Open Networking Foundation

TAPI Reference Implementation Agreement

TR-547

Version 1.1

ONF Document Type: Technical Recommendation

Disclaimer

THIS SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE.

Any marks and brands contained herein are the property of their respective owners.

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

1 Introduction

1.1 General introduction to the model

This ONF Technical Recommendation (TR) is the Reference Implementation Agreement (RIA) for a TRANSPORTAPI (TAPI) based RESTCONF implementation focused on the v2.1.3 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/v2.1.3

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 [ONF 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 interface is proposed, are conceptually described in Figure 1-1. This reference NBI will be the single interface between Operations Support System (OSS), Orchestrator, (super or parent) Controller, etc.[1] 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 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).

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[2] 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

[1] Any system with a repository that maintains alignment with a view of the underlying system as presented by the controller.
[2] At the time management is automated it simply becomes control as explained by [ONF TR-512].

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.

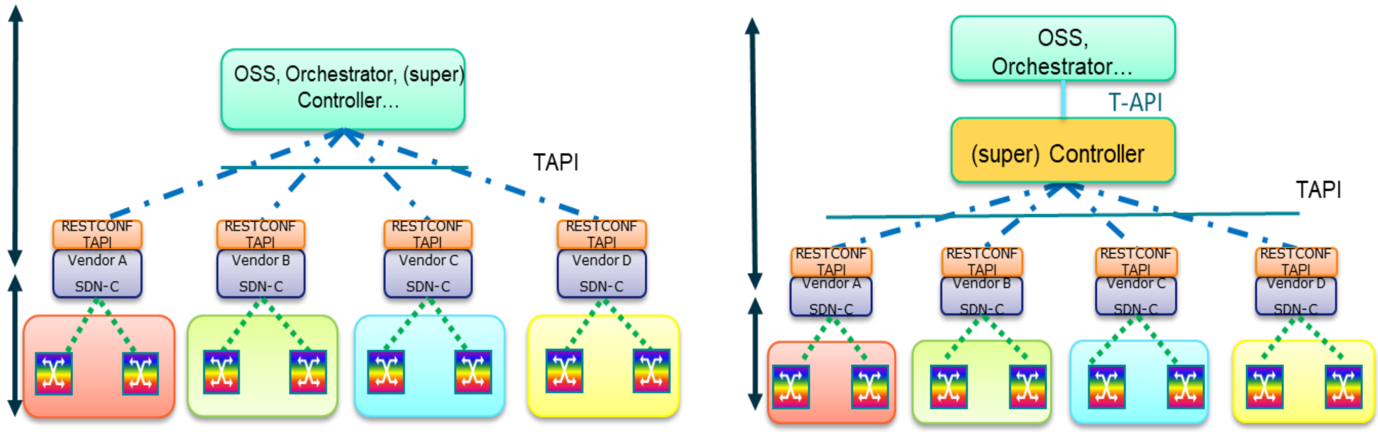


Figure 1-1 Example SDN architecture for WDM/OTN network

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 7895] and [RFC 8525] to allow a client to discover the YANG module conformance information for the server in case the client wants to use it. The mandatory **{+restconf}/yang-library-version** resource is used to clearly identify the version of the YANG library used by the server.

The server MUST implement the "*ietf-yang-library*" module, which MUST identify all the YANG modules used by the server, within the "modules-state/module" and "yang-library/module-set/module" list resource. The module set resource is located at (both implementations are accepted so far):

- **According to [RFC 7895]: {+restconf}/data/ietf-yang-library**:**modules-state**
- **According to [RFC 8525]: {+restconf}/data/ietf-yang-library**:**yang-library**

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 encoded 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 does not have 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
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 TAPI Notifications

The TAPI v2.1.3 tapi-notification data model defines:

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

- One YANG notification statement called *notification* that wraps all notifications generated by the server and includes fields like notification-type, target-object-type, target-object-identifier, target-object-name, event-time-stamp or sequence-number.

[**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]*". However, [RFC 8040] further explicitly states, in Sect 6.3.1, "*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.netconf-json.stream**] this RIA mandates **the support of the NETCONF event stream with JSON encoding format,** as defined in Section 3.2.3 of [RFC5277] and Section 6.2 of [RFC 8040].

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 notification** defined in `tapi-notification.yang` with JSON encoding, as shown. 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)

In addition to the existing "tapi-notification" event stream (see previous section) an implementation MAY support the dynamic creation of TAPI NotificationSubscriptionServices (named notif-subscription in v.2.1.3, further TAPI versions will address the naming convention). 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.

```
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-
subscription-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 6.8. (UC 13a et al).

2.7.1.6 Notification relevant parameters

For TAPI 2.1.3 the defined notification is as follows:

```
notifications:
  +---n notification
     +--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 source-indicator?           source-indicator
     +--ro layer-protocol-name?        tapi-common:layer-protocol-name
     +--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 tca-info
     |  +--ro is-transient?            boolean
     |  +--ro threshold-crossing?      threshold-crossing-type
     |  +--ro threshold-parameter?     string
     |  +--ro threshold-value?         uint64
     |  +--ro perceived-severity?      perceived-tca-severity
     |  +--ro measurement-interval?    tapi-common:date-and-time
     |  +--ro suspect-interval-flag?   boolean
     +--ro alarm-info
     |  +--ro is-transient?         boolean
     |  +--ro perceived-severity?   perceived-severity-type
     |  +--ro probable-cause?       string
     |  +--ro service-affecting?    service-affecting
     +--ro uuid?                       uuid
     +--ro name* [value-name]
        +--ro value-name    string
        +--ro value?        string
```

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

Notification	**/tapi-notification:notification**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
notification-type	One of { OBJECT_CREATION, OBJECT_DELETION, ATTRIBUTE_VALUE_CHANGE, ALARM_EVENT, 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.	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) An additional name value pair MUST be included:
	- "value-name": "**DRI**"
	- "value": Data Resource Identifier or 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: TAPI v.2.1.3 the target-object-name has a min-element = 1 and the list has key "value-name"
				
				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 SHOULD include them.
				
				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_t 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, ODU, PHOTONIC_MEDIA
	}
		RO	O	• Provided by *tapi-server*

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_CHAN • 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. Note that this includes ONLY the object and not the RESTCONF reply for a similar	RO	C	• Provided by *tapi-server* • This field MUST appear ONLY with notification-type OBJECT_CREATION • NOTE: the JSON object must be included as a string. This means that the double quotes MUST be escaped, as described at ecma-

	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"...
	}
	}
				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)*"

tca-info	See UC16b	RO	C	• Provided by *tapi-server* • This field MUST appear ONLY with notification-type THRESHOLD_CROSSING_ALERT
alarm-info	See TAPI Alarms Section	RO	C	• Provided by *tapi-server* • This field MUST appear ONLY with notification-type ALARM_EVENT
name	List of {value-name, value}	RO	O	• Provided by *tapi-server*
uuid	Notification UUID	RO	M	• Provided by *tapi-server*

2.7.1.7 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).

- The creation of an object which is included in one or more list(s) (by reference or by value) MUST trigger: 1) a creation notification for such object followed by 2) an attribute change notification for the referencing object(s).

- 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 object(s) UNLESS such change caused changes in other attributes of the referencing object(s).

- Generally speaking, a *containment relationship* (container/contained) in which a contained 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.

Examples:

- A change in a Connection 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)

- 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 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 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 thare are additional changes.

Because of this:

A change in a local object MUST NOT trigger a notification in the parent global object and, unless there is a dedicated filter for the local object type, the client will not be notified of such change. Note that this RIA use cases focus on the notification of global objects but without excluding the usage of local object types.

Examples:

- If a client subscribes to a CONNECTIVITY_SERVICE object type only, changes in the CSEPs will not be notified.
- If a client subscribes to a CONNECTION object type only, changes in the connection Routes will not be notified.

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 v2.1.3 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).

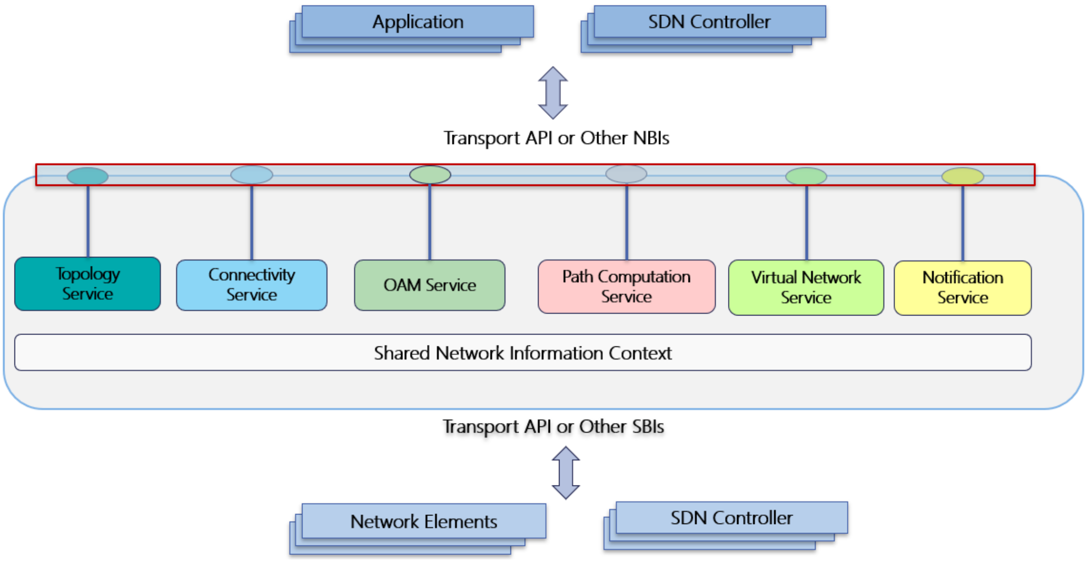


Figure 3-1 Transport API Functional Architecture

The entire list of YANG models composing the TAPI information model can be found in Table 2.

Table 2: TAPI YANG models summary.

Model	Version	Revision (dd/mm/yyyy)
tapi-common.yang	2.1.3	23/04/2020
tapi-connectivity.yang	2.1.3	16/06/2020
tapi-equipment.yang	2.1.3	23/04/2020
tapi-eth.yang	2.1.3	23/04/2020
tapi-dsr.yang	2.1.3	23/04/2020

tapi-streaming.yang	2.1.3	16/06/2020
tapi-notification.yang	2.1.3	16/06/2020
tapi-oam.yang	2.1.3	23/04/2020
tapi-odu.yang	2.1.3	23/04/2020
tapi-photonic-media.yang	2.1.3	16/06/2020
tapi-path-computation.yang	2.1.3	23/04/2020
tapi-topology.yang	2.1.3	23/04/2020

These models can be found at: https://github.com/OpenNetworkingFoundation/TAPI/blob/v2.1.3/YANG

TAPI models are pruned/refactored from the ONF Core Information Model (Core IM) 1.4 [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 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)

This normative document is located at:

https://wiki.opennetworking.org/display/OTCC/TAPI+Alarm+and+Threshold+Crossing+Alert+List

and is a living document (that will continue to be advanced independently from the RIA releases). At the time this RIA was published the version of the Alarm and TCA list was named "TAPI_Alarm_TCA_List_v1.0.0".

The following sections provide a brief overview of the main TAPI concepts which will be used along the rest of the document.

3.2.2 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** exposed to the TAPI client applications representing the available customer-facing access points for requesting network services. This set must allow connectivity-service creation at the following layers:
  - **DSR Layer:** Models a Digital Signal of an unspecified rate, it could be any type of DSR signal such xGigE, FC-x, STM-x or OTU-k which are included as DSR **tapi-common:LAYER_PROTOCOL_QUALIFIER** valid identities in ***tapi-dsr***. This value can be used when the intent is to represent a generic digital layer signal without making any statement on its format or overhead (processing) capabilities.
  - **ODU Layer:** Models the ODU layer as per [ITU-T G.709]
  - **PHOTONIC_MEDIA Layer:** Models the OCH, OTSi, OTSiA, OTSiG, OMS, OTS and Media channels as per [ITU-T G.872].
- A **topology-context** which includes one or more top-level **Topology** objects which are:
  - Dynamic representations of the L2-L0 network based on stateful synchronization of the SDN Controller with the network elements.
  - For more details, please see Section 4**.**
- A **connectivity-context** which includes the list of **Connectivity-Service** and **Connection** objects created within the TAPI Context.
  - For more details, please see Section 5
- 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.
  - For more details, please see Section 3.4

3.2.3 Node and Topology Aspects of Forwarding Domain

The Forwarding-Domain described in the ONF Core IM [ONF TR-512], represents the opportunity to enable forwarding between its edge-points. 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.

3.2.3.1 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.

3.2.3.2 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.

3.2.3.3 Link

The TAPI Link is an abstract representation of the effective adjacency between two or more associated Nodes in a Topology. It is terminated by Node-Edge-Points of the associated Nodes.

3.2.4 Node Edge Point v/s Service End Point v/s Connection End Point

The TAPI Logical-Termination-Point (LTP) – is realized by three different constructs: Node-Edge-Point (NEP), Connection-End-Point (CEP) and Connectivity Service-End-Point (CSEP); they are by design, intended to be a generic, flexible modeling constructs, that can model:

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

So as such, the inherent flexibility, while preserving the underlying pattern, could lead to different model arrangements for same functional configuration. The Logical-Termination-Point (LTP) described in the ONF Core IM [ONF TR-512], represents encapsulation of the addressing, mapping, termination, adaptation, and OAM functions of one or more transport layers (including circuit and packet forms). Where the client – server relationship is fixed 1:1 and immutable, the different layers can be encapsulated in a single LTP instance. Where there is a n:1 relationship between client and server, the layers are split over separate instances of LTP.

Functions that can be associated/disassociated to/from a Connection, such as OAM, protection switching, and performance monitoring are referenced as secondary entities through the associated LTP instance.

Following an illustrative mapping between ITU-T G.800/805 and TAPI constructs is described.

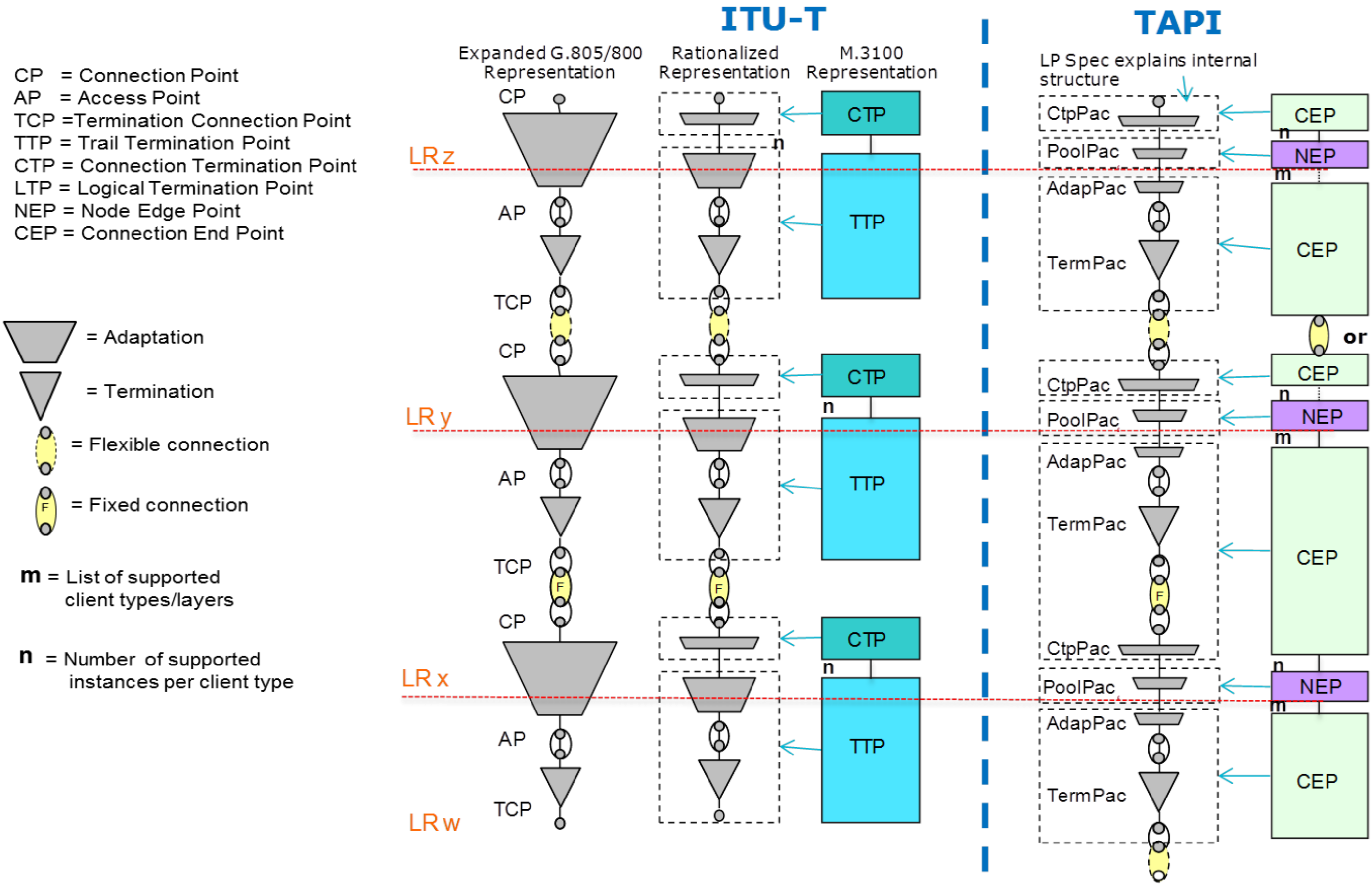


Figure 3-2 TAPI Mapping from ITU-T.

3.2.4.1 Node-Edge-Point (NEP)

The Node-Edge-Point represents the inward network-facing aspects of the edge-port functions that access the forwarding capabilities provided by the Node. Hence it provides an encapsulation of addressing, mapping, termination, adaptation and OAM functions of one or more transport layers (including circuit and packet forms) performed at the entry and exit points of the Node. The Node-Edge-Points have a specific role and directionality with respect to a specific Link.

3.2.4.2 Service-Interface-Point (SIP)

The TAPI Service-Interface-Point represents the outward customer-facing aspects of the edge-port functions that access the forwarding capabilities provided by the Node. Hence it provides a limited, simplified view of interest to external clients (e.g., shared addressing, capacity, resource availability, etc.), that enable the clients to request connectivity without the need to understand the provider network internals. Service-Interface-Point have a mapping relationship (one-to-one, one-to-many, many-to-many) to Node-Edge-Points.

3.2.4.3 Connection-End-Point (CEP)

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.

3.2.5 Equipment model

3.2.5.1 Basic definitions

Macroscopically, 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

A **device** is a logical grouping of Equipment and Access Ports that are closely located and form a support a coherent system of related functions.

A piece of **equipment** is any relevant physical entity (can be either field replaceable or not field replaceable).

Access Port (incl. pins and connectors) is a group of pins that together support a signal group where any one pin removed from the group will prevent all signals of the signal group from flowing successfully. The Access Port may simply reference a single connector. An AP "Belongs-to" a given "device" and thus an "equipment" and can be "Internal" to a device or "External".

A **physical-span** is a physical abstraction. Binds two or more Access Ports (referred to by their device-uuid and access-port-uuid) Is an adjacency between access ports , and such adjacency is supported by a group of strands.

A **strand** represents a continuous long, thin piece of a medium such as glass fiber or copper wire, with current support for strands with 2 ends. A splice can only be between 2 strands, the end of a strand may have a splice, a connector or be hidden

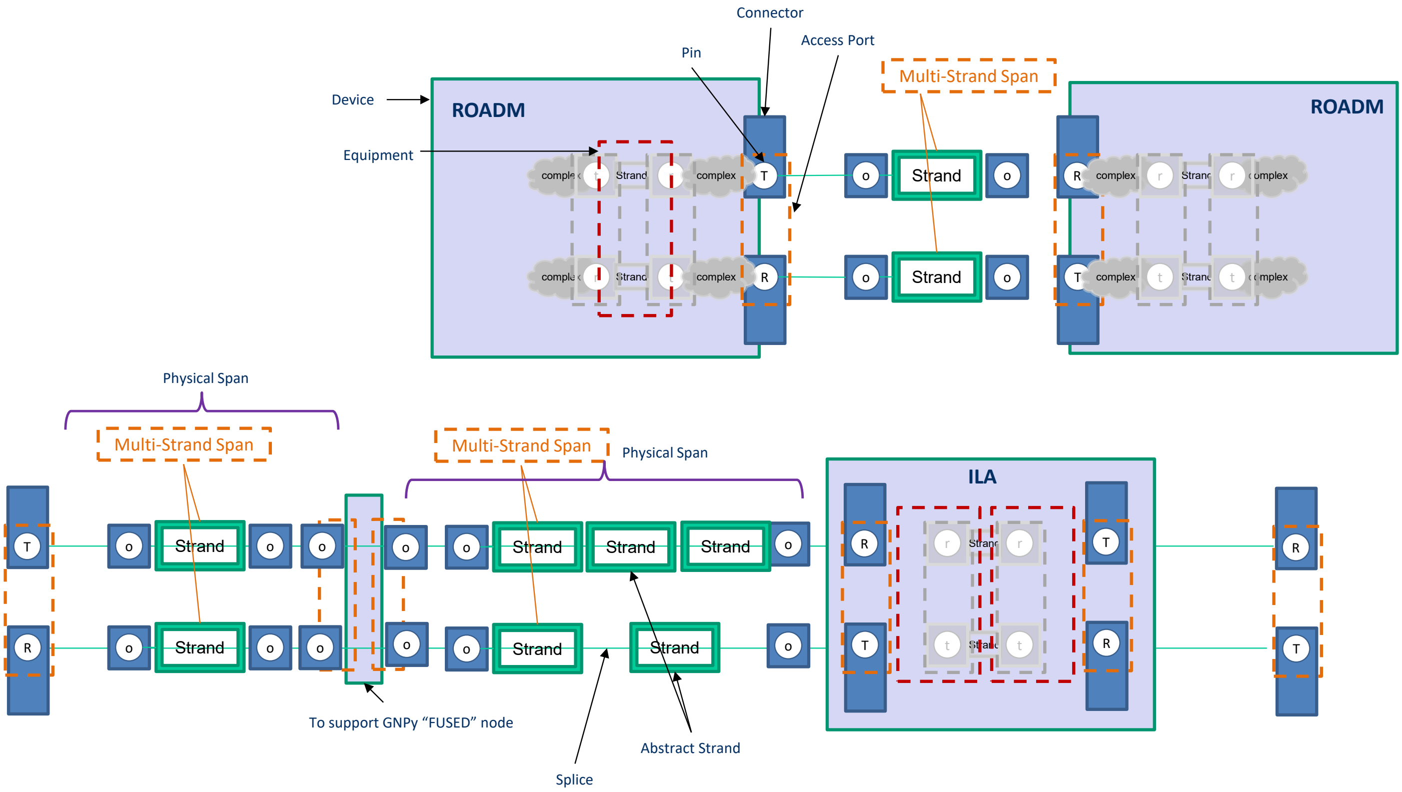


Figure 3-3 Simplified view of the TAPI Equipment / Physical model

3.2.6 Service, Connection and Route

3.2.6.1 Connectivity-Service (CS)

The TAPI Connectivity-Service represents an "intent-like" request for connectivity, between two or more Service-Interface-Points exposed by the Context. As such, Connectivity-Service is a container for connectivity request details and is distinct from the Connection that realizes the request. The requestor of the Connectivity-Service is expected to be able to express their intent using just an "external" Node view of Forwarding-Domain and the advertised Service-Interface-Points and not require knowledge of the "internal" Topology details of the Forwarding-Domain.

3.2.6.2 Connection

The TAPI Connection represents an enabled (provisioned) potential for forwarding (including all circuit and packet forms) between two or more Node-Edge-Points (another realization of LTP described in the ONF Core IM [ONF TR-512]) from the Node aspect of the Forwarding-Domain. A Connection is typically described utilizing the "internal" Topology view of Forwarding-Domain. The TAPI Connection is terminated by Connection-End-Points which are clients of the associated Node-Edge-Points. As such, the Connection is a container for provisioned connectivity that tracks the state of the allocated resources and is distinct from the Connectivity-Service request. 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 Forwarding-Domain (represented as a *tapi-topology:node* object).
- **Top Connections**– are defined as end-to-end connections between Connection-End-Points within the same layer which may span multiple Forwarding-Domains. Top connections are composed by zero or more XCs which belong to the same layer of the Top Connection. The general rules that apply to the creation of Top Connections are introduced in Section 5.1.

3.2.6.3 Route

The TAPI Route represents the route of a Top Connection through the Topology representation. It is described as a list of Connection End-Points (CEPs) cross-connected by the underlying Lower-Connections referenced in the lower-connection list of the Top Connection3.

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.

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

3.2.6.4 Path

The TAPI Path is used to represent the output of path computation APIs and is described by an ordered list of TE Links, either as strict hops (Node-Edge-Points) or as loose hops (Nodes).

3.2.7 TAPI Alarm Framework

TAPI alarms are a type of notifications emitted by the TAPI server (see Section 2.7). This section provides additional considerations and guidelines in view of its use in TAPI 2.1.3 and the future migration towards TAPI 2.3.1+. Further versions of this RIA will address issues related to backwards compatibility as well as new extensions and data models.

An alarm notification includes notification-type: ALARM_EVENT

3 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).

3.2.7.1 Relevant Parameters (alarm-info)

Alarm Event notifications have parameters included inside in the “alarm-info” object. This object MAY be extended by using the name value pairs of the notification object.

