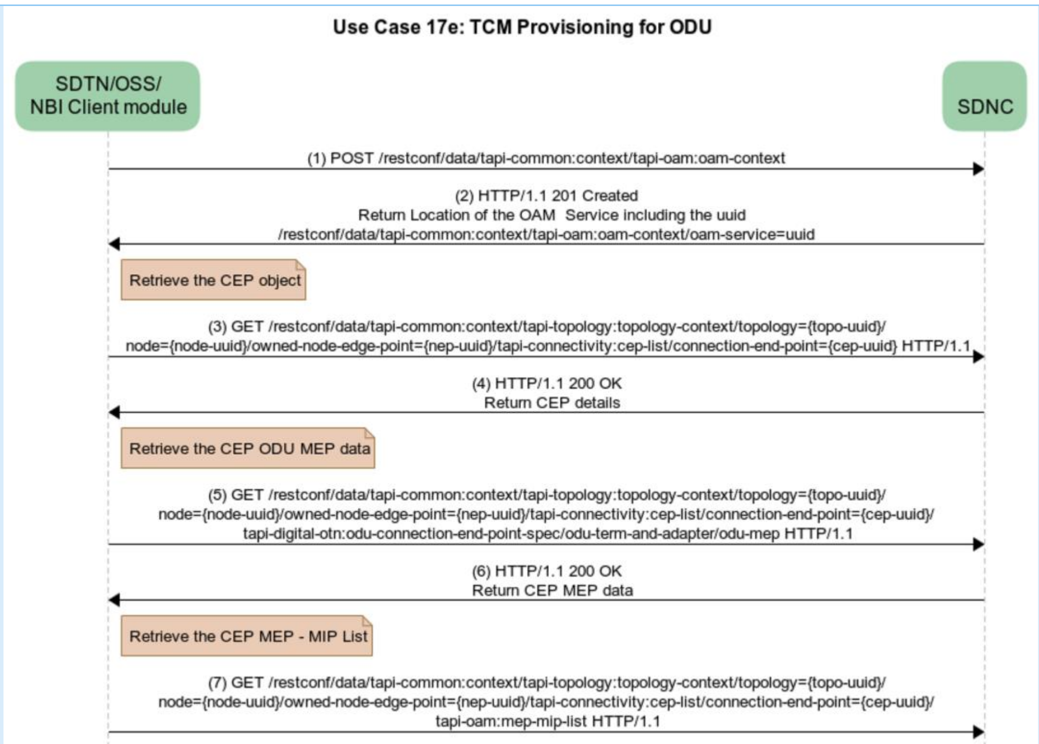


Figure 6-110 UC-17e: TCM Provisioning for ODU

6.8.7.1 Relevant parameters

Table 108: OAM Service object definition

OamService	**/tapi-common:context/tapi-oam:context/oam-service (see Table 86)**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**

Table 109: OamServicePoint object definition

OamServicePoint	**/tapi-common:context/tapi-oam:context/oam-service/oam-service-point (see Table 87)**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**

Table 110: Connection-end-point (**CEP**) object definition (UC17e)

connection-end-point	**/tapi-common:context/tapi-topology:topology-context/topology/node/owned-node-edge-point/tapi-connectivity:cep-list/connection-end-point**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
See UC1.0				
tapi-digital-otn:odu-connection-end-point-spec/odu-term-and-adapter/odu-mep/		RO	M	Provided by *tapi-server*

tapi-digital-otn:odu-connection-end-point-spec/odu-term-and-adapter/odu-mep/otn-oam-common		RO	M	Provided by *tapi-server*
tapi-digital-otn:odu-connection-end-point-spec/odu-term-and-adapter/odu-mep/odu-mep-status		RO	M	Provided by *tapi-server*

7 References

[RFC 8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC 6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC 7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC 7895] Bierman, A., Bjorklund, M., and K. Watsen, "YANG Module Library", RFC 7895, DOI 10.17487/RFC7895, June 2016, <https://www.rfc-editor.org/info/rfc7895>.

[RFC 8525] Bierman, A., et al, "YANG Library", RFC 8525, DOI 10.17487/RFC8525, March 2019, <https://www.rfc-editor.org/info/rfc8525>.

[OpenAPI] OpenAPI Specification Version 3.0.2, <https://swagger.io/specification/>

[CompDocs] TAPI RIA Associated Documents

https://wiki.opennetworking.org/display/OTCC/TAPI+RIA+Associated+Documents

- Alarm and TCA list file "TAPI_Alarm_TCA_List".
- Notification and Streaming Sequence file "TAPI_Notification_Streaming_Sequence".

[RFC 6455] Fette, I. and A. Melnikov, "The WebSocket Protocol", RFC 6455, DOI 10.17487/RFC6455, December 2011, <https://www.rfc-editor.org/info/rfc6455>.

[W3C.REC-SSE] Hickson, I., "Server-Sent Events", World Wide Web Consortium Recommendation REC-eventsource-20150203, February 2015 Considerations <http://www.w3.org/TR/2015/REC-eventsource-20150203>.

[ONF TR-527] Functional Requirements for Transport API, June 10, 2016, ONF TR-527, https://wiki.opennetworking.org/display/OTCC/TAPI+Documentation

[ONF TR-512] https://opennetworking.org/wp-content/uploads/2021/11/TR-512_v1.5_OnfCoreIm-info.zip

[ONF TR-548] TAPI v2.4.0 Reference Implementation Agreement – Streaming (TR-548 v2.0)

https://wiki.opennetworking.org/display/OTCC/TAPI+Reference+Implementation+Agreements+and+other+Documentation

[ITU-T G.709] ITU-T G.709: Interfaces for the optical transport network, G.709/Y.1331 (06/2020)

[ITU-T G.872] ITU-T G.872: Architecture of optical transport networks, ITU-T G.872 (12/2019)

[ITU-T G.805] ITU-T G.805: Generic functional architecture of transport networks, (03/2000)

[RFC 7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <http://www.rfc-editor.org/info/rfc7951>.

[TMF 814] https://www.tmforum.org/resources/reference/mtnm-r4-5-supporting-documents/

8 Definitions

8.1 Terms defined elsewhere

Forwarding Construct [ONF TR-512]

The ForwardingConstruct (FC) represents enabled constrained potential for forwarding between two or more FcPorts (representing the association of the FC to LTPs) at a particular specific Layer Protocol.

Forwarding Domain [ONF TR-512]

The ForwardingDomain (FD) class models the topological component that represents a forwarding capability that provides the opportunity to enable forwarding (of specific transport characteristic information at one or more protocol layers) between points. The FD object provides the context for and constrains the formation, adjustment and removal of FCs and hence offers the potential to enable forwarding.

Logical Termination Point [ONF TR-512]

The LogicalTerminationPoint (LTP) class encapsulates the termination and adaptation functions of one or more transport layers represented by instances of LayerProtocol. The encapsulated transport layers have a simple fixed 1:1 client-server relationship defined by association end ordering. The structure of LTP supports all transport protocols including analogue, circuit, and packet forms.

8.2 Abbreviations and acronyms

CEP	Connection End Point
CRUD	Create, Read/Retrieve, Update, Delete
CS	Connectivity Service
CSEP	Connectivity Service End Point
DSR	Digital Signal Rate
EMS	Element Management System
FC	Fibre Channel
FC	Forwarding Construct
FD	Forwarding Domain
ILA	InLine Amplifier
INNI	Internal Network-to-Network Interface
JSON	JavaScript Object Notation
LTP	Logical Termination Point
MC	Media Channel
MCA	Media Channel Assembly
MEG	Maintenance Entity Group
MEP	Maintenance Entity Group End Point
NBI	Northbound Interface
NEP	Node Edge Point

NMS	Network Management System
OADM	Optical Add-Drop Multiplexer
OAM	Operations, Administration, and Maintenance
OCH	Optical Channel
ODU	Optical Data Unit
OLP	Optical Line Protection
OLS	Optical Line System
OMS	Optical Multiplex Section
OSS	Operations Support Systems
OTN	Optical Transport Network
OTS	Optical Transmission Section
OTSi	Optical Tributary Signal
OTSiA	Optical Tributary Signal Assembly
OTSiG	Optical Tributary Signal Group
OTSiMC	Optical Tributary Signal Media Channel
OTSiMCA	Optical Tributary Signal Media Channel Assembly
OTU	Optical Transmission Unit
ROADM	Reconfigurable Optical Add-Drop Multiplexer
SDK	Software Development Kit
SDN	Software Defined Networking
STM	Synchronous Transport Module
SIP	Service Interface Point
TAPI or T-API	Transport API Information Model
UML	Unified Modeling Language
UNI	User-Network Interface
URI	Uniform Resource Identifier
UUID	Universally Unique Identifier
WDM	Wavelength Division Multiplexing
XC	Cross-Connection