```
/tapi-notification:notification:
   +--ro alarm-info
     |  +--ro is-transient?         boolean
     |  +--ro perceived-severity?   perceived-severity-type
     |  +--ro probable-cause?       string
     |  +--ro service-affecting?    service-affecting
```

Figure 3-4 YANG tree snippet for TAPI 2.1.3 alarm-info

The table below defines the relevant parameters that apply to alarm notifications, as well as additional considerations. Note that in view of a future migration to TAPI 2.3.1+, some required information objects are encoded as name value pairs. This will be reviewed in further versions of this specification.

Table 3: TAPI 2.1.3 Alarm information (alarm-info) Relevant Parameters

Attribute	Allowed Values/Format	Mod	Sup	Information Recorded	Notes
probable-cause	String conforming to TAPI Standard Alarm and TCA List (Section 3.2.1)	RO	M	LOS, AIS, LOF, Etc.	TAPI v.2.3.1+ this is mapped into **alarm-name.** If the alarm is related to an undefined Alarm or TCA entry, the probable cause MUST be “NATIVE”
native-alarm-name	string	RO	M	Alternative/Native/Local naming for the alarm event. Usually conveys the name used by the originator device.	TAPI 2.3.1+ data model field. TAPI 2.1.3 use value-name = "native-alarm-name"
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)	TAPI 2.3.1+ data model field. TAPI 2.1.3 use value-name = "native-alarm-info"
alarm-qualifier	String conforming to TAPI Standard Alarm and TCA List (Section 3.2.1) 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.	TAPI 2.3.1+ it is a data model field. In TAPI 2.1.3 use value-name = "alarm-qualifier"

				For example: for a OMS_OTS CEP (target-object) and a LOS alarm, the qualifier provides the actual layer (e.g. OTS).	
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	TAPI 2.3.1+ it is a data model field.
					In TAPI 2.1.3 use
					value-name =
					"alarm-category "

3.2.8 TAPI Threshold Crossing Alerts

TAPI Threshold Crossing Alerts (TCA) are a type of notifications emitted by the TAPI server (see Section 2.7). This section provides additional considerations and guidelines in view of its use in TAPI 2.1.3 and the future migration towards TAPI 2.3.1+. Further versions of this RIA will address issues related to backwards compatibility as well as new extensions and data models.

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. *NOTE: This version of the RIA does not specify how the thresholds are configured.*

[**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

Figure , 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).

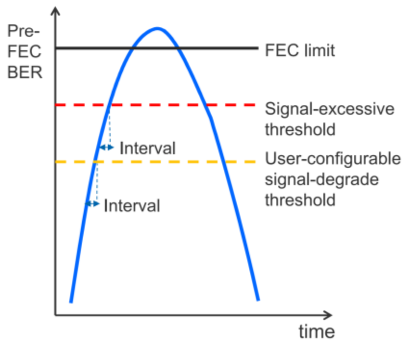


Figure 3-5 FEC function related thresholds

3.2.8.1 Relevant Parameters (tca-info)

TCA Event notifications have parameters included inside in the “tca-info” object. This object MAY be extended by using the name value pairs of the **notification object**.

```
+--ro tca-info
   |  |  |  +--ro is-transient?            boolean
   |  |  |  +--ro threshold-crossing?      threshold-crossing-type
   |  |  |  +--ro threshold-parameter?     string
   |  |  |  +--ro threshold-value?         uint64
   |  |  |  +--ro perceived-severity?      perceived-tca-severity
   |  |  |  +--ro measurement-interval?    tapi-common:date-and-time
   |  |  |  +--ro suspect-interval-flag?   Boolean
```

Figure 3-6 YANG tree snippet for TAPI 2.1.3 tca-info

Table 4: TAPI 2.1.3 Threshold Crossing Alert information (tca-info) Relevant Parameters

Attribute	Allowed Values/Format	Mod	Sup	Information Recorded	Notes
threshold-parameter	String conforming to TAPI Standard	RO	M	Name of the TCA/PM metric	TAPI v.2.3.1+ this is mapped into **threshold-indicator-name.**

	Alarm and TCA List (Section 3.2.1)				If the TCA is related to an undefined TCA entry, the threshold-parameter MUST be "NATIVE"
native-tca-info	(as name)	RO	M		NOTE: Unlike the *alarm-info* object, the *tca-info* object does not contain a field named *native-tca-info*. Consequently, a native identifier for the threshold parameter MUST appear in the NOTIFICATION name pair(s) with "name-value" = "native-tca-info"
threshold-crossing	tapi-alarm: threshold-crossing-type	RO	O	To identify weather the threshold has been crossed above or below.	In TAPI 2.3 this field no longer applies.
is-transient	boolean	RO	M	To indicate if the TCA event is related to a transient condition.	-
threshold-value	uint64 (see note)	RO	C	Indicates the value of the indicator which crossed the threshold.	Note that in RFC7951, sect 6.1 it is specified: *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.* For parameters that are encoded as uint64 (as per the data model), this field MUST be used. For parameters that are encoded as int64 or decimal64, a notification name with "name-value" : "threshold-value" and such lexical representation MUST be used, and this field MUST NOT be used. *Note that the units are implicit by the threshold-parameter for standard TCA* In TAPI 2.3 this corresponds to **threshold-observed-value.**

perceived-severity	One of {
	
	WARNING, CLEAR
	
	}
		RO	C	To indicate the severity.	If the TCA is NOT transient implementations MUST send a notification with perceived-severity "CLEAR" when the threshold is no longer crossed.

tca-qualifier	String conforming to TAPI Standard Alarm and TCA List (Section 3.2.1) 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.	TAPI 2.3.1+ it is a data model field. In TAPI 2.1.3 use value-name = "tca-qualifier"

tca-category	One of {
	
	ALARM_CATEGORY_EQUIPMENT
	
	ALARM_CATEGORY_ENVIRONMENT
	
	ALARM_CATEGORY_CONNECTIVITY
	
	ALARM_CATEGORY_PROCESSING
	
	ALARM_CATEGORY_SECURITY
	
	}
		RO	O	TCA Category	TAPI 2.3.1+ it is a data model field.
					
					In TAPI 2.1.3 use
					
					value-name =
					
					"tca-category "

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 5: Minimum subset required of TAPI RESTCONF Data API Table 5. [*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 5: 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 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	GET	<Optional>

recommendation is to specify the list of requested fields and to perform more specific GET operations.		
/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	Future candidate if node scale issue.
/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	Future candidate if node scale issue.
/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	Future candidate if scale issue
/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}?fields=connection(connection-uuid)	GET	UC 0c

/tapi-common:context/tapi-connectivity:connectivity-context/connection={uuid}	GET	UC 0c
/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

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/RESTCONG 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.

3.4 TAPI Notifications and TAPI Streaming

The current TAPI information model includes two mechanisms for reporting change

- the **tapi-notification@2020-06-16.yang**, which defines the TAPI notifications format but also 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-streaming@2020-06-16.yang**, which defines a specific TAPI streaming mechanism (as described in [ONF TR-548]).

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

4 Topology abstraction model

In this chapter a reference topology abstraction model is described. Due to the need of composing a unified view of the network resources along different TAPI implementations, some guidelines are required in order to constrain the possibilities or interpretations of the current proposed models. The topology model should provide the explicit multi-layer topology representation of the L2-L0 network including OTS, OMS, MC, OTSIMC, OTSi/OTSiA, ODU, DSR layers.

Note.1: The PHOTONIC_LAYER_QUALIFIER_MC and PHOTONIC_LAYER_QUALIFIER_OTSiMC layer-protocol-qualifier values have been newly introduced in TAPI v2.1.3. They are equivalent to previous PHOTONIC_LAYER_QUALIFER_SMC and PHOTONIC_LAYER_QUALIFER_NMC of v2.1.2 respectively.

Note.2: The OTU layer is intentionally simplified in the TAPI model. ODU and OTSiA/OTSi representation is considered enough to cover all our defined use cases.

Based on ONF TAPI 2.1.3 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 <u>*all layers in a single multi-layer topology instance*</u>. The nomenclature **T0 – Multi-layer topology** and **T0** is used interchangeably to reference this topology in the remaining document.

4.1 Model guidelines

To help clarify the following guidelines, 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]
```

To properly describe the topology abstraction model proposed, the following global guidelines are presented:

[TAPI-TOP-MODEL-REQ-1] The network logical abstraction collapses all network layers (DSR, ODU, OTSi/OTSiA and Photonic Media OTSiMC, MC, OMS, OTS), which are represented explicitly into a single topology **(T0 – Multi-layer topology),** 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
```

The following topology abstraction model proposes some degrees of freedom for the TAPI Server topology abstraction model implementation (e.g., L0 Photonic Media layer bidirectional/unidirectional or Transitional Link vs Multi-Layer node approaches). The T0 multilayer topology MUST include the following:

4.1.1 DSR/ODU Layers

[TAPI-TOP-MODEL-REQ-5] **DSR/ODU** multi-layer and multi-rate ***tapi-topology:nodes*** represent the internal mapping between DSR and ODU NEPs (multi-layer) and the multiplexing/de-multiplexing across different ODU rates (multi-rate).

[TAPI-TOP-MODEL-REQ-6] DSR/ODU nodes can represent transponder, muxponders or OTN switching functions.

[TAPI-TOP-MODEL-REQ-7] In such nodes, where the OTSi/OTSiA layers are not included, the NEP objects can have the following allowed combinations:

- For the **layer-protocol-name,** one of **DSR, ODU (**for TAPI 2.1.3) or **DIGITAL_OTN** (for TAPI 2.3+) values.
- For the **supported-cep-layer-protocol-qualifier,** a list of values within
  - identities with base **tapi-dsr:DIGITAL_SIGNAL_TYPE,** for the DSR layer or
  - identities with base **tapi-odu:ODU_TYPE** for the ODU layer

[TAPI-TOP-MODEL-REQ-8] NEPs forwarding capabilities within a node can be **optionally** constrained by using ***node-rule-group/rules/forwarding-rules*** (see snippet)

This feature might be required for some use cases where an external path computation entity is placed on top of the TAPI Server, the details of whether this requirement MUST be fulfilled will be introduced, where appropriate, in the use cases section. The NEPs can be segmented according to the following conditions:

- **Different layer-protocol-qualifier.** In case a multi-layer DSR/ODU node includes NEPs with different **layer-protocol-qualifier** types (i.e., between different DSR_SIGNAL_TYPEs or ODU_TYPE), each group SHALL be segmented with a node-rule-group, including:
  - **forwarding-rule=MAY_FORWARD_ACROSS_GROUP**
- **Not forwarding between same device ports**. In some case it might be relevant to restrict the forwarding between client ports at the same network device (e.g., transponder). In this case ALL NEPs related to client ports at the same device SHALL be segmented with a node-rule-group, including:
  - **forwarding-rule=CANNOT_FORWARD_ACROSS_GROUP**

In case these constrains exist in the network and a service is requested between NEPs which are not potentially connected, **the TAPI Server MUST reject any Connectivity Service request no matter this restriction was exposed or not.**

```
module: tapi-topology
  augment /tapi-common:context:
 +--rw topology-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 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-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
```

4.1.2 ODU/OTSi Layers

Please note, the TAPI Client MUST deal with both ODU-OTSi modelling approaches (transitional link and non-transitional link) as defined next

[TAPI-TOP-MODEL-REQ-9] [**transitional-link**] The ODU <-> OTSI transitions MAY be represented by a **transitional link** between ODU NEPs and OTSi NEPs. A transitional link MUST be represented as a ***tapi-topology:link*** object including a ***tapi-topology:transitioned-layer-protocol-name*** leaf-list attribute, which includes the layers corresponding to the transition, e.g., ["ODU", "PHOTONIC_MEDIA"].

```
module: tapi-topology
  augment /tapi-common:context:
    +--ro topology* [uuid]
       +--ro link* [uuid]
       |  +--ro transitioned-layer-protocol-name* string
```

The transitional-link representation MUST be present at 'Day 0' (for more detailed description please check the examples included in Section 6.1.2.2) between two NEP pools at the ODU and PHOTONIC_MEDIA (OTSi layer qualified) layers. These NEP pools are intended to represent solely the adjacency between the two nodes representing the electrical and optical side of the target optical terminal (i.e., these NEPs are not intended to represent the inverse multiplexing capabilities of the ODU CEPs over OTSi generated NEP resources).

The transitional-link topology model assumes a single NEP pool at the OTSi and ODU layers and each transitional link is an abstract representation of connectivity and connects two NEP pools.

At the OTSi side, each pool MUST include a reference to every NEP representing the TTP facing side of each optical line port exposing OTSi connectivity capacity. These references MUST be implemented by referencing individual OTSi NEPs associated to each physical Optical Line interface using the ***tapi-topology:node/owned-node-edge-point/aggregated-node-edge-point*** attribute within the NEP pool representation.

At the ODU side, it is assumed that the individual NEPs will be created (and "attached" to the ODU NEP pool) dynamically by TAPI server, as a response of the creation of OTSi connections in the lower layer, i.e., the individual ODU NEPs exposing ODU connectivity resources will only be available after the lower layer (OTSi) has been provisioned.

[TAPI-TOP-MODEL-REQ-10] [**non-transitional link**] Alternatively, the ODU <-> OTSi transitions/adaptations MAY be represented within the same DSR/ODU/OTSi multi-layer node, by including the OTSi/OTSiA layers representation of the optical side of the optical terminals.
In such case, some NEPs will have the ***tapi-photonic-media:PHOTONIC_LAYER_QUALIFER_OTSi*** . The ODU<-> OTSi transition(s) MUST be represented as 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, 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
```

Moreover, in case of the multi-layer node implementation, the following requirements set (**OTSi/Photonic Media layers**), related to the OTSI PHOTONIC_MEDIA layer, MUST apply to the multi-layer node construct too.

4.1.3 OTSi/Photonic Media Layers

[TAPI-TOP-MODEL-REQ-11] The OTSi layer qualifier corresponds to the optical side of the optical terminals (transponders/muxponders). It consists of nodes including OTSi client endpoints representing the Trail Termination Points (TTPs) of the OTSi connections and endpoints representing the physical connectivity with ROADM/FOADM add/drop ports. This optical line interfaces representation shall be available at 'Day 0' i.e., after the Optical Terminals commissioning stage and prior to any service deployment over the optical line interfaces. For more detail, please check the examples included in section 6.1.2.2.

[TAPI-TOP-MODEL-REQ-12] 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,** that will have supported CEP qualifier as unspecified as in [TAPI-TOP-MODEL-REQ-15]

[TAPI-TOP-MODEL-REQ-13] Independently of such NEPs directionality, additional **PHOTONIC_MEDIA NEPs** representing potential OTSi connectivity MUST be BIDIRECTIONAL. Such NEPs will have supported qualifier either OTSi or OTSiA, as in [TAPI-TOP-MODEL-REQ-15]

[TAPI-TOP-MODEL-REQ-14] NEPs representing potential OTSi connectivity MUST be present over CEPs qualified as **layer-protocol-qualifier:LAYER_PROTOCOL_QUALIFIER_UNSPECIFIED** to represent the effective forwarding of OTSi connections over the non-qualified PHOTONIC_MEDIA link between Transponder Line Ports and ROADM Add/Drop ports.

[TAPI-TOP-MODEL-REQ-15] NEPs that represent physical connectivity between transponder/muxponder line ports and ROADM/FOADM's add/drop MUST have:

- **layer-protocol-name= PHOTONIC_MEDIA**
- **supported-cep-layer-protocol-qualifier= [LAYER_PROTOCOL_QUALIFIER_UNSPECIFIED]**

NEPs representing potential OTSi connectivity MUST have:

- **layer-protocol-name= PHOTONIC_MEDIA**
- **supported-cep-layer-protocol-qualifier=**

  [LAYER_PROTOCOL_QUALIFIER_ OTSi/OTSiA]

[TAPI-TOP-MODEL-REQ-16] [**Collapsing**] Each OTSi/OTSiA NEP **MAY** include the ***tapi-photonic-media:media-channel-node-edge-point-spec*** to represent the supportable, available, and occupied media channel spectrum pool resources (in addition to the OTSi CEP otsi-termination/selected-spectrum). Similarly, the corresponding LAYER_PROTOCOL_QUALIFIER_UNSPECIFIED CEP **MAY** include the ***tapi-photonic-media:media-channel-connection-end-point-spec*** to represent the encompassing (envelop) spectrum of the OTSiA and the corresponding optical power. Alternatively, this CEP **MAY** include the ***tapi-photonic-media:ots-connection-end-point-spec*** for the same purpose.

Notes:

1) TAPI v2.1.3 does not define a ***tapi-photonic-media:otsi-node-edge-point-spec***. OTSi capability is defined at the SIP level.
2) TAPI v2.1.3 / RIA 1.1 agreed layering does not consider a media-channel NEP/CEP at the transceiver line port (this will be addressed in 2.3 and later).
3) Consequently, this RIA admits the collapsing of spectrum and power resource information in the OTSi NEP and the UNSPECIFIED CEP respectively.

[TAPI-TOP-MODEL-REQ-17] [**OLP Node**] Generally, transponder / muxponder line ports and ROADM / FOADM's add/drop ports are a 1:1 relation. In case Optical Line Protection systems (OLPs) are present, OLP complexity shall be always represented in the Photonic Media layer. Consequently, the OLP appears as a single node, logically part of the Optical Line System (for further description please see Use Case 5b).

4.1.4 Photonic-Media Layers

This sections regards the Photonic Media and Open Line System (OLS) network segments.

[TAPI-TOP-MODEL-REQ-18] The Photonic-Media nodes can represent OLP, ROADM/FOADM and ILA functions. Such nodes are linked by PHOTONIC_MEDIA links (whose NEPs may be augmented with OTS/OMS layers monitoring OAM functions). These node SHALL expose the capability of creating Media Channel connection and connectivity services between its endpoints.

[TAPI-TOP-MODEL-REQ-19] The NEPs layer qualifications allowed **at the Photonic-Media tapi-topology:nodes** are:

- **layer-protocol-name= PHOTONIC_MEDIA**
- **supported-cep-layer-protocol-qualifier = [**

  LAYER_PROTOCOL_QUALIFIER_UNSPECIFIED,

PHOTONIC_LAYER_QUALIFIER_MC,

PHOTONIC_LAYER_QUALIFIER_OTSiMC

]

Note: TAPI 2.3 also adds **PHOTONIC_LAYER_QUALIFIER_OTS_OMS**

[TAPI-TOP-MODEL-REQ-20] 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 occupied by a given OTSi (As a consequence, MC spectrum must be wider or equal than the OTSiMC). Please see Figure 4-1 graphical representation for more clarity.

PHOTONIC_MEDIA/**LAYER_PROTOCOL_QUALIFIER_UNSPECIFIED** NEPs/CEPs at the OLS side of a link between a Transponder Line Port and ROADM Add/Drop port represent the OTS/OMS layers. The MC NEPs MUST be supported on top of such LAYER_PROTOCOL_QUALIFIER_UNSPECIFIED CEPs.

The OTSiMC NEP/CEPs MAY be optionally represented on top of the PHOTONIC_LAYER_QUALIFER_MC CEPs. OTSiMC provide monitoring information of the spectrum of an individual OTSi and its inclusion depends on the HW monitoring capabilities.

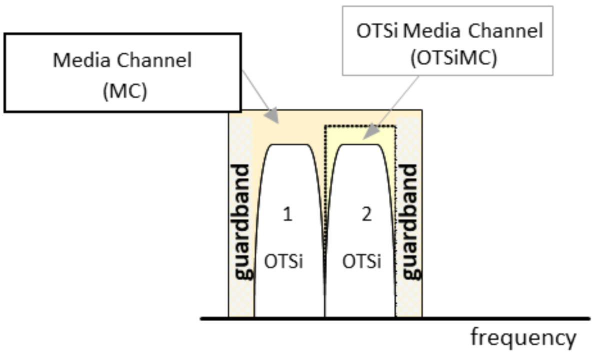


Figure 4-1 Media-channel entities relationship.