9 Individuals engaged

9.1 Editors

Ramon Casellas	CTTC
Arturo Mayoral López de Lerma	Meta
Nigel Davis	Ciena
Andrea Mazzini	Nokia

9.2 Contributors

Pedro Amaral	Infinera
Karthik Sethuraman	NEC
Malcolm Betts	ZTE
Jonathan Sadler	Infinera
Kam Lam	FiberHome
Jia Qian	ZTE
Ronald Zabaleta	Telefónica

9.3 Acknowledgements

Sequence diagrams were created using websequencediagrams.com capability

10 Appendix: Changes from versions

10.1 Changes between v1.0 and v1.1

- Several RESTCONF usage enhancements
    - XRD and JRD
    - Clarification on JSON encoded Empty Lists
    - Minor clarification on query filtering filter
- TAPI Streaming integrated as optional (and references to TR-548 added)
- Section on RESTCONF Notification and RESTCONF stream discovery/create/subscription added
- SSE v WebSockets clarified
- State propagation via RESTCONF notification detailed
- TAPI virtual network yang removed
- Standard alarm and TCA added
- Equipment/physical model clarified
- TAPI alarm and TCA (for notification channel) improved
    - Note that TAPI Streaming has a separate definition
- TAPI Streaming identified as an alignment and change mechanism
- Clarification to minimum subset of TAPI RESTCONF Data API table
- Clarification and correction in various requirements
- Correction to the Shelf/Slot/Port numbering strategy
- RESTCONF Responses for Common operations added with error info
- Use Case 0a, 0b and 0c adjusted to use “fields” as opposed to “depth”
- Significant improvements in flow description for UC 0b
- Two methods offered in UC 0c (now including get of all connections in the context)
- Improved tables with parameters for the different TAPI entities.
    - Use relevant parameters for use cases enhanced and corrected
        - Corrections to Mandatory/Optional/Conditional throughout
- Plug ID concept description improved
- OTSiA usage clarified
- UC 4b improved
- Support for new operator uses cases has also been added, such as:
    - Multi-domain OTN interdomain links discovery.
    - Asymmetric DSR Service Provisioning, DSR UNI to OTUk E-NNI grey interface.
    - Subscription to Notification Service for Alarm and Threshold Crossing Alert (TCA) events.
    - Initial draft Path Computation use cases.
    - Notification of Alarm and Threshold Crossing Alert (TCA) events
      (Includes new use cases: 0d, 1g, 1h, 2a, 2b, 2c, 3d, 3e, 3f, 5d, 11a, 11b, 13b, 13c, 16a, 16b)
- Line-by-line review of version 1.0, resulting in better and more detailed explanations, enhanced document structure and overall consistency and readability.
- Incorporates feedback from Interop testing of TAPI 2.1.3, such as the need to supplement RESTCONF related standards specifications to facilitate interoperability.
- The Reference Implementation Agreement has also been supplemented with a spreadsheet specifying over 100 standard Alarms and PM Parameters.

10.2 Changes between v1.1 and v2.0

- Updated UML/YANG - 2.4.0
- Deprecated RPCs have been mainly removed and the intention is to not use RPCs

- TAPI Data API list has been enhanced

- Introduction of Profiles in the tapi-common:context
    - Specification of profiles for transceiver properties, OMS / OTS attributes, ROADM paths, amplification functions and fibers
    - Introduction of OAM profiles