[TAPI-TOP-MODEL-REQ-21] 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. **PHOTONIC_LAYER_QUALIFIER_OTSiMC** NEP **MAY** include the ***tapi-photonic-media:media-channel-node-edge-point-spec.***

[TAPI-TOP-MODEL-REQ-22] **tapi-topology:link** objects between **tapi-topology:nodes** which need to be represented in this topology are PHOTONIC_MEDIA (**layer-protocol-name MUST be PHOTONIC_MEDIA** as specified in Table 18) collapsing OTS and OMS layers. **These links MUST be pre-configured in the network in absence of services**.

[TAPI-TOP-MODEL-REQ-23] In case OLP constructs are present for OMS or OTS protection, this should be represented in TAPI by using ***tapi-topology:resilience-type/tapi-topology:protection-type*** link's attribute. Underlying switch control for OMS or OTS protection is out of the scope of this modelling.

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 6: 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 ::= NodeElement [Rack] [Shell [SubShell]] [Slot [SubSlot]] PortId

NodeName ::= "/ne=<nw-ne-name>"

```
/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 real **Network Element (NE)** name configured in the network (i.e., not managed by the SDN-C) and MUST be unique along all exposed interfaces of the network control and management planes (i.e., Network Management Systems (NMSs) or Element Management Systems (EMSs) exposing network information).
- **<r_index>** is the real **Rack index** configured in the network (i.e., not managed by the SDN-C) and MUST be unique along all exposed interfaces of the network control and management planes (i.e., Network Management Systems (NMSs) or Element Management Systems (EMSs) exposing network information).
- **<sh_index>** is the real **Shelf index** configured in the network (i.e., not managed by the SDN-C) and MUST be unique along all exposed interfaces of the network control and management planes (i.e., Network Management Systems (NMSs) or Element Management Systems (EMSs) exposing network information).
- **<s_sh_index>** is the real **Sub-Shelf index** configured in the network (i.e., not managed by the SDN-C) and MUST be unique along all exposed interfaces of the network control and management planes (i.e., Network Management Systems (NMSs) or Element Management Systems (EMSs) exposing network information).
- **<sl_index>** is the real **Slot index** configured in the network (i.e., not managed by the SDN-C) and MUST be unique along all exposed interfaces of the network control and management planes (i.e., Network Management Systems (NMSs) or Element Management Systems (EMSs) exposing network information).
- **<s_sl_index>** is the real **Sub-Slot index** configured in the network (i.e., not managed by the SDN-C) and MUST be unique along all exposed interfaces of the network control and management planes (i.e., Network Management Systems (NMSs) or Element Management Systems (EMSs) exposing network information).
- **<p_index>** is the real **Port index** configured in the network (i.e., not managed by the SDN-C) and MUST be unique along all exposed interfaces of the network control and management planes (i.e., Network Management Systems (NMSs) or Element Management Systems (EMSs) exposing network information).

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 7: 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 6-32 in Section 6.4.2.2 (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"
    }
]
```

4.3 Network scenarios

4.3.1 Scenario 1: ROADM network equipped with OTN matrices

This first scenario represents a three-ROADM network equipped with OTN matrices to perform grooming of Ethernet 1G and 10G client signals into 100G line-side interfaces transmitting OTSi colored optical wavelengths.

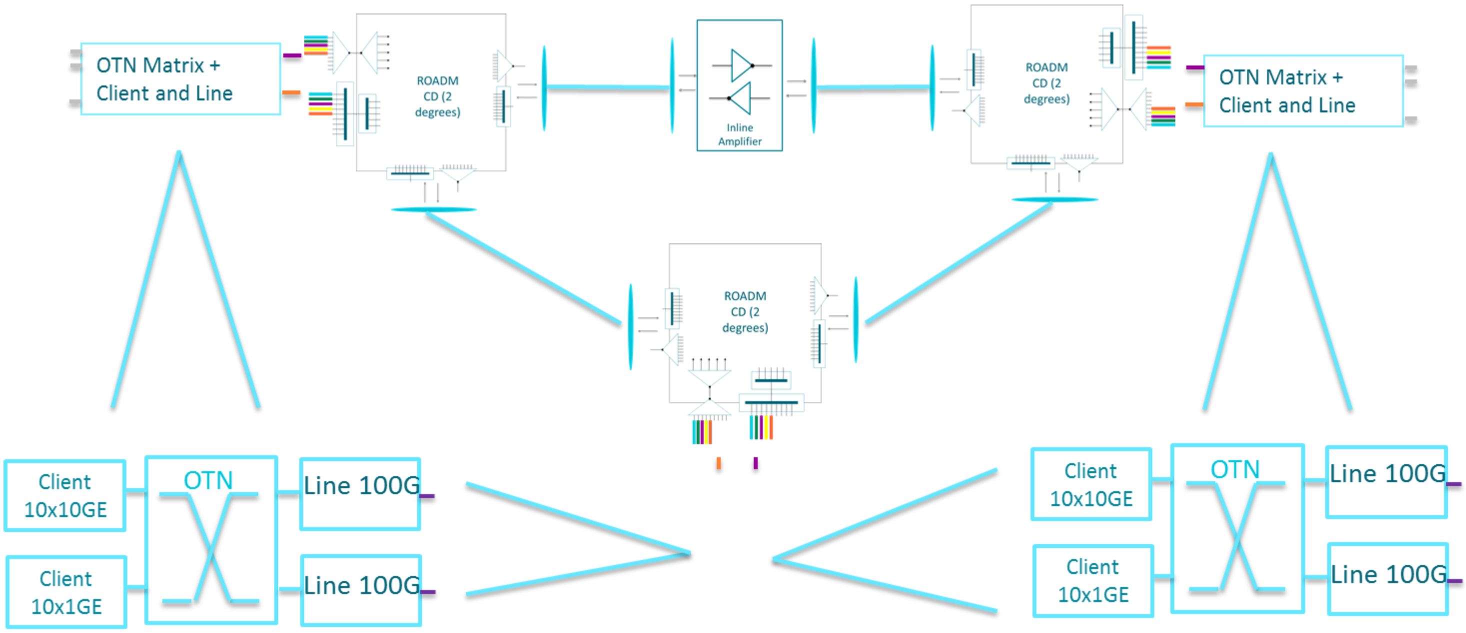


Figure 4-2 NS-1: OTN/WDM Network scenario

4.3.1.1 Model representation (Transitional Link approach)

Please note, in the following representations, the OMS and OTS layers are present purely for information purposes. The specification has stated that these two layers are collapsed from the switching representation perspective. They are represented using LAYER_PROTOCOL_QUALIFIER_UNSPECIFIED layer-protocol-qualifier construct. Please find more details in [TAPI-TOP-MODEL-REQ-22].

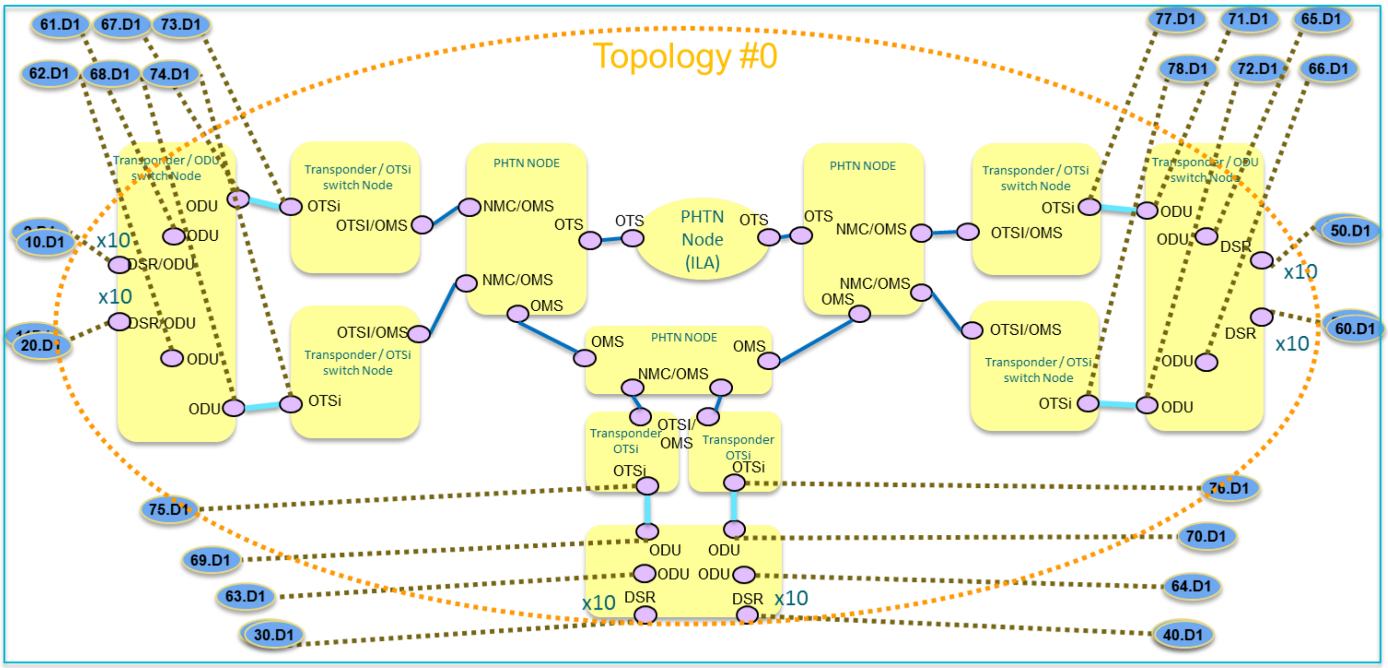


Figure 4-3 NS-1. T0: TAPI Topology Flat Abstraction model, transitional link approach.

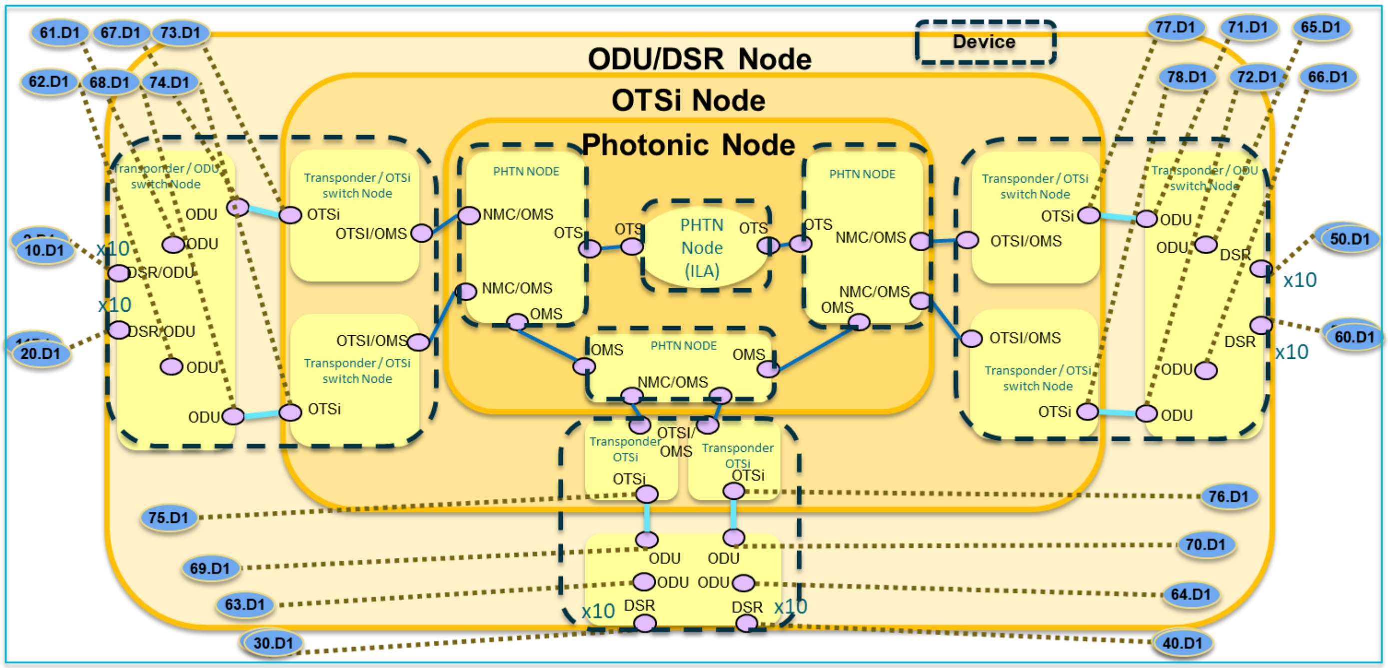


Figure 4-4 NS-1. T0: TAPI Topology Flat Abstraction model, transitional link approach (Device view).

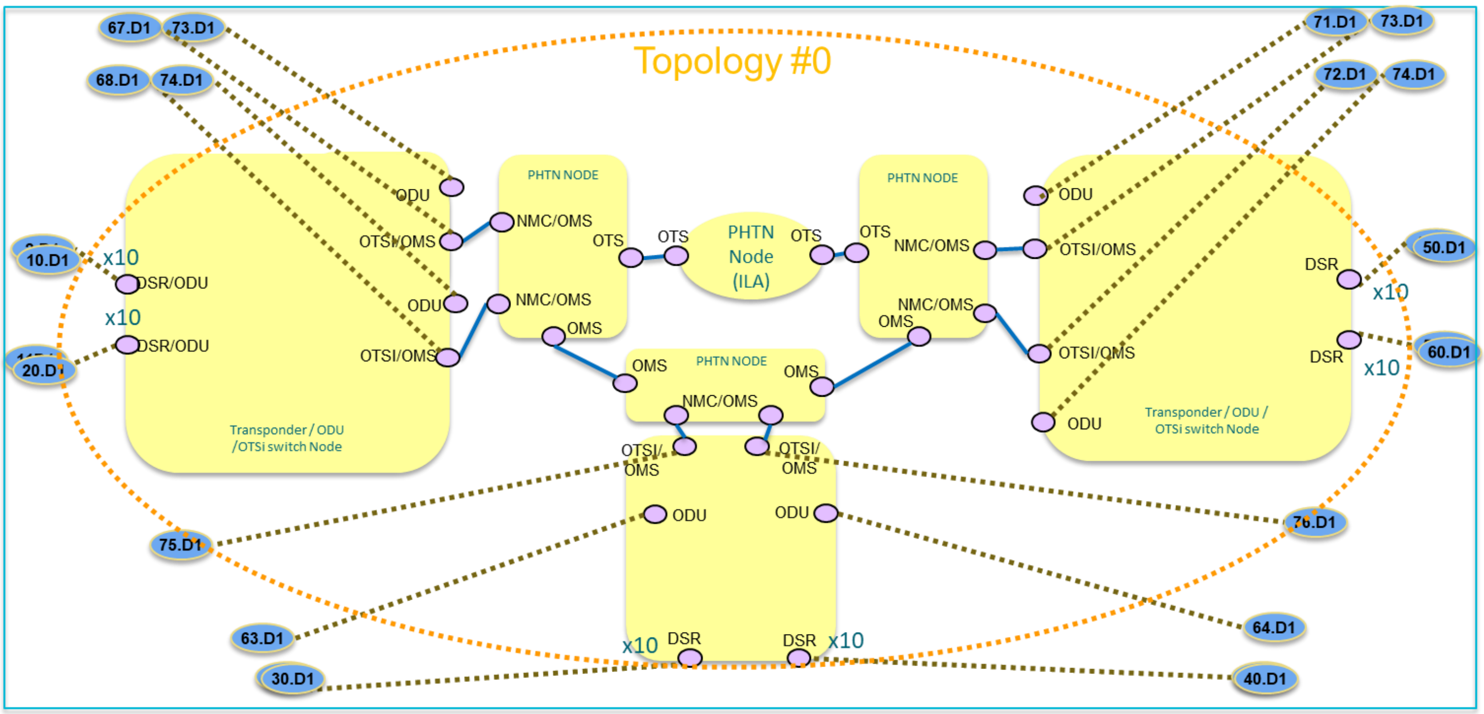


Figure 4-5 NS-1.T0: TAPI Topology Flat Abstraction model multi-layer node approach.

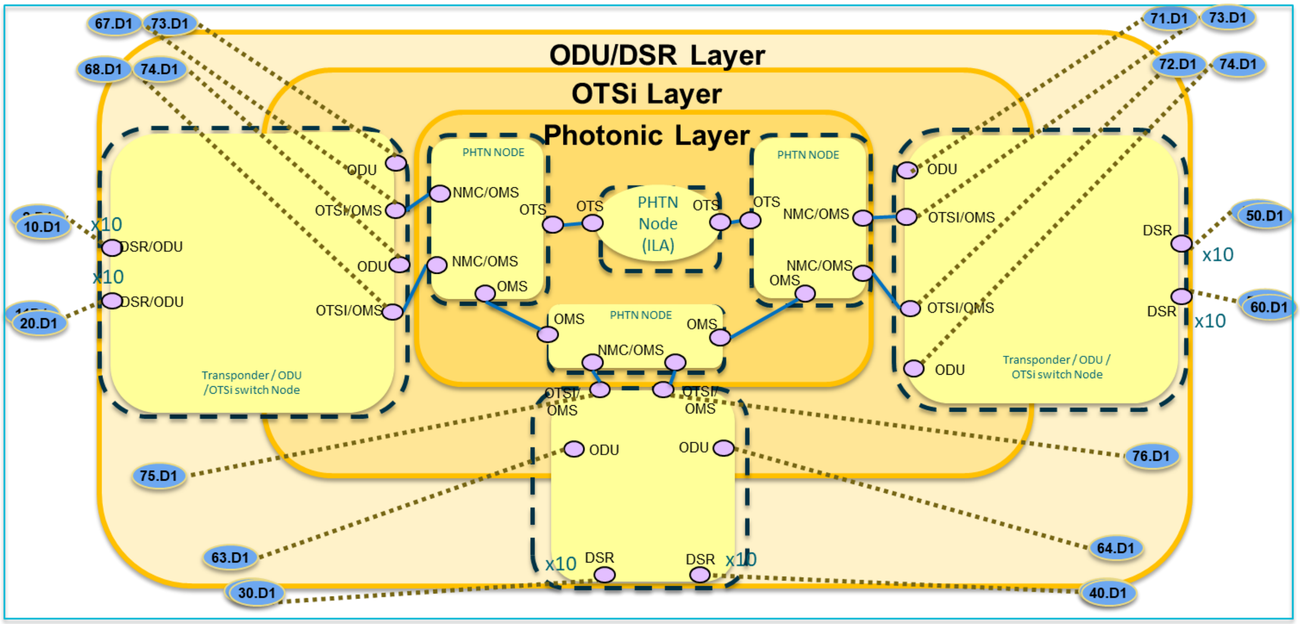


Figure 4-6 NS-1.T0: TAPI Topology Flat Abstraction model multi-layer node approach (Device view).

4.3.2 Scenario 2: Point-to-point DWDM link + Mesh DWDM network

The second scenario consists of two separated networks: one network is a point-to-point DWDM link separated by 2 In-Line-Amplifiers (ILAs) and terminated in Fixed OADM structures consisting of MUX/DEMUX; and the second is a similar three-ROADM network this time equipped with Transponder/Muxponder cards, the two transponders present in the network are connected to a Line Side Optical Line Protection (OLP) structures to provide optical 1+1 protection resilient schema.

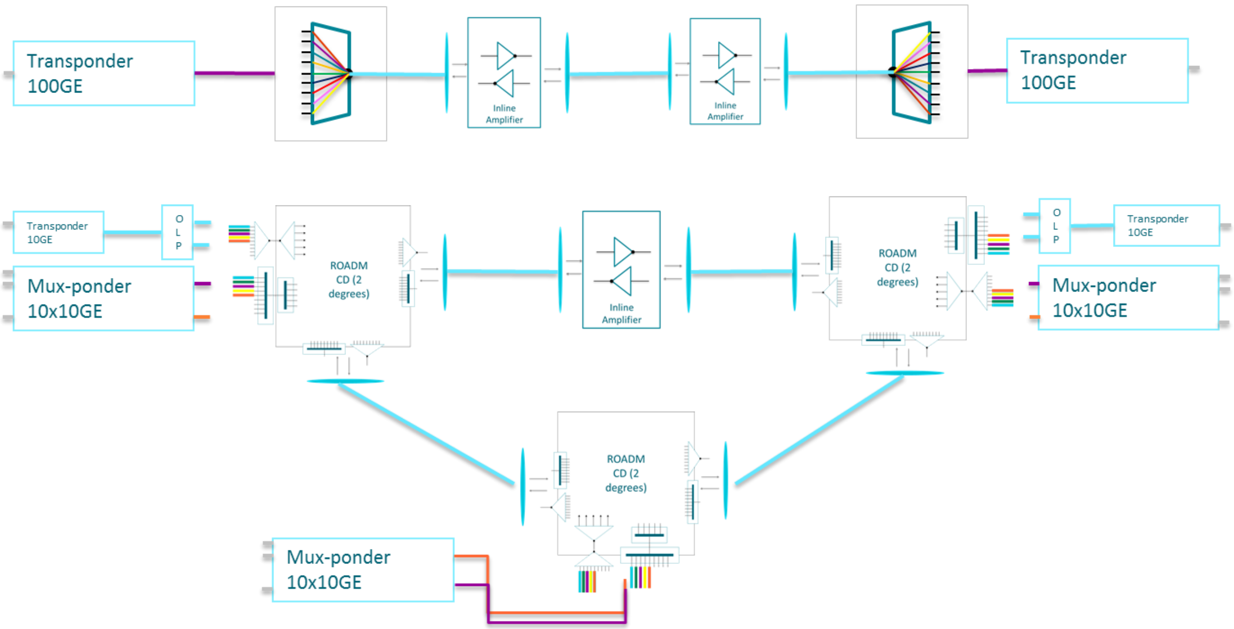


Figure 4-7 NS-2: OTN/WDM Network scenario 2.

4.3.2.1 Model representation

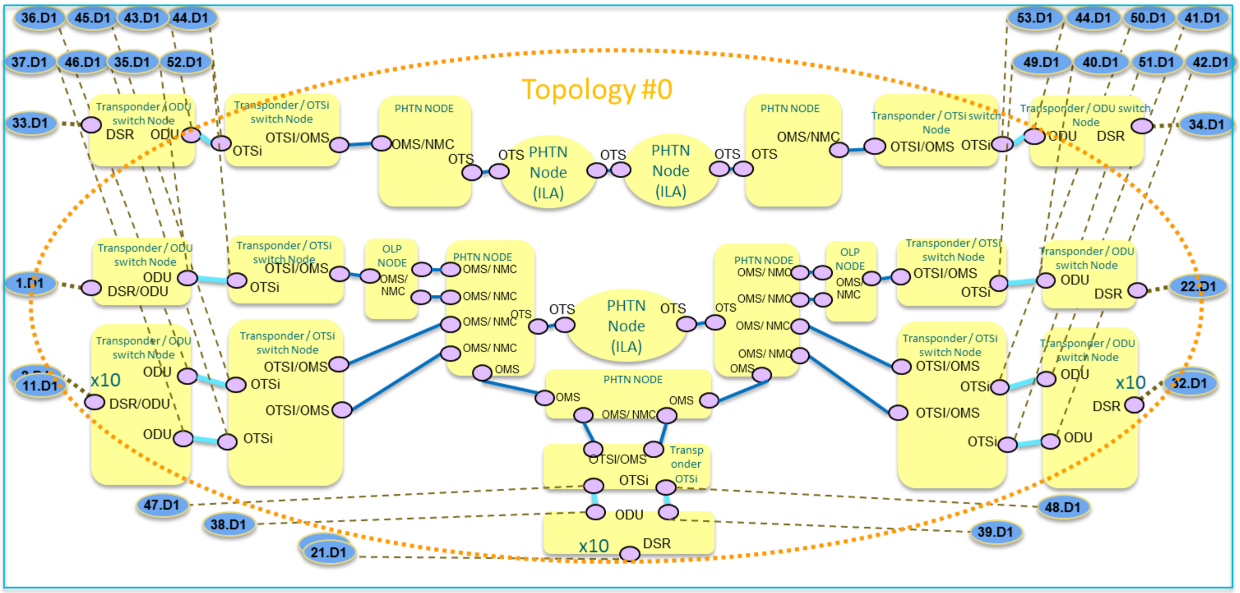


Figure 4-8 NS-2. T0: TAPI Topology Flat Abstraction Transitional Link model.

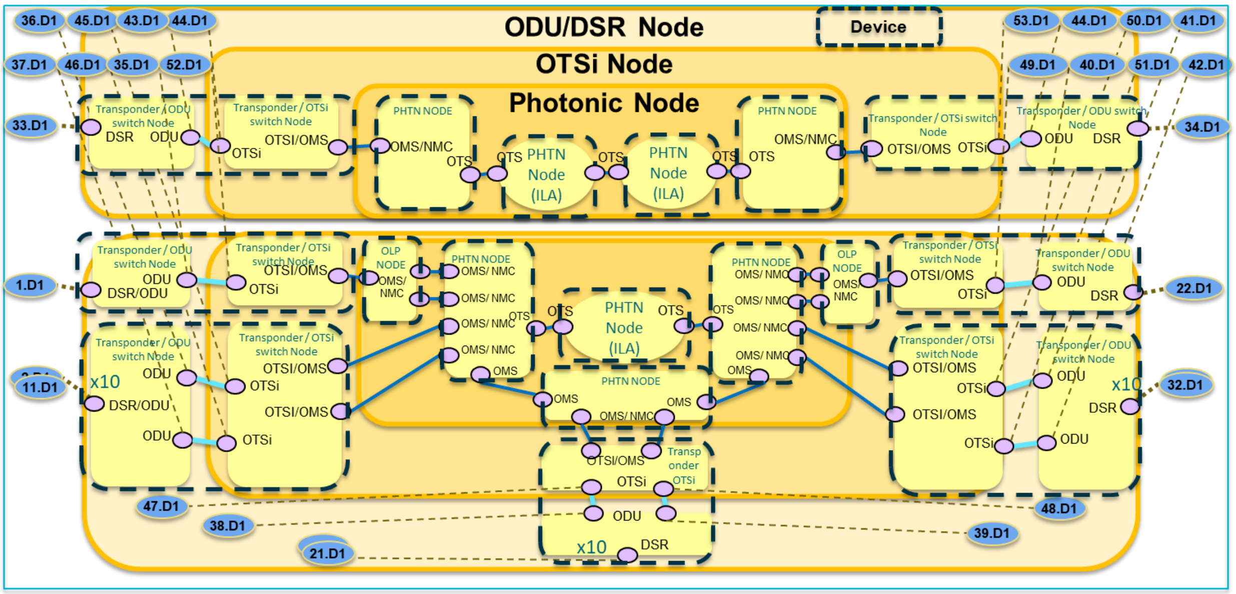


Figure 4-9 NS-2. T0: TAPI Topology Flat Abstraction Transitional Link model (Device view).

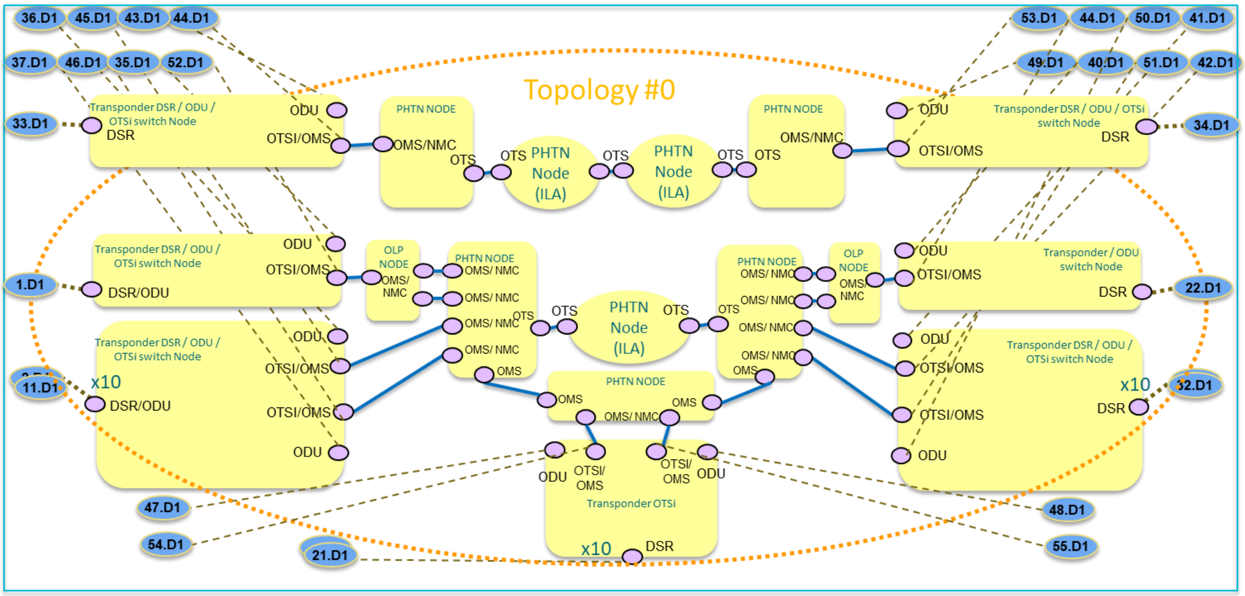


Figure 4-10 NS-2. T0: TAPI Topology Flat Abstraction Multi-Layer Node model.

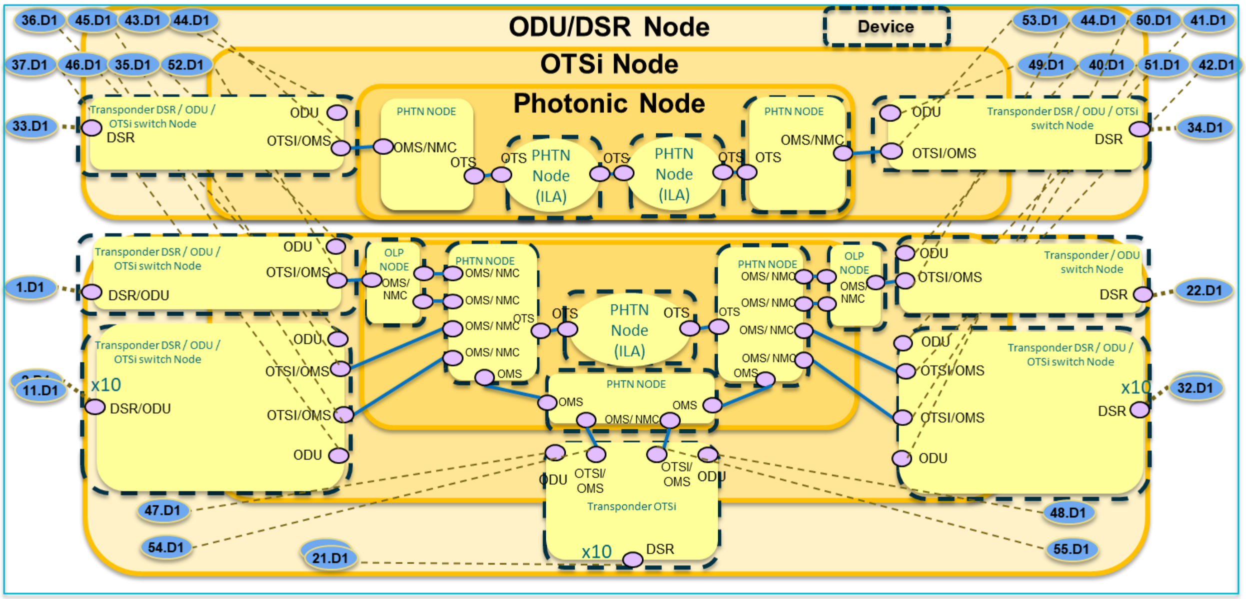


Figure 4-11 NS-2. T0: TAPI Topology Flat Abstraction Multi-Layer Node model (Device view).

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.

The topology absence model is excluded. TAPI model covers connectivity-service without connections but in this reference implementation agreement this option is not covered.

5.1 Model guidelines

The following guidelines MUST be met by all implementations following the current specification.

[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 all proposed network layers:

- **DSR Layer:** Models a Digital Signal of an unspecified format. This value can be used when the intent is to represent a generic digital layer signal without making any statement on its format or overhead (processing) capabilities.
- **ODU Layer:** Models the ODU layer as per described in [ITU-T G.709].
- **PHOTONIC_MEDIA Layer:** Models the OTSi/OTSiA, and Media Channels (OTSiMC, MC) as described in [ITU-T G.872].

[TAPI-CONN-MODEL-REQ-2] [**top-connection-def**] The connectivity model MUST include the concept of **Top Connection(s),** which is defined as the end-to-end connections between CEPs *within the same layer* which may belong to different Forwarding-Domains (TAPI Nodes). They can be either *terminated* ("infrastructure trails") or *non-terminated (*connecting client signals). The OTSi Connection is "always terminated", i.e., there is never an "unterminated" OTSi service, rather it is a MC/OTSiMC service. Please see Section 3.2.6.2 for more complete information.

[TAPI-CONN-MODEL-REQ-3] [**top-connection**] A ***tapi-connectivity:connectivity-service*** MUST, after being successfully provisioned by the TAPI Server, include a reference to all Top Connection ***tapi-connectivity:connection*** objects supporting/composing it 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).

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 (For example, for a DSR CS with intermediate ODU-HO switching and a OTSi regenerator, the logical order would be DSR, ODU-LO, ODU-HO_1, ODU-HO_2, OTSi_1, OTSi_2, OTSiMC, MC...).

2/ Given the unspecified nature of the OMS/OTS layers in TAPI v2.1.3 such connections appear as qualifier UNSPECIFIED. 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). Further versions of this RIA may relax this requirement to only mandate that the only top-most connection shall be listed.

```
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 the ***tapi-connectivity:connection/route*** object containing the list of connection-end-points (CEPs) traversed by the connection at that layer.

```
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**, however the absolute source of truth, to infer the logical order of the CEPs within the Route object by the TAPI client by the knowledge of the Topology information (links) and the NEP and CEP associations, which MUST univocally represent the correct sequence of CEPs for each Top Connection.

[TAPI-CONN-MODEL-REQ-6] [**lower-connection**] The Top Connection MUST include a reference to all the lower connections generated in the 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
```

[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. For example, an ODUk CEP producing a lower order ODUj NEP or an ODUk CEP producing DSR NEP.

```
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

In the proposed model, the TAPI server MUST include a reference to each layer Top Connection within the Connectivity Service's Connection list as stated in [TAPI-CONN-MODEL-REQ-3]. The Connectivity Service routing across different layers MUST be inferred by means of the Top-connections composing/supporting the connectivity-service (referenced within the ***tapi-connectivity:connectivity-service/connection*** list attribute) and their lower-connections, and by the tapi-topology <-> tapi-connectivity model relationships. 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
```

Please note that the previous statement is valid for all layer-protocol and layer-protocol-qualifier transitions but for the HO-ODUk <-> OTSi layer transition which MAY be represented also by a ***tapi-topology:link*** object including a ***tapi-topology:transitioned-layer-protocol-name*** attribute as described in [TAPI-TOP-MODEL-REQ-9]. See a descriptive example in the following figure:

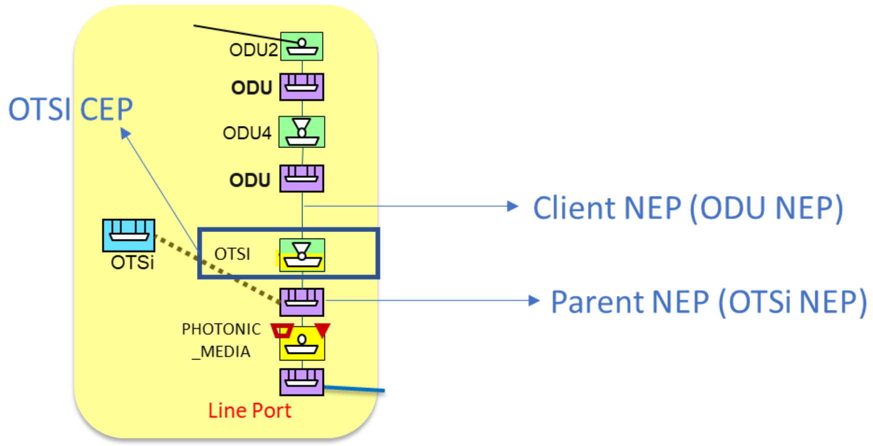


Figure 5-1 Client/parent NEP relations of CEP objects for multi-layer transitions representation.

[TAPI-CONN-MODEL-REQ-9] Additionally, IF a ***tapi-topology:link*** object is generated to represent the adjacency between a pair of NEPs connected by the sequence of cross-connections included as lower-connections of a Top Connection object 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
```

5.1.1.1 Relationship CS and Top-Level Connections

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.

At DSR layer:

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

- Once the DSR Top Connection is operational (***tapi-connectivity:connection/tapi-common:operational-state = ENABLED***), the DSR CS becomes operational too and adds the DSR Top Connection within its connection list.

- The DSR Top Connection MUST include the explicit route referencing CEPs associated to NEPs at the DSR layer (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 its lower-connection list. (Note: that the DSR XC within a transceiver MAY or MAY NOT be present, depending on the representation of the layer flexibility).

At the ODU layer the CS triggers the creation of:

[TAPI-CONN-MODEL-REQ-12] 0-N Top Connections at the ODUj layer qualifiers, which describe(s) the multiplexing of Low Order (LO)-ODU signals into High Order (HO)-ODU signals.

- Once a given ODUj Top Connection is operational (***tapi-connectivity:connection/tapi-common:operational-state = ENABLED***), it MUST be included within the CS connection list.
- Each ODUj Top Connection MUST include the list of ODUj lower connections.
- When each ODUj Top Connection become operational (***tapi-connectivity:connection/tapi-common:operational-state = ENABLED***) in increasing order from lower ODU layer (higher ODU rate) to the DSR layer, the immediately upper layer adjacency is enabled (a higher layer NEP is created “over” the operational CEP) allowing the upper layer Top Connection to be realized.
- When all ODUj Top Connections are operational, a new ***tapi-topology:link*** at the DSR layer (**layer-protocol-name=DSR**) **MAY be optionally** generated between the DSR NEPs produced by the CEPs (Trail Termination Points) of the Topmost Connection with the ODUj Layer rate and referenced by the ***tapi-connectivity: supported-client-link*** attribute***.***

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

- Once a HO-ODUk Top Connection is operational (***tapi-connectivity:connection/tapi-common:operational-state = ENABLED***), it MUST be included within the CS connection list.
- When a HO-ODUk Top Connection becomes operational (***tapi-connectivity:connection/tapi-common:operational-state = ENABLED***) the lower-rate ODU layer adjacency is enabled (client layer NEPs is created “over” the operational CEPs) allowing the upper layer Top Connection to be realized.

[TAPI-CONN-MODEL-REQ-14] (Possibly) multiple HO-ODUk XC Connections, describing the lower partitioning level of HO-ODUk Top Connection. They MUST be included within the Top Connection lower-connection list.

At the PHOTONIC_LAYER_QUALIFER_OTSI layer the CS triggers the creation of:

[TAPI-CONN-MODEL-REQ-15] One or more Top Connection(s) at the OTSi qualifier between the two OTSi CEPs over OTSi NEPs supporting the HO-ODUk CEPs.

- Once the OTSi Top Connection(s) are operational (***tapi-connectivity:connection/tapi-common:operational-state = ENABLED***), they MUST be included within the CS connection list.

- ***If the multi-layer node modelling approach defined in [TAPI-TOP-MODEL-REQ-10] is followed*** (non-transitional links) and related HO-ODUk NEPs are not present in the multi-layer node, then these NEPs MUST be generated to allow the HO-ODUk Top Connection to be realized.
- When the OTSi Top Connection(s) become(s) operational, an ODU ***tapi-topology:link*** between the related HO-ODUk NEPs **MAY be** generated, representing the potential adjacency between these two NEPs at the HO-ODUk layer/rate. If generated, the new link MUST be referenced by the Top Connection, which realizes it, as a ***tapi-connectivity: supported-client-link.***
- OTSi/OTSiA Top Connection(s) MUST include the route referencing CEPs associated to NEPs of PHOTONIC_MEDIA Layer with OTSi supported-layer-qualifier.

[TAPI-CONN-MODEL-REQ-16] Possibly multiple OTSi/OTSiA, XC Connections, describing the lower partitioning level of OTSI Top Connection and MUST be included within its lower-connection list. Note that such OTSi XC may be present, for example, in Transponder or OTSi switch nodes.

- ***If the multi-layer node modelling approach defined in [TAPI-TOP-MODEL-REQ-10] is followed*** (non-transitional links) OTSi XC Connections may not be generated, nor referenced by the lower-connection list. In this case, the Top Connection's route MUST ONLY consist of the CEPs referenced at **tapi-connectivity:connection/connection-end-point** list.

At the Photonic Media Layer, the CS triggers the creation of one or more PHOTONIC_LAYER_QUALIFER_MC and optionally one or more tapi-photonic-media:PHOTONIC_LAYER_QUALIFER_OTSiMC Top Connections as follows:

[TAPI-CONN-MODEL-REQ-17] The MC Top Connection(s) at the Photonic Media layer between the two CEPs supported by the MC NEPs *facing*[4] the OTSi NEPs. At server PHOTONIC_MEDIA layer, the NEPs are linked by a PHOTONIC_MEDIA link.

- Once the MC Top Connection is operational (***tapi-connectivity:connection/tapi-common:operational-state = ENABLED***), it MUST be included within the CS connection list.
- When the MC Top Connection becomes operational (***tapi-connectivity:connection/tapi-common:operational-state = ENABLED***), an OTSi ***tapi-topology:link*** between the OTSi NEPs **MAY be optionally** generated, representing the potential adjacency between these two NEPs at the OTSi layer, allowing ALL OTSi Top Connections supported by the MC to be realized. The new OTSi generated link [5] MUST be referenced by the MC Top Connection within the ***tapi-connectivity:supported-client-link*** *list (note that in this case there is no direct stacking between the OTSi NEP and the MC CEP, as in other scenarios)*
- MC Top Connection(s) MUST include the explicit route referencing CEPs associated to NEPs of PHOTONIC_MEDIA Layer with MC supported-layer-qualifier.

[TAPI-CONN-MODEL-REQ-18] Possibly multiple MC XC Connections, describing the lower partitioning level of MC Top Connection, MUST be included within its lower-connection list.

---

4 Since there is no MC level NEP at the transceiver line port with TAPI v.2.1.3, the OTSi NEP is usually represented at the same level of the MC NEP at the ROADM add/drop port. A link between both NEPs (OTSi-MC) is for further study.

5 Please note this may not be generated in case of an Open Line System disaggregated scenario where OTSi layer NEP facing the MC NEPs are not managed by the TAPI Server.

OTSiMC layer representation, including Top Connections, XCs and CEPs, is only required to represent monitoring capabilities at the filters but not switching (switching is just happening at the lower MC layer).

[TAPI-CONN-MODEL-REQ-19] One or more OTSiMC Top Connections (depending on the number of OTSi composing the OTSiA) at the Photonic Media layer between the two client NEPs generated by MC CEPs logically linked to PHOTONIC_MEDIA Add/Drop ports connected to the OT's line ports.

- Once the each OTSiMC Top Connection is operational (***tapi-connectivity:connection/tapi-common:operational-state = ENABLED***), it MUST be included within the CS connection list.
- Each OTSiMC Top Connection MUST include the explicit route referencing CEPs associated to NEPs of PHOTONIC_MEDIA Layer with OTSiMC supported-layer-qualifier.

[TAPI-CONN-MODEL-REQ-20] Possibly multiple OTSiMC XC Connections, describing the lower partitioning level of OTSiMC Top Connection and MUST be included within its lower-connection list.