- Reflected new layering considerations
    - OTSiMC extended to the transponder, unifying OTSi and OTSiMC
    - Introduction of DIGITAL_OTN layer protocol name and OTU qualifiers.
    - Unspecified layer qualifier has been deprecated and replaced by explicit OMS OTS_MEDIA qualifiers
    - The PHOTONIC_LAYER_QUALIFIER_{ SMC, OMSA, OTSA, OTS_OMS } layer qualifiers are deprecated. The PHOTONIC_LAYER_QUALIFIER_{ OCH, NMC, OTSi, OTSiA } layer qualifiers are not used (candidates for future deprecation). Usage of OTSiMC which integrates the ITU-T OTSi and MC concepts (as well as the OCH). The PHOTONIC_LAYER_QUALIFIER_{MCA, OTSiMCA} when applied to ROADM-to-ROADM scenarios are left for further study. The PHOTONIC_LAYER_QUALIFIER_{OTSiA, OTSiMCA} when applied to Transceiver-to-Transceiver scenarios are left for further study.
    - Corrections to various layers and qualifiers
    - Layering (OTSiMC extension, OTU, OMS, OTS_MEDIA) has been refined (as noted earlier)
- Network topology descriptions have been improved
- Transitional link is deprecated.
- Service deletion (UC10) has been improved with guidelines on ownership of connections.
- Improved UNI and ENNI considerations in a dedicated section
    - Various UNI models
    - Simplified UNI and ENNI scenarios- ENNI model clarified (which is specifically important for asymmetric scenarios)
- New model (tapi-fm), which includes the consolidation of all fault management capabilities, has been added
- Clarification on Global and Local objects
- Clarification on RESTCONF root tree discovery

- Updated RESTCONF subscription and notification mechanisms
    - RESTCONF notification has been updated
    - RESTCONF stream discovery improved
    - Provided guidelines on notification generation. Additional documentation explaining what notifications are generated
    - Streaming and notifications aligned in tapi-fm
    - Notification mechanism now uses proper object notifications by augmenting with the object
    - TAPI Streaming and TAPI RESTCONF Notification have been aligned to follow a single model of alarms as specified in tapi-fm
    - Added companion document on Notification Sequences. Improved Standard alarms document

- Updated Provisioning Scenarios
    - Addition of per layer protocol constraints (LPC), removing the need for CSEP-based workarounds.
    - Review of all provisioning use cases in view of new layering and the usage of LPC. Add MC provisioning based on ITU-T n and m parameters.
    - Enhancements to the connectivity-service and connection model. Clarified the notion of top-level connection.
    - Adopted a single partitioning hierarchy level between top-level connections and their lower-connections

  - Removed the requirement to list all top connections in a Connectivity Service (for scalability reasons). Implementations are expected to list only the immediate top connection for a Connectivity Service and to rely on the connections' lower connections and the newly introduced server connections lists for connection navigation and mapping
  - Improved and detailed scenarios and drawings of key structures
  - Significant review of SIP / NEP / CEP / CSEP parameters
  - Many examples and provisioning scenarios of how to use the CSEPs and SIPs etc. covering e.g. asymmetric and serial compound link
  - Clarified existing UC (e.g. UC1c, UC1e and UC2a) to clarify OTSiA constraints to DSR/ODU services (no direct OTSiA provision covered)

- New section on optical power considerations
- Clarify Mandatory / Conditional statements in some use cases.
  - Work on Conditional/Mandatory properties where the conditions have been improved significantly and many previously mandatory properties have been clarified as conditional) Note that the R/W complexity has not yet been fully untangled (prevents reuse of tables)

- Introduction of Physical Layer Impairment (PLI) model
  - Effort to align to ongoing IETF CCAMP models as well as previous existing practice (GNPy)
  - Detailed UC12d
  - Extended existing tables to include PLI information
  - Addressed layering complexities, especially when considering regeneration and amplifiers (to be further developed including protection).

- Improvements to the equipment model description and to the equipment model to include physical route and strand joint (to allow for fine grained impairments)
  - Added Use Case on Physical route

- Support of OAM use cases
  - OAM section has been significantly updated (will require some further clarification in 2.4.1)
  - Description of the embedded and independent OAM service provisioning models
  - New OAM use cases such as Provisioning of OAM job and Tandem monitoring.
  - Introduction (as draft state) of OAM uses cases related to Optical Power Monitoring.
  - Simplified Network Connection Monitoring (NCM)

End of Document