5.1.1.2 Example of DSR connectivity-service

The next fragment shows a (quasi) complete generic example of a DSR connectivity-service *including up to the media channel qualifier* of the photonic media layer, to clearly identify the connection hierarchy association described by the previous set of requirements. Given that in TAPI v.2.1.3 OMS/OTS layers are unspecified, this RIA does not mandate the listing of connections below the MC qualifier.

```
{"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"},
                      {"connection-uuid":"ODUj_TOP_1"},
                       ...
                      {"connection-uuid":"ODUj+N_TOP_N"},
                      {"connection-uuid":"HO-ODUk_TOP_1"},
                      {"connection-uuid":"OTSi_TOP_1"},
                      {"connection-uuid":"OTSIMC_TOP_1"},
                      {"connection-uuid":"MC_TOP_1"}
               ]
                   }
        ],

        "connection":[
               {"uuid": "DSR_TOP_1",
                "lower-connection":[
                      {"connection-uuid":"DSR_XC_1"},
                      {"connection-uuid":"DSR_XC_2"}
                ]},
               {"uuid": "ODUj_TOP_1",
```

```
                 "lower-connection":[
                       {"connection-uuid":"ODUj_XC_1"},
                       {"connection-uuid":"ODUj_XC_2"},
                 ]},

                … (repeated for N ODUj layer rates)

                {"uuid": "ODUj_TOP_N",
                 "lower-connection":[
                       {"connection-uuid":"ODUj_XC_1"},
                       {"connection-uuid":"ODUj_XC_2"},
                 ]},
                {"uuid": "HO-ODUk_TOP_1",
                 "lower-connection":[
                       {"connection-uuid":"HO-ODUk_XC_1"},
                       {"connection-uuid":"HO-ODUk_XC_2"}
                 ]},
                {"uuid": "OTSi_TOP_1",
                 "lower-connection":[
                       {"connection-uuid":"OTSi_XC_1"},
                       {"connection-uuid":"OTSi_XC_2"}
                 ]},
                {"uuid": "OTSiMC_TOP_1",
                 "lower-connection":[
                       {"connection-uuid":"OTSiMC_XC_1"},
                       {"connection-uuid":"OTSiMC_XC_2"},
                              …
                       {"connection-uuid":"OTSiMC_XC_N"}
                 ]} ,
                {"uuid": "MC_TOP_1",
                 "lower-connection":[
                       {"connection-uuid":"MC_XC_1"},
                       {"connection-uuid":"MC_XC_2"},
                              …
                       {"connection-uuid":"MC_XC_N"}
                 ]}

        ]}
        }
}
```


[TAPI-CONN-MODEL-REQ-21] The deletion of a CS is rejected if any client CS exists. Further versions of TAPI MAY add an in-use state to the Connectivity Service object (for example, adding additional enumeration values to ***tapi-common:lifecycle-state***).

NOTE: Lifetime issues and interactions between client and server connectivity services will be addressed in a subsequent version of this RIA specification. The current version does not mandate any specific behavior other than the one specified in REQ-23

5.1.2 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:

- 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 among lower-connections, 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
                     -> ...connectivity-context/connection/uuid
         |  |  +--ro switch-control-uuid
                     -> ...connectivity-context/connection/switch-control/uuid
         |  +--ro switch* [local-id]
         |  |  +--ro selected-connection-end-point*
                   [topology-uuid node-uuid node-edge-point-uuid connection-end-point-uuid]
         |  |  |  +--ro connection-end-point-uuid
                        -> ...cep-list/connection-end-point/uuid
         |  |  +--ro selected-route* [connection-uuid route-local-id]
         |  |  |  +--ro connection-uuid    -> ...connectivity-context/connection/uuid
         |  |  |  +--ro route-local-id     -> ...connection/route/local-id
         |  |  +--ro selection-control?  selection-control
         |  |  +--ro selection-reason?   selection-reason
         |  |  +--ro switch-direction?   tapi-common:port-direction
         |  |  +--ro local-id                         string
         |  |  +--ro name* [value-name]
         |  +--ro uuid                                  uuid
         |  +--ro name* [value-name]
         |  +--ro resilience-type
         |  |  +--ro restoration-policy?   restoration-policy
         |  |  +--ro protection-type?      protection-type
         |  +--ro restoration-coordinate-type?          coordinate-type
         |  +--ro restore-priority?                     uint64
         |  +--ro reversion-mode?                       reversion-mode
         |  +--ro wait-to-revert-time?                  uint64
         |  +--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
```

5.1.3 Topology and service constrains for connectivity services

[TAPI-CONN-MODEL-REQ-25] To implement different use cases that imply the constraint of the service path along the network topology, the following parameters of the ***tapi-connectivity:connectivity-service*** object MUST be supported.

```
module: tapi-connectivity
  augment /tapi-common:context:
    +--rw connectivity-context
      +--rw connectivity-service* [uuid]
      |  +--rw coroute-inclusion
      |  |  +--rw connectivity-service-uuid?
      |  +--rw diversity-exclusion* [connectivity-service-uuid]
      |  |  +--rw connectivity-service-uuid
      |  +--rw diversity-policy? diversity-policy
      |  +--rw include-topology*                        tapi-common:uuid
      |  +--rw avoid-topology*                          tapi-common: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
```

5.2 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.

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 8: Responses for GET Operations

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

			supposed to contain the missing attribute	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 9: 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	The request specifies an unacceptable value for one or more parameters. An

			supposed to contain the missing attribute	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-NETCONF entity holds the lock	Access to the requested lock is denied because the lock is currently held by another entity.
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 10: Responses for DELETE Operations

Error-tag	**TAPI *error-app-tag***	**HTTP Response status code**	**Error-info**	**Description**
		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-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		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 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>).

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 and 3.4.
- **Compacted Log Streaming mode** – As described in [ONF 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 (polling mode)

Number	**UC0a**
Name	**Context & Service Interface Points discovery (polling mode)**
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/ODU/PHOTONIC_MEDIA
Type	Discovery

<table>
<tr><td>Description & Workflow</td><td>
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 11 .
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 12 which are marked as Mandatory (M) in the Support (Sup) column.

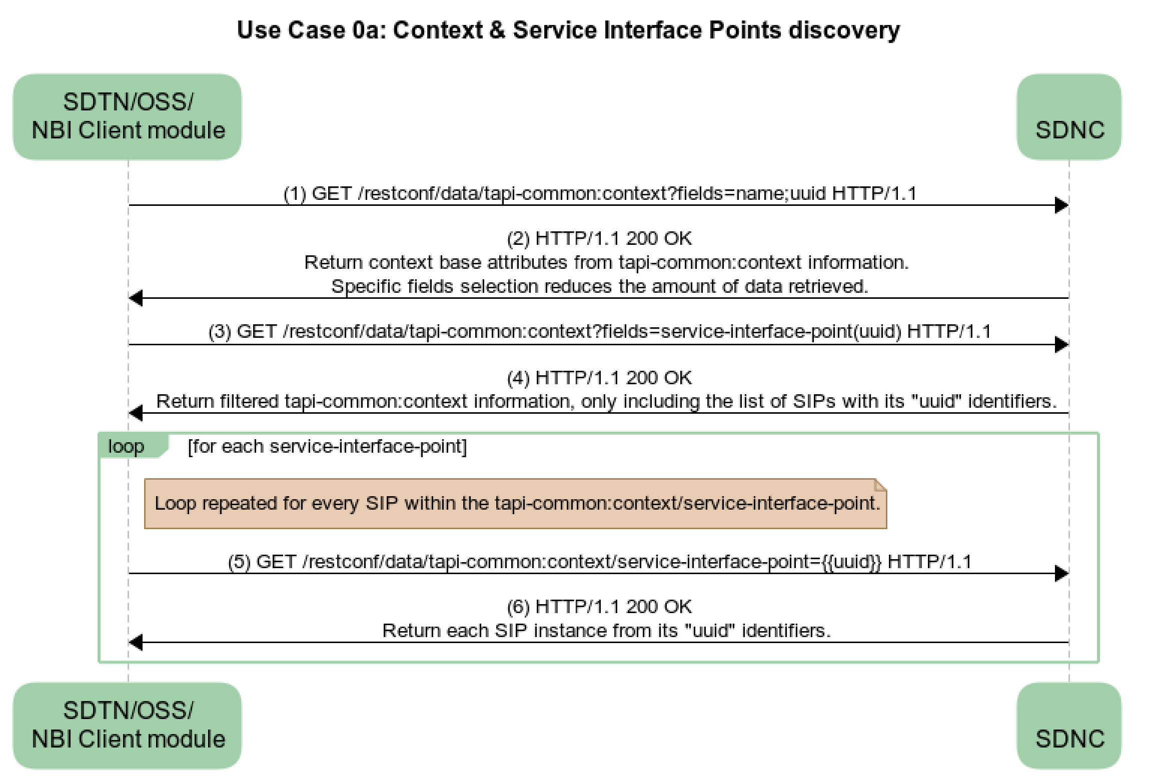


Figure 6-1 UC-0a: Context and Service Interface Point - Workflow.
</td></tr>
</table>

6.1.1.1 Relevant parameters

Table 11: 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:	RO	M	• Provided by *tapi-server*

	"value-name": "CONTEXT_NAME" "value": " [0-9a-zA-Z_]{64} "value-name": "VENDOR_NAME" "value": "[0-9a-zA-Z_]{64}"			• 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
notification-context	• List of {notif-subscription} • List of {notification}	RO	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 in the context. • The list of subscriptions MUST be present IF the user has configured them.
topology-context	• {network-topology-service} • List of {topology}	RO	M	• Provided by *tapi-server*
connectivity-context	• List of {connectivity-service} • List of {connection}	RO	C	• Provided by *tapi-server* Note: see Section 2.4 regarding TAPI lists and presence containers.

Table 12: 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.*

layer-protocol-name	One of the values ["DSR", "ETH", "ODU", or "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.*
supported-layer-protocol-qualifier	List of ["DIGITAL_SIGNAL_TYPE", "ODU_TYPE", "PHOTONIC_LAYER_QUALIFIER"]	RO	C	• Provided by tapi-server • Children identities defined for ["DIGITAL_SIGNAL_TYPE", "ODU_TYPE", "PHOTONIC_LAYER_QUALIFIER"] MUST be supported depending on hardware capabilities.
administrative-state	One of ["UNLOCKED", "LOCKED"]	RW	M	• Initial value provided by *tapi-server* • Subsequent updates 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	M	• Provided by *tapi-server*
total-potential-capacity	"total-size": {value, unit} • "value": "[0-9]{8}", • "unit": One of ["TB", "TBPS", "GB", "GBPS", "MB", "MBPS", "KB", "KBPS", "GHz", "MHz"]	RO	C	• Provided by *tapi-server* • NOTE: theoretical maximum bandwidth you can set up on the SIP. For example, 100 Gb. • NOTE: The use of capacity *objects*, values and units is technology-specific.
available-capacity	"total-size": {value, unit} • "value": "[0-9]{8}", • "unit": One of ["TB", "TBPS", "GB", "GBPS", "MB", "MBPS", "KB", "KBPS", "GHz", "MHz"]	RO	C	• Provided by *tapi-server* • NOTE: The use of capacity *objects*, values and units is technology-specific.
tapi-photonic-media:otsi-service-interface-point-spec/otsi-capability	• List of "supportable-central-frequency-spectrum-band":{ lower/upper-central-frequency, frequency-constraint: {adjustment-granularity, grid-type} }	RO	C	• Provided by *tapi-server* • NOTE: This block of parameters **MUST** augment SIPs of layer PHOTONIC_MEDIA exposing

	○ “central-frequency”: " [0-9]{9}", ○ “adjustment-granularity”:[“UNCONSTRAINED”, “G_3_125GHZ”, “G_6_25GHZ”, “G_12_5GHZ”, “G_25GHZ”, “G_50GHZ”, “G_100GHZ”,] ○ “grid-type”: [“GRIDLESS”, “FLEX”, “CWDM”, “DWDM”] • List of “supportable-application-identifier”:{ application-identifier-type, application-code} ○ “application-identifier-type”:[“PROPRIETARY”, “ITUT_G959_1”, “ITUT_G698_1”, “ITUT_G698_2”, “ITUT_G696_1”, “ITUT_G695”,] ○ “application-code”: " [0-9a-zA-Z_]{64}" • List of “supportable-modulation”:[“RZ“, “NRZ”, “BPSK”, “DPSK”, “QPSK”, “8QAM”, “16QAM”] • “total-power-warn-threshold”: {total-power-lower-warn-threshold-default /min/max, total-power-upper-warn-threshold-default /min/max”} ○ “total-power-*-warn-threshold”: "[0-9].[0-9]{7}"			OTSI/OTSiG service provisioning capabilities. • The lower/upper-**central-frequency** of the laser specified in MHz. Those correspond to the lower and upper central frequencies of the band. • **frequency-constraint** specify the rest of feasible central frequencies within the band. • **Adjustment-granularity** in Gigahertz. As per ITU-T G.694.1, it is used to calculate nominal central frequency". • The **grid-type** specifies the reference set of frequencies used to denote allowed nominal central frequencies that may be used for defining applications. • **NOTE: total-power-warn-threshold** attribute is optional.
tapi-photonic-media:otsi-service-interface-point-spec/power-management-capability	• "supportable-maximum-output-power":{total-power, power-spectral-density} ○ "total-power":"[0-9].[0-9]{64}", ○ "power-spectral-density":"[0-9].[0-9]{64}" • "supportable-minimum-output-power":{total-power, power-spectral-density} ○ "total-power":"[0-9].[0-9]{64}", ○ "power-spectral-density":"[0-9].[0-9]{64}" • "tolerable-maximum-input-power":{total-power, power-spectral-density} ○ "total-power":"[0-9].[0-9]{64}", ○ "power-spectral-density":"[0-9].[0-9]{64}"	RO	C	• Provided by *tapi-server* • NOTE: Even if the data model lists RW, these values are expected to be RO. • NOTE: This block of parameters **MUST** augment SIPs of layer PHOTONIC_MEDIA exposing OTSI/OTSiG service provisioning capabilities.

	• "tolerable-minimum-input-power":{total-power, power-spectral-density} ○ "total-power":"[0-9].[0-9]{64}", ○ "power-spectral-density":"[0-9].[0-9]{64}"			
tapi-photonic-media:media-channel-service-interface-point-spec/mc-pool	• List of "supportable/available/occupied-spectrum":{upper/lower-frequency, frequency-constraint: {adjustment-granularity, grid-type} } ○ "upper/lower-frequency": " [0-9]{9}", ○ "adjustment-granularity":["UNCONSTRAINED", "G_3_125GHZ", "G_6_25GHZ", "G_12_5GHZ", "G_25GHZ", "G_50GHZ", "G_100GHZ",] ○ "grid-type": ["GRIDLESS", "FLEX", "CWDM", "DWDM"]	RO	C	• Provided by *tapi-server* • NOTE: This block of parameters **MUST** augment SIPs of layer PHOTONIC_MEDIA exposing MC service provisioning capabilities. • The **upper/lower-frequency** bound of the media channel spectrum specified in MHz • **Adjustment-granularity** in Gigahertz. As per ITU-T G.694.1, it is used to calculate nominal central frequency". • The **grid-type** specifies the reference set of frequencies used to denote allowed nominal central frequencies that may be used for defining applications.
tapi-photonic-media:media-channel-service-interface-point-spec/power-management-capability	• "supportable-maximum-output-power":{total-power, power-spectral-density} ○ "total-power":"[0-9].[0-9]{64}", ○ "power-spectral-density":"[0-9].[0-9]{64}" • "supportable-minimum-output-power":{total-power, power-spectral-density} ○ "total-power":"[0-9].[0-9]{64}", ○ "power-spectral-density":"[0-9].[0-9]{64}" • "tolerable-maximum-input-power":{total-power, power-spectral-density} ○ "total-power":"[0-9].[0-9]{64}", ○ "power-spectral-density":"[0-9].[0-9]{64}" • "tolerable-minimum-input-power":{total-power, power-spectral-density} ○ "total-power":"[0-9].[0-9]{64}", ○ "power-spectral-density":"[0-9].[0-9]{64}"	RO	C	• Provided by *tapi-server* • NOTE: This block of parameters **MUST** augment SIPs of layer PHOTONIC_MEDIA exposing MC service provisioning capabilities. • NOTE: Even if the data model lists RW, these values are expected to be RO.

6.1.2 Use Case 0b: Topology discovery (synchronous mode)

Number	**UC0b**
Name	**Topology discovery (synchronous mode)**
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. Note, 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***
	    o 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 13 (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:
	    o The parameters of the node as defined in Table 14
	    o The list of node-edge-points (owned-node-edge-point) of the node
	    o The parameters for each node-edge-point as defined in Table 15
	    o The list of connection-end-points of a node-edge-point
	    o 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 18. 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 13 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*.

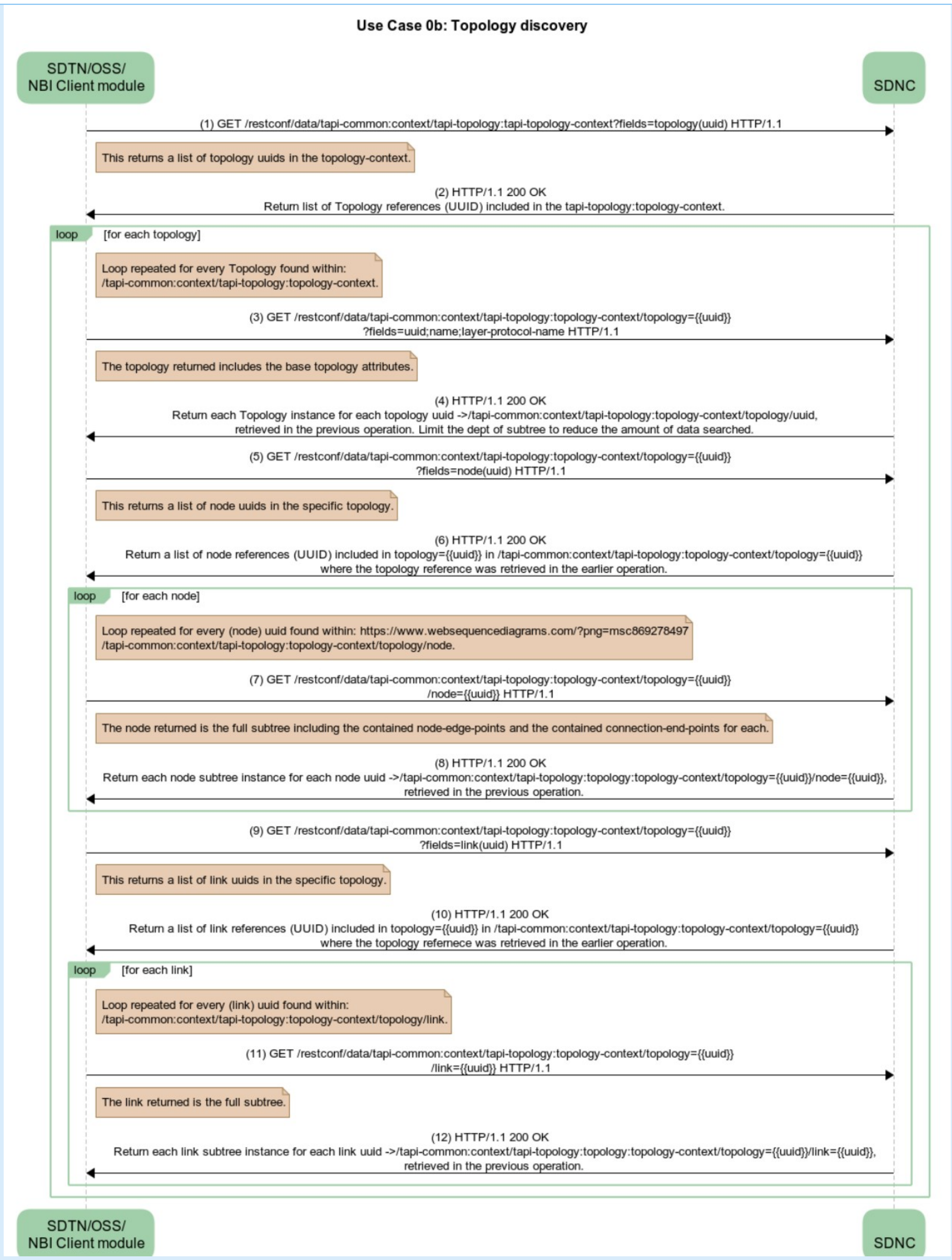


Figure 6-2 UC-0b: Topology discovery - Workflow.

6.1.2.1 Relevant parameters

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

Table 13: 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	List including elements from ["DSR", "ODU", "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 14: 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
layer-protocol-name	List including elements from ["DSR", "ODU", "PHOTONIC_MEDIA"]	RO	M	• Provided by ***tapi-server***
administrative-state	One of ["UNLOCKED", "LOCKED"]	RO	M	• Provided by ***tapi-server***
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	M	• Provided by ***tapi-server***

total-potential-capacity	"total-size": {value: unit} • "value": "[0-9]{8}", • "unit": see note	RO	O	• Provided by ***tapi-server*** • Unit depends on layer
available-capacity	"total-size": {value: unit} • "value": "[0-9]{8}", • "unit": see note	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***
encapsulated-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 15
node-rule-group	List of {***node-rule-group***}	RO	O	• Provided by ***tapi-server*** • See Table 16

Table 15: 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	List of {value-name: value} • "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", "ODU", "PHOTONIC_MEDIA"]	RO	M	• Provided by *tapi-server*
supported-cep-layer-protocol-qualifier	List of qualifiers	RO	M	• Provided by *tapi-server* • All children identities defined for ["DIGITAL_SIGNAL_TYPE", "ODU_TYPE",

				"PHOTONIC_LAYER_QUALIFIER"] MUST be supported when applicable.
administrative-state	One of ["UNLOCKED", "LOCKED"]	RO	M	• Provided by *tapi-server* NOTE: even if the model specifies
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	M	• Provided by *tapi-server*
termination-state	One of ["LP_CAN_NEVER_TERMINATE", "LT_NOT_TERMINATED", "TERMINATED_SERVER_TO_CLIENT_FLOW", "TERMINATED_CLIENT_TO_SERVER_FLOW", "TERMINATED_BIDIRECTIONAL", "LT_PERMENANTLY_TERMINATED", "TERMINATION_STATE_UNKNOWN"]	RO	O	• Provided by *tapi-server*
termination-direction	One of ["BIDIRECTIONAL", "SINK", "SOURCE"]	RO	M	• Provided by *tapi-server*
link-port-direction	One of ["BIDIRECTIONAL", "INPUT","OUTPUT"]	RO	M	• Provided by *tapi-server*
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, unit} • "value": "[0-9]{8}", • "unit": ["TB", "TBPS", "GB", "GBPS", "MB", "MBPS", "KB", "KBPS"]	RO	M	• Provided by *tapi-server* NOTE: Concrete layers may not have a directly mapped value in bps (e.g., PHOTONIC_MEDIA and GHz).
available-capacity	"total-size": {value: unit} • "value": "[0-9]{8}", • "unit": ["TB", "TBPS", "GB", "GBPS", "MB", "MBPS", "KB", "KBPS"]	RO	M	• Provided by *tapi-server* NOTE: Concrete layers may not have a directly mapped value in bps (e.g., PHOTONIC_MEDIA and GHz).

aggregated-node-edge-point	List of { *node-edge-point-ref* }	RO	O	• Provided by *tapi-server*
mapped-service-interface-point	List of {***"/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.
cep-list/connection-end-point	List of { *connection-end-point* }	RO	M	• Provided by *tapi-server*
tapi-photonic-media: media-channel-node-edge-point-spec	"mc-pool": {supportable/available/occupied-spectrum} List of "supportable/available/occupied-spectrum":{upper/lower-frequency, frequency-constraint: {adjustment-granularity, grid-type} } o "upper/lower-frequency": " [0-9]{9}", o "adjustment-granularity":["UNCONSTRAINED", "G_3_125GHZ", "G_6_25GHZ", "G_12_5GHZ", "G_25GHZ", "G_50GHZ", "G_100GHZ",] o "grid-type": ["GRIDLESS", "FLEX", "CWDM", "DWDM"]	RO	C	• Provided by *tapi-server* • This block of parameters MUST augment PHOTONIC_MEDIA NEPs exposing media channel characteristics (MC qualifier) and MAY augment an OTSi NEP as per 0 • The **upper/lower-frequency** bounds of the media channel spectrum specified in MHz • **Adjustment-granularity** in Gigahertz. As per ITU-T G.694.1, it is used to calculate nominal central frequency". • The **grid-type** specifies the reference set of frequencies used to denote allowed nominal central frequencies that may be used for defining applications.
tapi-odu:odu-node-edge-point-spec	"odu-pool": {client capacity, max-client-instances, max-client-size}	RO	O	• Provided by *tapi-server*
NOTE: In TAPI v2.3 tapi-photonic-media:otsi-node-edge-point-spec is to be defined				

Table 16: 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	List of {value-name: value} • "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 17

Table 17: 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	"[0-9a-zA-Z_]{32}"	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"	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 18: 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	List of {value-name: value} • "value-name": "LINK_NAME" "value": " [0-9a-zA-Z_]{64}"	RO	M	• Provided by *tapi-server*
layer-protocol-name	List of elements from ["DSR", "ODU", "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. Note there is a dedicated transitioned-layer-protocol-name for transitional links. In such case, the current RIA convention is to list the client LAYER only
administrative-state	One of ["UNLOCKED", "LOCKED"]	RO	M	• Provided by *tapi-server*
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	M	• Provided by *tapi-server*
direction	One of ["BIDIRECTIONAL", "UNIDIRECTIONAL"]	RO	M	• Provided by *tapi-server* NOTE: TAPI v2.1.3 description "Is applicable to simple Links where all LinkEnds are BIDIRECTIONAL (the Link will be BIDIRECTIONAL) or UNIDIRECTIONAL (the Link will be UNIDIRECTIONAL)" is **deprecated**
total-potential-capacity	"total-size": {value: unit} • "value": "[0-9]{8}", • "unit": ["TB", "TBPS", "GB", "GBPS", "MB", "MBPS", "KB", "KBPS"]	RO	C	• Provided by *tapi-server* • NOTE: Concrete layers may not have a directly mapped value in bps (e.g., PHOTONIC_MEDIA and GHz).
available-capacity	"total-size": {value: unit} • "value": "[0-9]{8}", • "unit": ["TB", "TBPS", "GB", "GBPS", "MB", "MBPS", "KB", "KBPS"]	RO	C	• Provided by *tapi-server* • NOTE: Concrete layers may not have a directly mapped value in bps (e.g., PHOTONIC_MEDIA and GHz).
transitioned-layer-protocol-name	List of {**layer-protocol-name}**	RO	C	• Provided by *tapi-server* • Only applicable to transitional-links.
resilience-type	Includes restoration-policy protection-type	RO	C	• Provided by *tapi-server* • Depends on the use case
cost-characteristic	List of {cost-name: cost-value} • "cost-name": "HOP_COUNT" "cost-value": "[0-9]{8}"	RO	O	
latency-characteristic	List of { traffic-property-name: fixed-latency-characteristic }	RO	O	

	• "traffic-property-name": "FIXED_LATENCY" "fixed-latency-characteristic": "[0-9]{8}"			
risk-characteristic	List of {risk-characteristic-name: risk-identifier-list} • "risk-characteristic-name": ["SRLG", "SRNG"] "risk-identifier-list": List of "[0-9a-zA-Z_]{64}"	RO	O	
node-edge-point	List of {"*node-edge-point-ref*"}	RO	M	• Provided by *tapi-server*

6.1.2.2 Expected results

In this section we introduce the detail TAPI-Topology modelling expected at "Day 0" (i.e., after the commissioning stage of the network devices into the SDN-C, but before any service is configured).

6.1.2.2.1 UC0b - Example 0: TAPI topology representation at "day 0" following Transitional Link modelling approach

The detailed description of the assumptions assumed by the Reference Implementation to compose the Topology at Day '0' based on a Transitional-Link model, are described in [TAPI-TOP-MODEL-REQ-9].

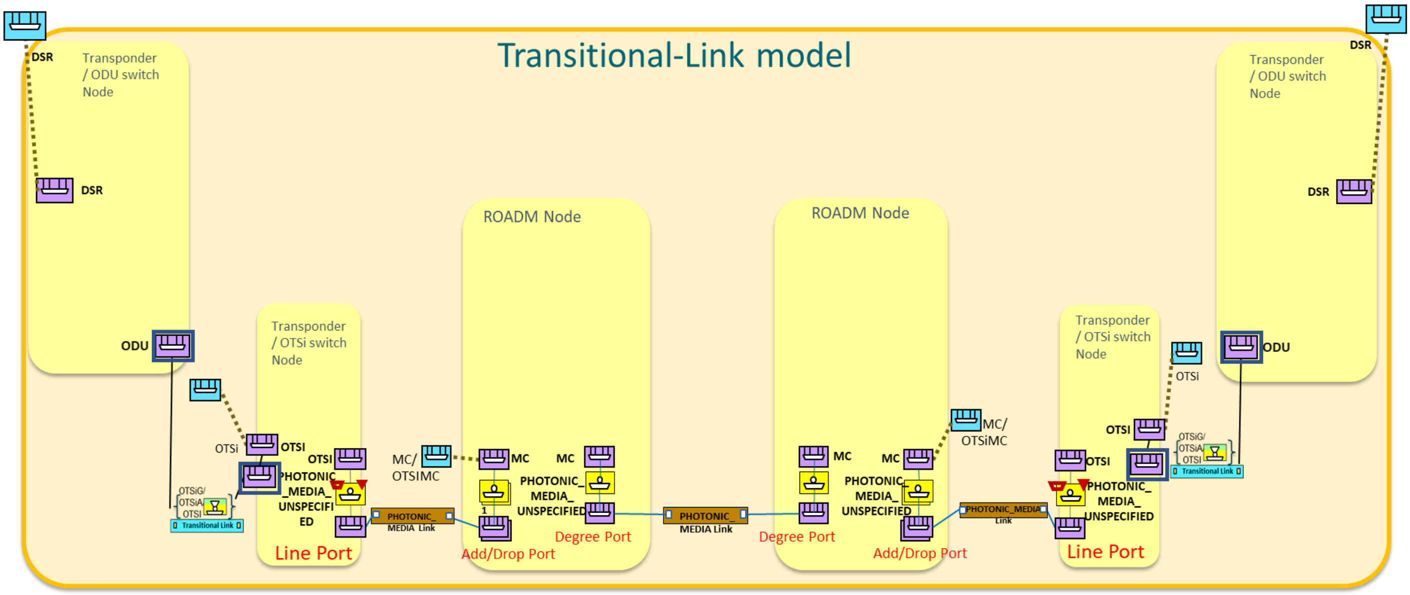


Figure 6-3 TAPI topology representation at "day 0" following Transitional Link modelling approach.

6.1.2.2.2 UC0b - Example 0: TAPI topology representation at "day 0" following Multi-layer Node modelling approach

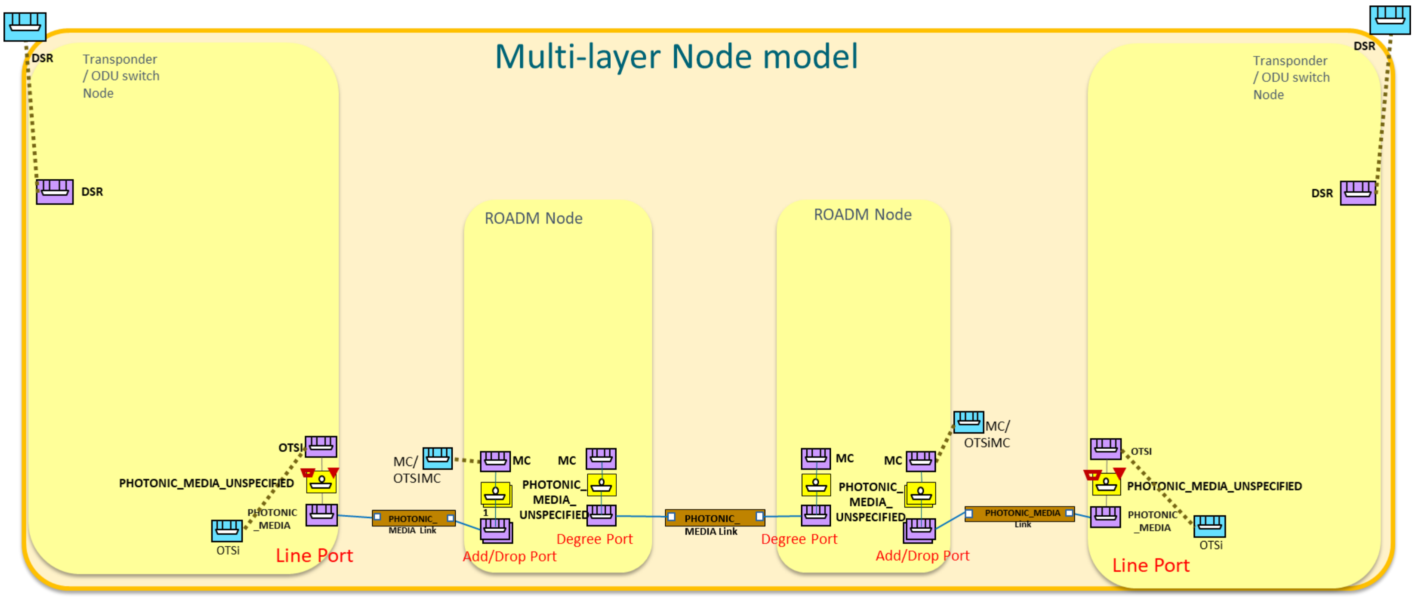
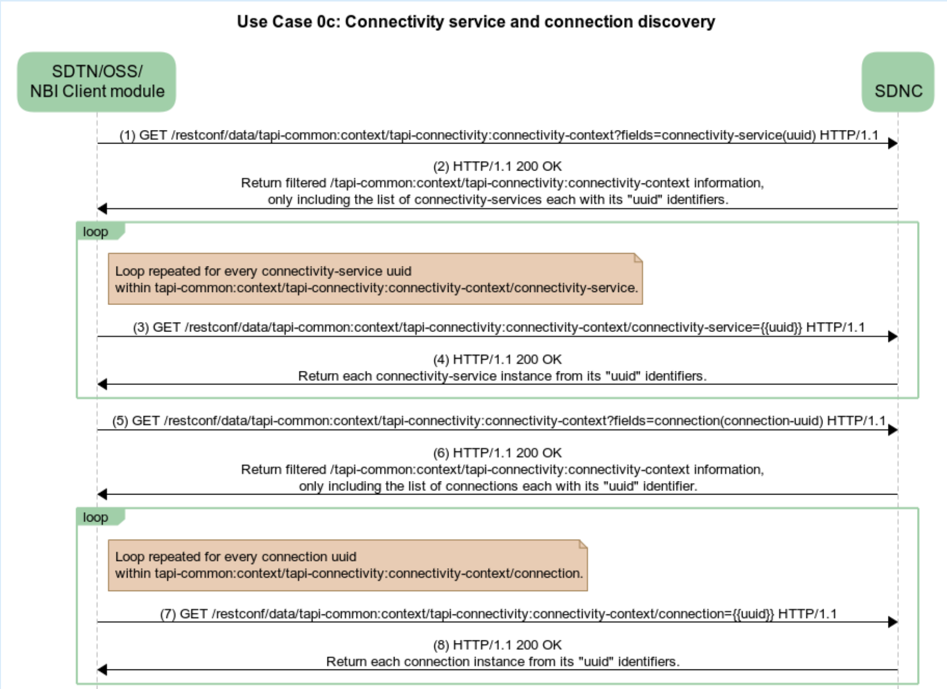


Figure 6-4 TAPI topology representation at "day 0" following Multi-layer Node modelling approach.

6.1.3 Use case 0c: Connectivity Service and Connection discovery (synchronous mode)

Number	UC0c
Name	Connectivity Service and Connection discovery (synchronous mode)
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, ODU, 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/ODU/PHOTONIC_MEDIA layers and the retrieval of the connections' information. 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.



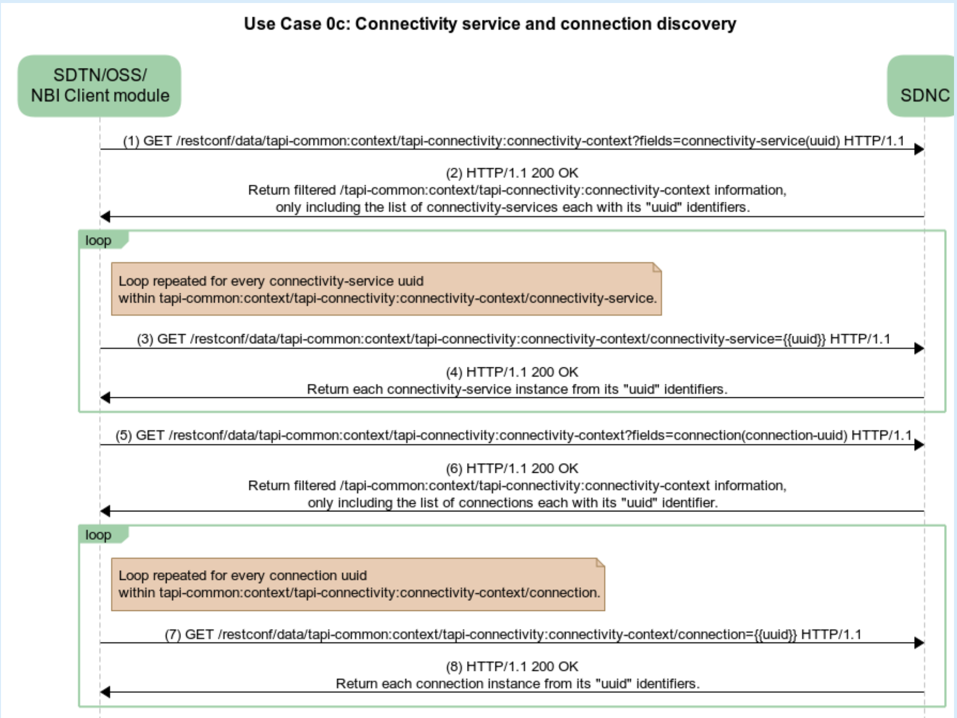
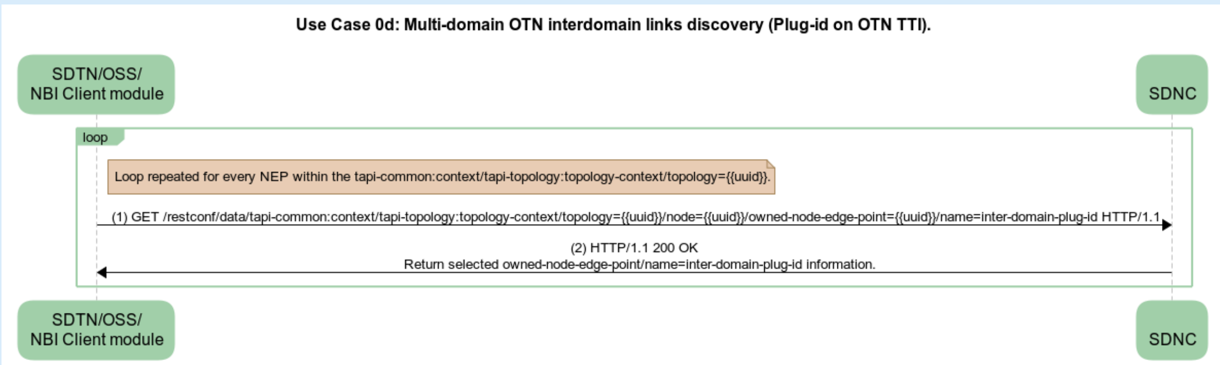


Figure 6-5 UC-0c: Connectivity Service - Workflows UC 0c-1 (top) and UC 0c-2 (bottom)

6.1.3.1 Relevant parameters

For the details about the parameters for each object retrieved in the previously described RESTCONF call operation, please refer to the UC1.0

6.1.4 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**, which can be encoded in TAPI as "name" attribute(s) of the tapi-topology:owned-node-edge-point objects representing the topology resource corresponding to the E-NNI interface.
Layers involved	DSR, ODU, 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 **6.1.2**. 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 6-6: UC0d workflow **Note**: the {inter-domain-plug-id} generically refers to either plug-id-inter-domain-local-id or plug-id-inter-domain-remote-id as defined below.

6.1.4.1 Relevant parameters

To support UC0d, the following name value pairs MUST be added:

node-edge-point	**/tapi-common:context/tapi-topology:topologycontext/topology/node/owned-node-edge-point**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
name	List of {name and name-value pairs}	RO	M	• Provided by *tapi-server*. Example values for illustration purposes.

See below an example of the required encoding:

```
"name" : [
  {
   "name" : "plug-id-inter-domain-local-id",
   "name-value":"localid-string"
  },
  {
   "name" : "plug-id-inter-domain-remote-id",
    "name-value":"remoteid-string"
   }...
```

6.1.4.2 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 mechanism proposed in this use case is the **inter-domain-plug-id** concept, which can be encoded in TAPI as "name" attribute of the ***tapi-topology:owned-node-edge-point*** objects representing the topology resource corresponding to the E-NNI interface. The ***tapi-topology:owned-node-edge-point*** object structure is presented in the following object:

```
/tapi-common:context:
    +--rw topology-context
       +--ro topology* [uuid]
          +--ro node* [uuid]
          |  +--ro owned-node-edge-point* [uuid]
          |  |  +--ro name* [inter-domain-plug-id]
          |  |  |  +--ro value-name    string
          |  |  |  +--ro value?        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.

6.1.4.2.1 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 shall consist 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 globally unique in its layer network (i.e., within the TAPI Context), the access point identifier should not change while the access point remains in existence.
- where it may be expected that the access point may be required for path set-up across an inter-operator boundary, the access point identifier must be available to other network operators.
- 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.
- the set of all access point identifiers belonging to a single administrative layer network should form a single access point identification scheme.
- the scheme of access point identifiers for each administrative layer network can be independent from the scheme in any other administrative layer network.

6.1.4.2.2 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)

6.2 Unconstrained E2E Service Provisioning

Definitions and Considerations

[**unconstrained**] The term indicates that the TAPI-Client is not introducing any **routing constraint** in the service request. The provisioning relies completely on the routing capabilities of the TAPI-Server to select the network resources employed to provide the desired service characteristics.

[**1:N out of scope**] For Connectivity Services, the inverse multiplexing case (1:N) support in TAPI 2.1.3 (in case of single SIP), using additional CSEP is not specified in this recommendation. The required *CSEPHasAssembledCSEPs* necessary association is defined in TAPI 2.3+.

[**server restrictions, bottom-up**] Generally speaking, a "bottom-up" approach, in which two different connectivity services (for example MC and OTSiMC) are established sequentially (first the server layer and next the client layer and the client one is constrained to use the server one) is possible. In this case, the client connectivity service is provisioned *referring to* the server connectivity service by uuid (*server-connectivity-service-end-point refers to an existing CSEP)*, and the user is able to specify constraints at each of the provisioning steps. The following yang tree clarifies the use of 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?              -> /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
```

[**server restrictions, top-down**] In addition to the bottom-up approach, some unconstrained service provisioning use cases (such as UC1c, UC1g and UC2b) request a Connectivity Service adding server layer 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:

- These use cases rely on the usage of additional CSEP(s) where appropriate (for example, 2 CSEPs at the source, 2 at the destination), using the *CSEPHasServerCSEP* relationship (*server-connectivity-service-end-point*). The purpose of the additional CSEPs is to ONLY convey restrictions that apply at the server layer. The CSEP here does not refer to an existing CSEP of another connectivity service.

- The *connectivity-service-uuid* within the server-connectivity-service-end-point MUST use *the same uuid value* of the (wrapping) connectivity service being requested.

- The *connectivity-end-point-local-id* within the server-connectivity-service-end-point MUST be the local id of the supporting CSEP.

- Both related CSEPs MUST refer to the same SIP. Note that, in constrained provisioning (Use Cases 3.X) routing constraints can be specified through e.g., include Link or Connectivity Service/Connection coroute inclusion.

- The server MAY instantiate as many top level and supporting connections as needed.

- The server MAY/MAY NOT 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. Note that if the MC connectivity service is created, the OTSiMC's *server-connectivity-service-end-point* still points to the OTSiMC connectivity service and the relationship between connectivity services is captured in the list of used connections. Both the OTSiMC and the MC connectivity services SHALL contain the MC top level connection within their respective ***tapi-connectivity:connectivity-service/connection*** list, as specified in [TAPI-CONN-MODEL-REQ-3]

- See UC10 for the guidelines referring to connectivity service deletion.

6.2.1 Use case 1.0: Generic Unconstrained 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, ODU
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 on. The underlying connection provisioning and management (including lower layer connections e.g., ODU, OTSi, MC, OTSiMC with intermediate regeneration connections if needed) is performed by the SDN Domain controller. The path of each lower layer connection (e.g., ODU or OCh/OTSi, OMS) across the network topology is calculated by the controller and the connection automatically provisioned. The “unconstrained” term refers that the TAPI-Client is not introducing any routing constraint in the service request, thus rely completely into the routing capabilities of the TAPI-Server to select the network resources employed to provide the desired service characteristics. Moreover, the TAPI-Client is not providing technology specific Traffic-Engineering constrains such time-slot selection at the ODU layer, or optical-spectrum selection for the routing of OTSi connections.
Layers involved	DSR/ODU/PHOTONIC_MEDIA
Type	Provisioning
Description & Workflow	The Use Case 1.0: Connectivity Service provisioning consists of the creation of a connectivity-service between Service-Interface-Points at the either the DSR ODU 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 response message with the Location Header as specified in https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html#sec9.5.

Once the resource has been created (here a pooling or event-trigger mechanism need to be defined in order to reconciliate the information), the NBI Client may retrieve all information of the Connectivity-Service (3), Connection list of references (5) and all its Connection objects (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 in Table 19 (Connectivity-Service) Table 20 (Connection).

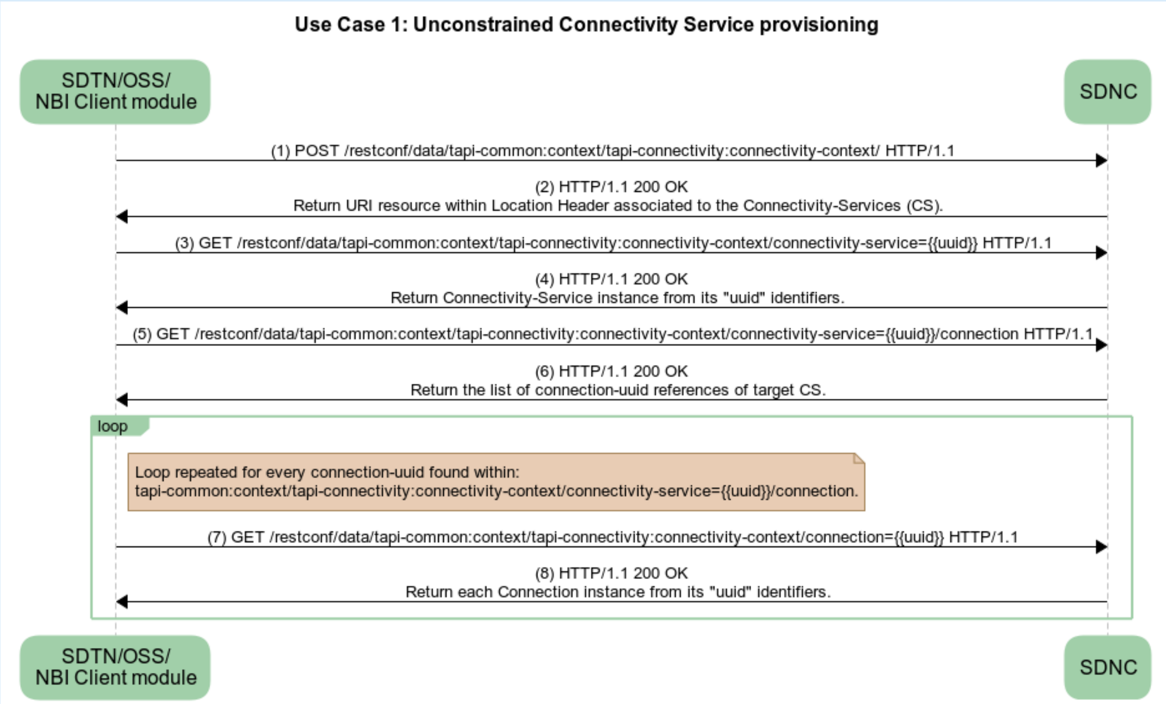


Figure 6-7 UC-1: Unconstrained end-to-end service provisioning.

NOTE: This RIA does not mandate the use of GET for a list, so Step (5) is optional and should be **/restconf/data/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service={{uuid}}?fields=connection(uuid)**

6.2.1.1 Relevant parameters

Table 19: 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. The canonical representation uses lowercase	RW	M	• Provided by *tapi-client*
name	List of {value-name: value} 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.*
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	M	• Provided by *tapi-server*
requested-capacity	"total-size": {value: ,unit:} • "value": "[0-9]{8}", • "unit": depending on the requested layer (e.g., "GBPS" for DSR or "GHz" for PHOTONIC_MEDIA)	RW	C	• Provided by *tapi-client.* • Whether this object is mandatory will depend on the layer and use case. • NOTE: It is mandatory for PHOTONIC_MEDIA when specifying a slot width. • NOTE: TAPI v2.3 includes the **layer-protocol-qualifier** so the requested-capacity MAY be omitted if there is no ambiguity.
service-type	"POINT_TO_POINT_CONNECTIVITY", or "POINT_TO_MULTIPOINT_CONNECTIVITY"	RW	O	• Provided by *tapi-client*
service-layer	One of the values "DSR", "ODU" or "PHOTONIC_MEDIA"	RW	M	• Provided by *tapi-client*
preferred-transport-layer	One of the values "DSR", "ODU" or "PHOTONIC_MEDIA"	RW	O	• 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. Since TAPI v2.1.3 does not specify the OMS/OTS layers as specific protocol qualifiers, this

				RIA does not mandate the listing of layers below the MC.
end-point	List of {**connectivity-service-end-point**}	RW	M	• Provided by *tapi-client* • Min elements 2. Note: even in P2P services there may be >2 endpoints (e.g., to specify server layer constraints).

The Connectivity service includes at least two Connectivity-service-end-points (CSEPs).

Table 20: 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	"[0-9a-zA-Z_]{32}"	RW	M	• Provided by *tapi-client*
name	List of {value-name: value} 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", "ODU" or "PHOTONIC_MEDIA"	RW	M	• Provided by *tapi-client*
layer-protocol-qualifier	Depends on the Layer Protocol Name	RW	M	• Provided by tapi-server • For DSR, all children identities defined for "DIGITAL_SIGNAL_TYPE" MUST be supported. • For ODU, all children identities defined for ODU_TYPE MUST be supported. • For photonic media, all children identities defined for PHOTONIC_LAYER_QUALIFIER MUST be supported.
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", "INPUT", "OUTPUT"]	RW	O	• Provided by *tapi-client*
role	One of ["SYMMETRIC", "ROOT", "LEAF", "TRUNK" or "UNKNOWN"]	RW	O	• Provided by *tapi-client* • This RIA only considers P2P and SYMMETRIC
capacity	"total-size": {value: unit} • "value": "[0-9]{8}", • "unit": see CS	RW	O	• Provided by *tapi-client* • Depends on the Layer and Use Case
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 top-level connection CEPs that are instantiated over the NEP that the CSEP's SIP is bound to.

Table 21: Connection object definition

connection	**/tapi-common:context/tapi-connectivity: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 "DSR", "ODU", "PHOTONIC_MEDIA" depending on the Layer of the connection	RO	M	• Provided by *tapi-client*
operational-state	One of ["ENABLED", "DISABLED"]	RO	M	• Provided by *tapi-server*
lifecycle-state	One of ["PLANNED", "POTENTIAL_AVAILABLE",	RO	M	• Provided by *tapi-server*

	"POTENTIAL_BUSY", "INSTALLED", "PENDING_REMOVAL"]			
direction	One of ["UNIDIRECTIONAL", "BIDIRECTIONAL"]	RO	M	• Provided by *tapi- server* • See tapi-common:forwarding-direction
lower-connection	List of {**connectivity-ref** - */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 may be not applicable or be empty: i) Cross-connections, ii) Top-connections where the representation of lower partitioning levels does not provide further information (e.g., OTSi where the XC is encapsulated in the termination point). As described in [TAPI-CONN-MODEL-REQ-16]
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*
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.
supported-client-link	List of {link-ref - topology-uuid + link-uuid	RO	O	• Provided by *tapi-server*

Table 22: 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", "ODU", "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", "PHOTONIC_LAYER_QUALIFIER”] MAY be supported depending on the relevant qualifier .
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	M	• Provided by *tapi-server*
termination-state	One of ["LP_CAN_NEVER_TERMINATE", "LT_NOT_TERMINATED", "TERMINATED_SERVER_TO_CLIENT_FLOW", "TERMINATED_CLIENT_TO_SERVER_FLOW", "TERMINATED_BIDIRECTIONAL", "LT_PERMENANTLY_TERMINATED", "TERMINATION_STATE_UNKNOWN"]	RO	O	• Provided by *tapi-server* • *NOTE: This is considered optional for RIA 1.1. Additional guidelines for the usage of this data node will be provided in a subsequent version of this specification.*
termination-direction	One of ["BIDIRECTIONAL", “SINK”, “SOURCE”]	RO	M	• Provided by *tapi-server*
connection-port-direction	One of ["BIDIRECTIONAL","INPUT","OUTPUT"]	RO	M	• 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*
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*
client-node-edge-point	List of { ***node-edge-point-ref*** }	RO	M	• Provided by *tapi-server*
tapi-odu:odu-connection-end-point-spec	**{odu-connection-end-point-spec}**	RO	C	• Provided by *tapi-server* • MUST augment CEPs at the ODU layer • See Table 23
tapi-photonic-media:otsi-connection-end-point-spec	**{ otsi-connection-end-point-spec }**	RO	C	• Provided by *tapi-server* • MUST augment CEPs at the PHOTONIC_MEDIA layer with OTSi layer-protocol-qualifier. • See • Table 24
tapi-photonic-media:media-channel-connection-end-point-spec	**{media-channel-connection-end-point-spec}**	RO	C	• Provided by *tapi-server* • MUST augment CEPs at the PHOTONIC_MEDIA layer with MC/OTSiMC layer-protocol-qualifier. • MAY augment CEPs at the PHOTONIC_MEDIA layer with OTSi layer-protocol-qualifier (as per [TAPI-TOP-MODEL-REQ-16].
tapi-photonic-media:ots-connection-end-point-spec	**{ots-connection-end-point-spec}**	RO	C	• Provided by *tapi-server* • MAY augment CEPs with protocol-qualifier:LAYER_PROTOCOL_QUALIFIER_UNSPECIFIED for transceiver line ports and MUST augment such CEPs for ROADMs • See Table 27

Table 23: 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-odu:odu-connection-end-point-spec**

Attribute	Allowed Values/Format	Mod	Sup	Notes
odu-common	{ odu-type, odu-rate, odu-rate-tolerance } • "odu-type": [ODU_TYPE], • "odu-rate": [0-9]{12}, • "odu-rate-tolerance": [0-9]{12},	RO	M	• Provided by *tapi-server* • All children identities defined for ["ODU_TYPE"] MUST be supported. • **odu-rate-tolerance** Standardized values are defined in Table 7-2/G.709

odu-term-and-adapter	{ opu-tributary-slot-list, auto-payload-type, configured-client-type, configured-mapping-type, accepted-payload-type, named-payload-type, hex-payload-type}
	• 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
	  ○ "named-payload-type": ["UNKNOWN", "UNINTERPRETABLE"]
	  ○ "hex-payload-type": "[0-9]{64}",
	• fec-parameters : { pre-fec-ber, post-fec-ber, corrected-bytes, corrected-bits, uncorrectable-bytes, uncorrectable-bits}
	  ○ "pre-fec-ber": "[0-9]{64}",
	  ○ "post-fec-ber": "[0-9]{64}",
	  ○ "corrected-bytes": "[0-9]{64}",
	  ○ "corrected-bits": "[0-9]{64}",
	  ○ "uncorrectable-bytes": "[0-9]{64}",
	  ○ "uncorrectable-bits": "[0-9]{64}",
	• odu-cn-effective-time-slot: List of "[0-9]{64}"
		RO	M	• Provided by *tapi-server*
				• *Configured-client-type* accepts any child identities defined for ["DIGITAL_SIGNAL_TYPE"]
				• odu-term-and-adapter *is mandatory for CEPs that are TTP.*

odu-ctp	{tributary-slot-list, tributary-port-number, accepted-msi} • tributary-slot-list : List of "[0-9]{64}" • tributary-port-number: "[0-9]{64}" • accepted-msi? string	RO	M	• Provided by *tapi-server*

odu-protection	aps-enable : Boolean aps-level: "[0-9]{64}"	RO	O	• Provided by tapi-server

Table 24: otsi-connection-end-point-spec (**CEP**) object definition

otsi-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-connection-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**

otsi-termination	includes {
	selected-central-frequency,
	selected-application-identifier,
	selected-modulation,
	selected-spectrum,
	transmitted-power,
	received-power,
	laser-properties
	}
		RO	M	• Provided by *tapi-server*

where				
selected-central-frequency	{ central-frequency, frequency-constraint: {adjustment-granularity, grid-type} } • "central-frequency": "[0-9]{9}", • "adjustment-granularity":["UNCONSTRAINED", "G_3_125GHZ", "G_6_25GHZ", "G_12_5GHZ", "G_25GHZ", "G_50GHZ", "G_100GHZ",] • "grid-type": ["GRIDLESS", "FLEX", "CWDM", "DWDM"]	RO	M	• Provided by *tapi-server* • The **central-frequency** of the laser specified in MHz. It is the oscillation frequency of the corresponding electromagnetic wave. • **Adjustment-granularity** in Gigahertz. As per ITU-T G.694.1, it is used to calculate nominal central frequency". • The **grid-type** specifies the reference set of frequencies used to denote allowed nominal central frequencies that may be used for defining applications.

selected-application-identifier	{
	application-identifier-type,
	application-code
	}
	• "application-identifier-type":["PROPRIETARY", "ITUT_G959_1", "ITUT_G698_1", "ITUT_G698_2", "ITUT_G696_1", "ITUT_G695",]
	• "application-code":"[0-9a-zA-Z_]{64}"
		RO	M	• Provided by tapi-server

selected-modulation	One of ["RZ", "NRZ", "BPSK", "DPSK", "QPSK", "8QAM", "16QAM", "PAM4", "PAM8"]	RO	M	• Provided by *tapi-server*

selected-spectrum	{lower-frequency, upper-frequency, frequency-constraint: {adjustment-granularity, grid-type} } • "upper/lower-frequency": "[0-9]{9}", • "adjustment-granularity":["UNCONSTRAINED", "G_3_125GHZ", "G_6_25GHZ", "G_12_5GHZ", "G_25GHZ", "G_50GHZ", "G_100GHZ",] • "grid-type": ["GRIDLESS", "FLEX", "CWDM", "DWDM"]	RO	M	• Provided by *tapi-server* • The **frequencies** are specified in MHz. • **Adjustment-granularity** in Gigahertz. • The **grid-type** specifies the reference set of frequencies used to denote allowed nominal central frequencies.
transmited-power	{total-power, power-spectral-density} • "total-power":"[0-9].[0-9]{7}", • "power-spectral-density": "[0-9].[0-9]{7}",	RO	M	• Provided by *tapi-server* *NOTE: The term "transmited" is as it appears in the 2.1.3 Yang model.*
received-power	{total-power, power-spectral-density} • "total-power":"[0-9].[0-9]{7}", • "power-spectral-density": "[0-9].[0-9]{7}",	RO	M	• Provided by *tapi-server*
laser-properties	{laser-status, laser-application-type, laser-bias-current, laser-temperature} • "laser-status": ["ON","OFF","PULSING", "UNDEFINED"] • "laser-application-type": ["PUMP", "MODULATED", "PULSE"] • "laser-bias-current": "[0-9].[0-9]{7}", • "laser-temperature": "[0-9].[0-9]{7}",	RO	O	• Provided by *tapi-server*
NOTE: Additional Requirements				
OTU Support	The **tapi-photonic-media:otsi-connection-end-point-spec** augmentation does not include a "name"-based extensibility mechanism Implementations SHOULD use the "name" list in the base cep-list/connection-end-point object and include:	RO	O	

	1) "value-name" : "BAUD_RATE" "value": string encoding the baud rate (in Gigabaud) 2) "value-name" : "FEC_TYPE" "value": one of { "STANDARD_FEC_TYPE", "PROPRIETARY_FEC_TYPE" }			

Table 25: OTSI-Assembly-connection-end-point-spec (**OTSiA CEP**) object definition

otsi-assembly-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-assembly-connection-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
otsi-adapter	"number-of-otsi": [0-9]{9}	RO	O	• Provided by *tapi-server* *NOTE.1: The object otsi-assembly-connection-end-point-spec is now marked as deprecated in TAPI v2.1.3. Its use is no longer recommended.* *NOTE.2: The number of OTSi elements can be inferred either by:* *i) retrieving the OTSi CEP instances sharing the same client ODU NEP. Note that the OTSiG cannot be assumed to be transported over a single UNSPECIFIED NEP (several line ports) or* *ii) by adding a value-pair to the client ODU CEP (piggy-backing).*

Table 26: Media-channel-connection-end-point-spec (**MC CEP**) object definition

media-channel-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:media-channel-connection-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
media-channel	{occupied-spectrum, measured-power-ingress, measured-power-egress}	RO	M	Mandatory only for PHOTONIC_LAYER_QUALIFIER_MC layer-protocol-qualified CEPs
occupied-spectrum	{lower-frequency, upper-frequency, frequency-constraint} • "upper/lower-frequency": "[0-9]{9}", • "frequency-constraint": {adjustment-granularity, grid-type} ○ "adjustment-granularity": ["UNCONSTRAINED", "G_3_125GHZ", "G_6_25GHZ", "G_12_5GHZ", "G_25GHZ", "G_50GHZ", "G_100GHZ",] ○ "grid-type": ["GRIDLESS", "FLEX", "CWDM", "DWDM"]	RO	M	• Provided by *tapi-client* • The **upper/lower-frequency** boundaries of the band specified in MHz. • **Adjustment-granularity** in Gigahertz. As per ITU-T G.694.1, it is used to calculate nominal central frequency". • The **grid-type** specifies the reference set of frequencies used to denote allowed nominal central frequencies
measured-power-ingress	{total-power, power-spectral-density} • "total-power":"[0-9].[0-9]{7}", • "power-spectral-density": "[0-9].[0-9]{7}",	RO	C	• Provided by *tapi-server* • *NOTE: The presence of this attribute depends on actual hardware power monitoring capability.*
measured-power-egress	{total-power, power-spectral-density} • "total-power":"[0-9].[0-9]{7}", • "power-spectral-density": "[0-9].[0-9]{7}",	RO	C	• Provided by *tapi-server* • *NOTE: The presence of this attribute depends on actual hardware power monitoring capability.*

Table 27: ots-connection-end-point-spec (**OTS CEP**) object definition

ots-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-connection-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**

ots-media-channel	{ occupied-spectrum, measured-power-ingress, measured-power-egress }	RO	M	MAY augment protocol-qualifier:LAYER_PROTOCOL_QUALIFIER_UNSPECIFIED CEPs for transceiver line ports and MUST augment such CEPs for ROADMs
with				
occupied-spectrum	{lower-frequency, upper-frequency, frequency-constraint} • "upper/lower-frequency": "[0-9]{9}", • "frequency-constraint": {adjustment-granularity, grid-type} o "adjustment-granularity": ["UNCONSTRAINED", "G_3_125GHZ", "G_6_25GHZ", "G_12_5GHZ", "G_25GHZ", "G_50GHZ", "G_100GHZ",] o "grid-type": ["GRIDLESS", "FLEX", "CWDM", "DWDM"]	RO	M	• Provided by *tapi-client* • The **upper/lower-frequency** boundaries of the band (in MHz). • **Adjustment-granularity** in Gigahertz. As per ITU-T G.694.1, it is used to calculate nominal central frequency. • The **grid-type** specifies the reference set of frequencies used to denote allowed nominal central frequencies
measured-power-ingress	{total-power, power-spectral-density} • "total-power":"[0-9].[0-9]{7}", • "power-spectral-density": "[0-9].[0-9]{7}",	RO	M	• Provided by *tapi-server*
measured-power-egress	{total-power, power-spectral-density} • "total-power":"[0-9].[0-9]{7}", • "power-spectral-density": "[0-9].[0-9]{7}",	RO	M	• Provided by *tapi-server*

Table 28: 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	List of {value-name: value} "value-name": "ROUTE_NAME" "value": " [0-9a-zA-Z_]{64}"	RO	M	• Provided by *tapi-server*
connection-end-point	List of {"connection-end-point-ref - */tapi-common:context/tapi-topology:topology-context/topology/node/owned-node-edge-*	RO	M	• Provided by *tapi-server*

	point/tapi-connectivity:cep-list/connection-end-point/uuid "}			

Table 29: otsia-connectivity-service-end-point-spec (**OTSiA CSEP**) object definition

otsia-connectivity-service-end-point-spec	**/tapi-common:context/tapi-connectivity:connectivity-service/end-point/otsia-connectivity-service-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
number-of-otsi	[0-9]{9}	RW	M	• Provided by *tapi-client*
otsi-config	List of {otsi-config [local-id]} With otsi-config:{ central-frequency, spectrum, application-identifier, modulation, laser-control, transmit-power, total-power-warn-threshold-upper, total-power-warn-threshold-lower, local-id, name}	RW	M	• Provided by *tapi-client* • **NOTE:** **total-power-warn-threshold** attribute(s) are optional.
where				

central-frequency	• "central-frequency": "[0-9]{9}",
	• "frequency-constraint": {adjustment-granularity, grid-type}
	  ○ "adjustment-granularity": ["UNCONSTRAINED", "G_3_125GHZ", "G_6_25GHZ", "G_12_5GHZ", "G_25GHZ", "G_50GHZ", "G_100GHZ",]
	  ○ "grid-type": ["GRIDLESS", "FLEX", "CWDM", "DWDM"]
		RW	M	• Provided by *tapi-client*
				• The **central-frequency** of the laser specified in MHz. It is the oscillation frequency of the corresponding electromagnetic wave.
				• **Adjustment-granularity** in GHz, as per ITU-T G.694.1. It is used to calculate nominal central frequency.
				
				• The **grid-type** specifies the reference set of frequencies used to denote allowed nominal central frequencies.

spectrum	{lower-frequency, upper-frequency, frequency-constraint} • "upper/lower-frequency": "[0-9]{9}", • "frequency-constraint": {adjustment-granularity, grid-type}	RW	M	• Provided by *tapi-client* • The **upper/lower-frequency** boundaries of the band specified in MHz. • **Adjustment-granularity** in Gigahertz. As per ITU-T G.694.1, it

	○ “adjustment-granularity”: [“UNCONSTRAINED”, “G_3_125GHZ”, “G_6_25GHZ”, “G_12_5GHZ”, “G_25GHZ”, “G_50GHZ”, “G_100GHZ”,] ○ “grid-type”: [“GRIDLESS”, “FLEX”, “CWDM”, “DWDM”]			is used to calculate nominal central frequency". • The **grid-type** specifies the reference set of frequencies used to denote allowed nominal central frequencies
application-identifier	{application-identifier-type, application-code} • “application-identifier-type”:[“PROPRIETARY”, “ITUT_G959_1”, “ITUT_G698_1”, “ITUT_G698_2”, “ITUT_G696_1”, “ITUT_G695”,] “application-code”:”[0-9a-zA-Z_]{64}"	RW	M	• Provided by *tapi-client*
modulation	One of [“RZ“, “NRZ”, “BPSK”, “DPSK”, “QPSK”, “8QAM”, “16QAM”, "PAM4", "PAM8"]	RW	M	• Provided by *tapi-client*
transmit-power	{total-power, power-spectral-density} • “total-power”:"[0-9].[0-9]{7}", • “power-spectral-density”: "[0-9].[0-9]{7}",	RW	M	• Provided by *tapi-client*
laser-control	One of {"FORCED-ON", "FORCED-OFF", "AUTOMATIC-LASER-SHUTDOWN", "UNDEFINED"]	RW	O	• Provided by *tapi-client* • NOTE: The usage of this object will be clarified in subsequent versions of this RIA.
total-power-warn-threshold-upper	[0-9].[0-9]{7}	RW	O	• Provided by *tapi-client*
total-power-warn-threshold-lower	[0-9].[0-9]{7}	RW	O	• Provided by *tapi-client*
local-id	"[0-9a-zA-Z_]{32}"	RW	M	• Provided by *tapi-client*
Additional requirements				

OTU Support	This requirement involves the selection of the FEC and Baud Rate. Although each component OTSi has a "name" extension mechanism, this RIA assumes that the FEC and Baud Rate are the same for all the group. Consequently, this RIA specifies the use of the CSEP "name" list in the base **tapi-connectivity:end-point** object and include: 1) "value-name" : "BAUD_RATE" "value": string encoding the baud rate (in Gigabaud) 2) "value-name" : "FEC_TYPE" "value": one of { "STANDARD_FEC_TYPE", "PROPRIETARY_FEC_TYPE" }	RO	O	• Provided by *tapi-client*

Table 30: mca-connectivity-service-end-point-spec (**MCA CSEP**) object definition

mca-connectivity-service-end-point-spec	**/tapi-common:context/tapi-connectivity:connectivity-service/end-point/mca-connectivity-service-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
number-of-mc	[0-9]{9}	RW	M	• Provided by *tapi-client*.
capacity	{value: unit} • "value": "[0-9]{8}", • "unit": ["GHz"]	RW	C	• Provided by *tapi-client*. Total capacity of an MCA/OTSiMCA. It may be omitted if spectrum is specified. Cannot be omitted if spectrum is not specified. • *NOTE*: In 2.3 this is replaced by a list *media-channel-bw-config-pac* that allows specifying individual MC *widths.*
mc-config	List of {mc-config [local-id]} with { spectrum,	RW	M	• Provided by *tapi-client.*

	power-management-config-pac }			
Where				

spectrum	{lower-frequency, upper-frequency, frequency-constraint}
	• “upper/lower-frequency": "[0-9]{9}",
	• "frequency-constraint": {adjustment-granularity, grid-type}
	  o “adjustment-granularity”: [“UNCONSTRAINED”, “G_3_125GHZ”, “G_6_25GHZ”, “G_12_5GHZ”, “G_25GHZ”, “G_50GHZ”, “G_100GHZ”,]
	  o “grid-type”: [“GRIDLESS”, “FLEX”, “CWDM”, “DWDM”]
		RW	M	• Provided by *tapi-client*
				• The **upper/lower-frequency** boundaries of the band specified in MHz.
				• **Adjustment-granularity** in Gigahertz. As per ITU-T G.694.1, it is used to calculate nominal central frequency".
				
				• The **grid-type** specifies the reference set of frequencies used to denote allowed nominal central frequencies

power-management-config-pac	{
	intended-maximum-output-power, intended-minimum-output-power, expected-maximum-input-power, expected-minimum-input-power
	}
	• "intended-maximum-output-power":{total-power, power-spectral-density}
	  o "total-power":"[0-9].[0-9]{64}",
	  o "power-spectral-density":"[0-9].[0-9]{64}"
	• "intended-minimum-output-power":{total-power, power-spectral-density}
	  o "total-power":"[0-9].[0-9]{64}",
	  o "power-spectral-density":"[0-9].[0-9]{64}"
	• "expected-maximum-output-power":{total-power, power-spectral-density}
	  o "total-power":"[0-9].[0-9]{64}",
	  o "power-spectral-density":"[0-9].[0-9]{64}"
		RW	C	• Provided by *tapi-client*
				• *NOTE: The client MAY provide power indications if available.*
				
				Example: assuming an OLS controller with an MCA domain:
				
				expected-maximum-input-power, expected-minimum-input-power MAY convey the attached transceiver max and min launch (TX) power. (expected from the transceiver)
				This specifies constraints related to power tolerance at the input
				
				intended-maximum-output-power, intended-minimum-output-power MAY convey the power intended to be delivered to the local transceiver i.e., after the signal has crossed the amplification/attenuation of the optical line system.
				(intended to be delivered)
				This specifies constraints related to power that the OLS should guarantee.

	• "expected-minimum-output-power":{total-power, power-spectral-density} ○ "total-power":"[0-9].[0-9]{64}", ○ "power-spectral-density":"[0-9].[0-9]{64}"			
local-id	"[0-9a-zA-Z_]{32}"	RW	M	Provided by *tapi-client*

6.2.1.2 Expected results

In the following subsections we include detail examples of the expected results after the successful provisioning of connectivity-services. Please note that all examples follow the rules detailed in section 5.1.1. To show some detailed TAPI connectivity examples, first a simple legend of icons and basic arrangement is included.

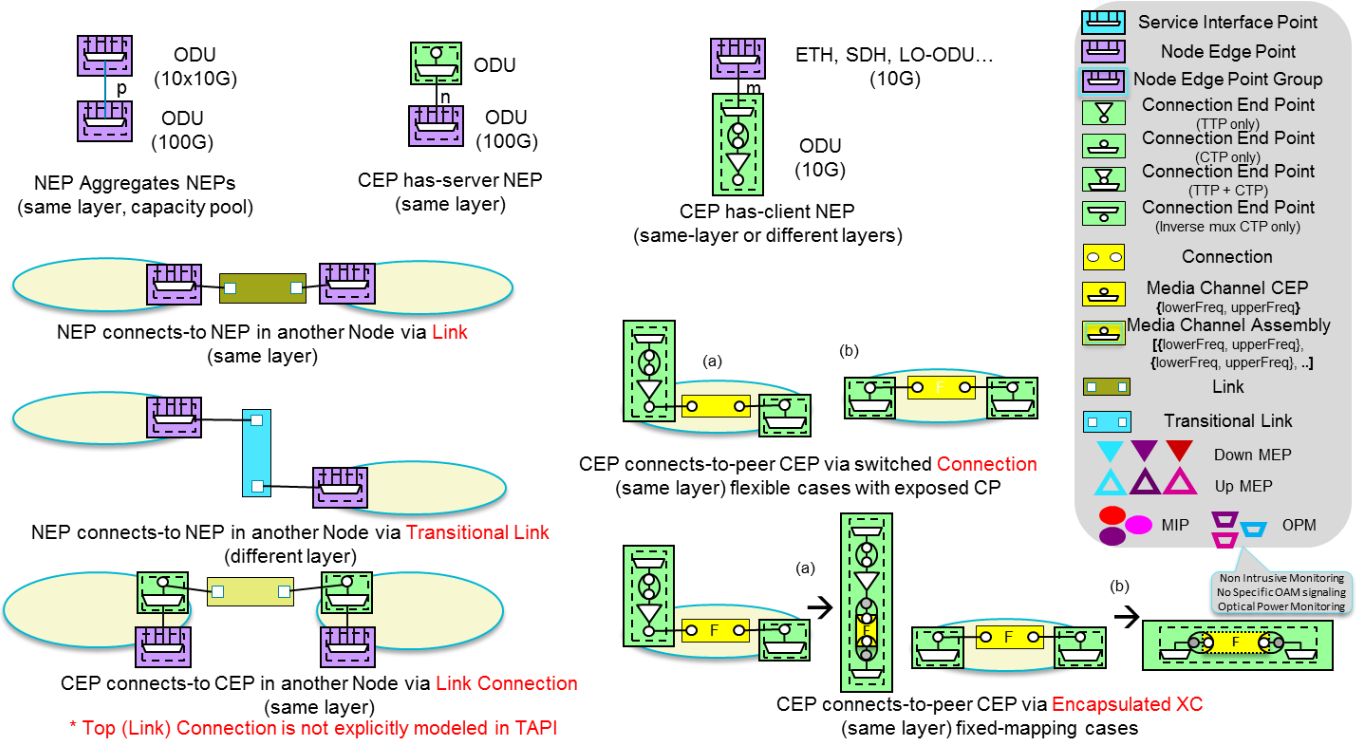


Figure 6-8 TAPI Logical Termination Point Template – Basic Arrangements.

6.2.1.2.1 Example: 10GE client signal over ODU2 over ODU4 (DSR-ODU Fixed Mapping, flexible ODU allocation)

The following diagrams illustrate a possible sequence of generation of the required TAPI topology and connectivity objects and its relationships according to the rules described in section 5.1.1. However, the internal TAPI server workflow MAY vary and, if notification or streaming services are available, the sequence of asynchronous notifications received by the TAPI client MAY be different. Thus, the objective of this and the subsequent examples detailing the use cases is to illustrate the object composition and most importantly to define the expected result after a connectivity-service provisioning.

Please note, that in the following examples (all included examples in Section 6), a modelling simplification (represented by blue square) of the client interface (UNI) has been introduced. For the current version of the RIA, the modelling of the UNI client facing side interfaces is not yet covered in detail, so the actual implementation decisions are left to the vendor according to each individual HW capabilities. Thus, please consider the following examples as a guideline of representation the connectivity modelling of the network facing side (e.g., the representation of how the multiplexing of DSR signals over ODU over OTSi shall be modelled). In the following figure we include the general assumed UNI simplifications represented in the examples included in this version of the RIA. The following figure represents different assumptions done in the UNI representation.

- Option 1: Assumes there is not flexibility at the DSR layer, but the fixed cross-connection (at DSR layer) are explicitly represented.
- Option 2: Assumes there is not flexibility at the DSR layer, and the fixed cross-connection (at DSR layer) are not explicitly represented.
- Option 3: Assumes there is flexibility at the DSR layer.

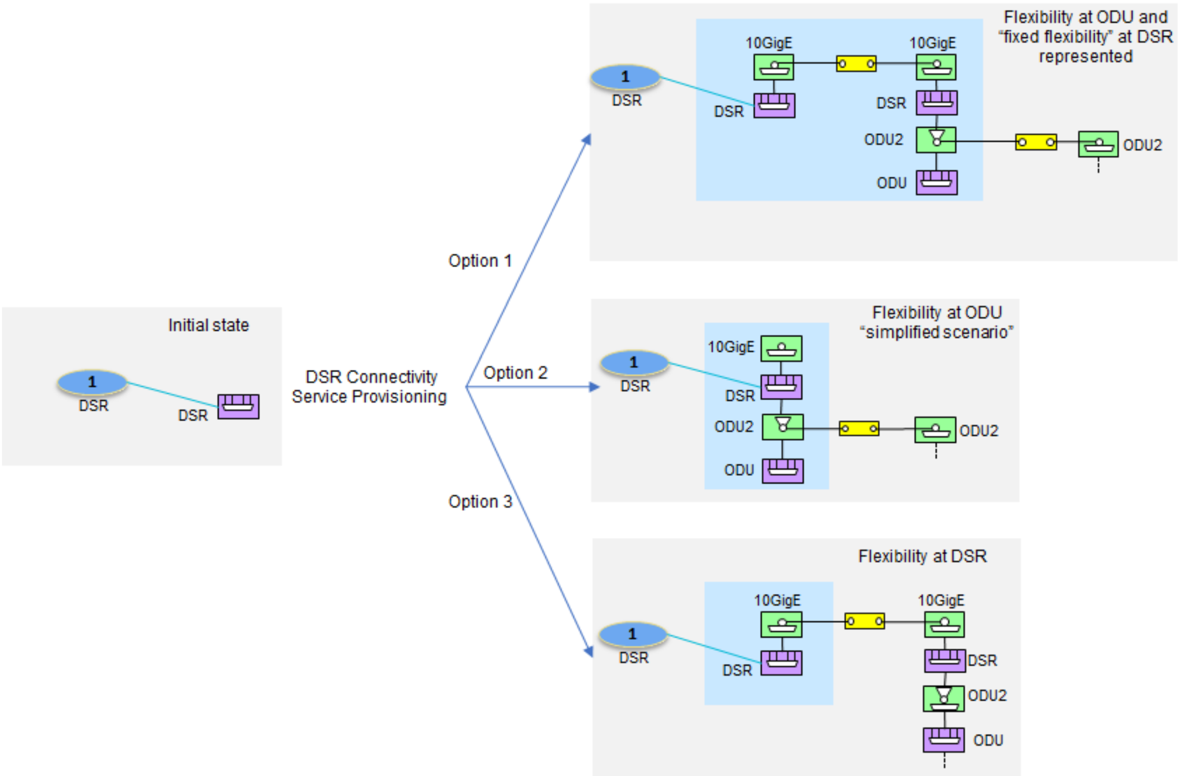


Figure 6-9 UNI Modelling simplifications

Please note, that the following versions of the RIA will address the UNI representation in detail so the previous simplifications may not be longer valid.

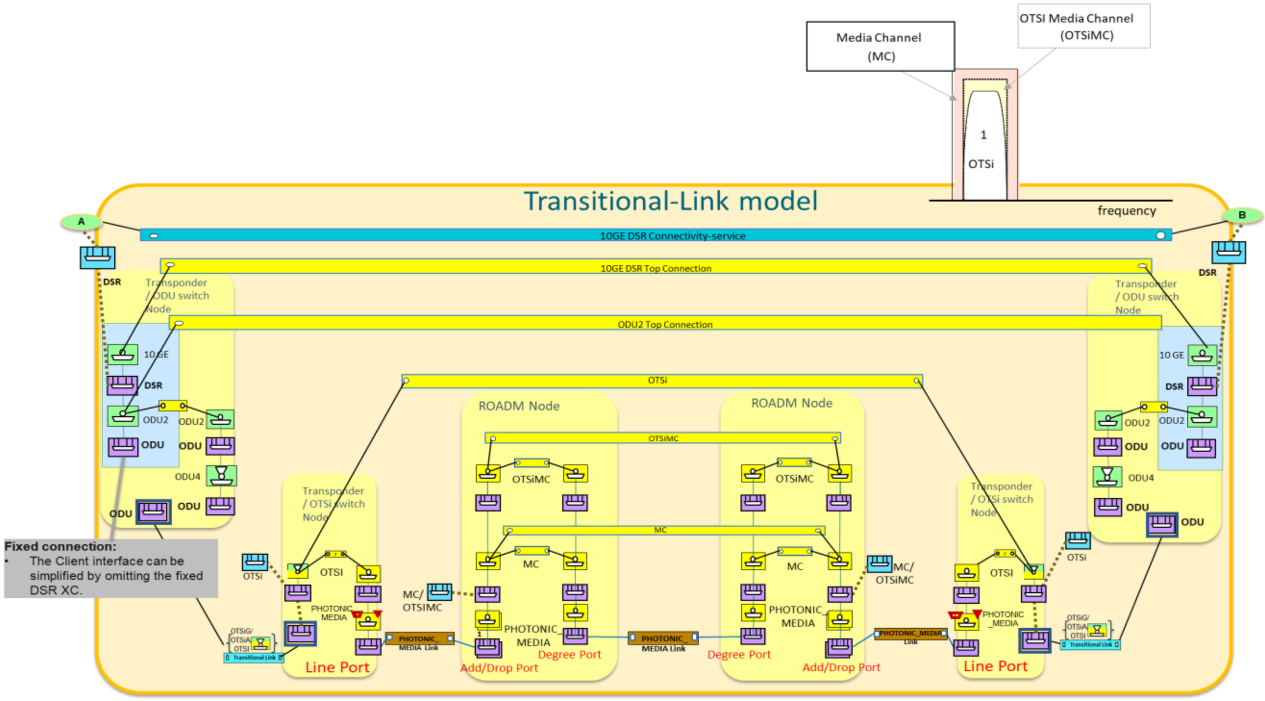


Figure 6-10 Connectivity Service 10GE client signal over ODU2 (DSR-ODU Fixed Mapping) over ODU4 over single OTSi – Transitional Link modelling

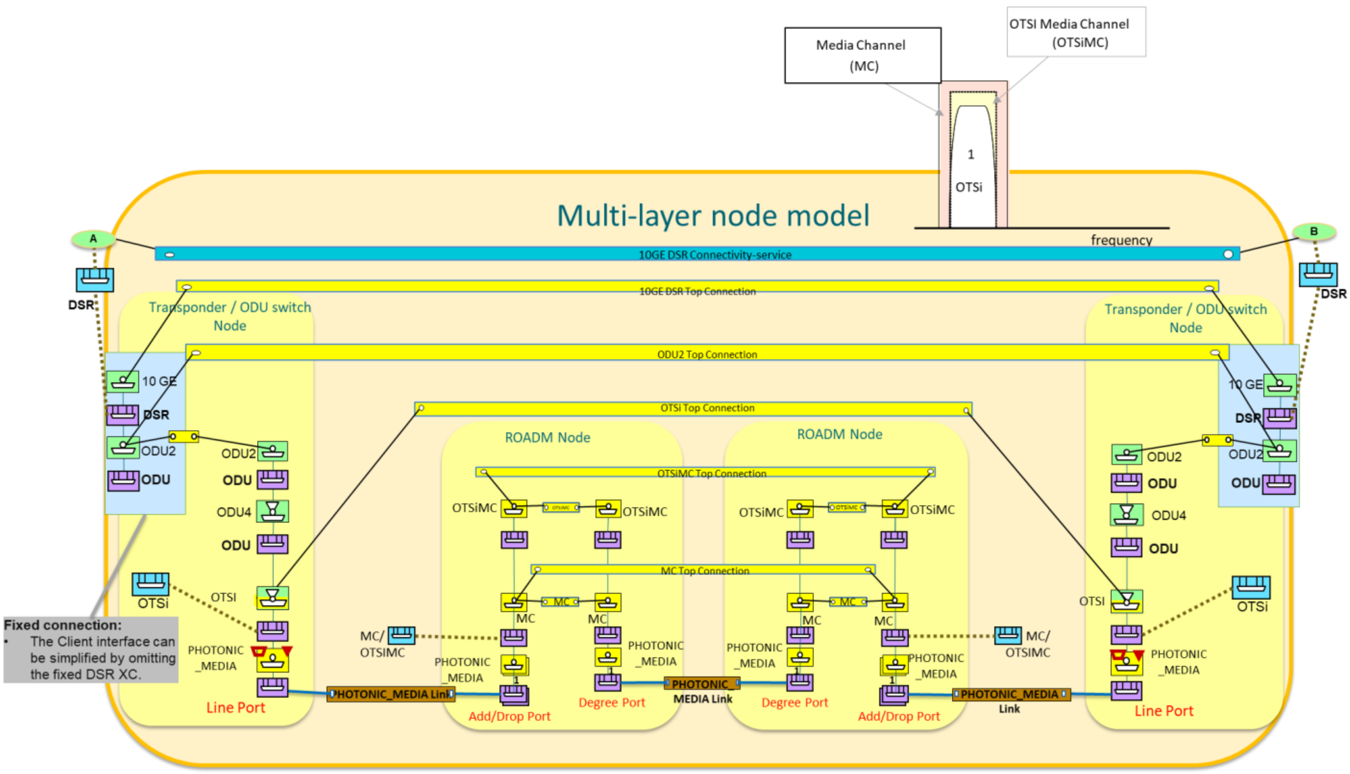


Figure 6-11 Connectivity Service 10GE client signal over ODU2 (DSR-ODU Fixed Mapping) over ODU4 over single OTSi – Multi-layer node modelling.

6.2.1.2.2 Example: 1GE client signal over ODU0 over ODU2 over ODU4 (Fixed DSR-ODU mapping, flexible ODU allocation)

For simplification OTSi and Photonic Media layers have been omitted from this example, as it is the exact same case depicted in the previous example.

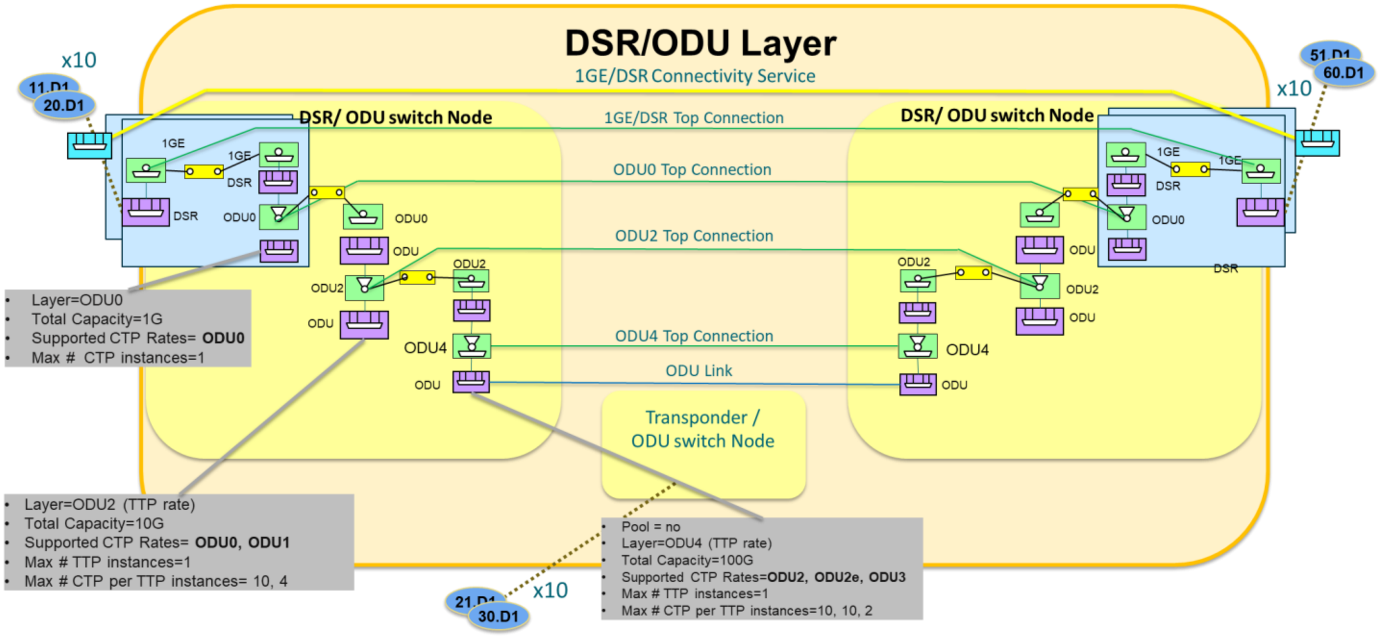


Figure 6-12 Connectivity Service 1GE client signal over ODU0 over ODU2 over ODU4 (Fixed DSR-ODU mapping, flexible ODU allocation)

6.2.1.2.3 Example: 1GE client signal over ODU0 over ODU2 over ODU4 (Fixed DSR-ODU mapping, flexible ODU allocation) with intermediate ODU0 switching.

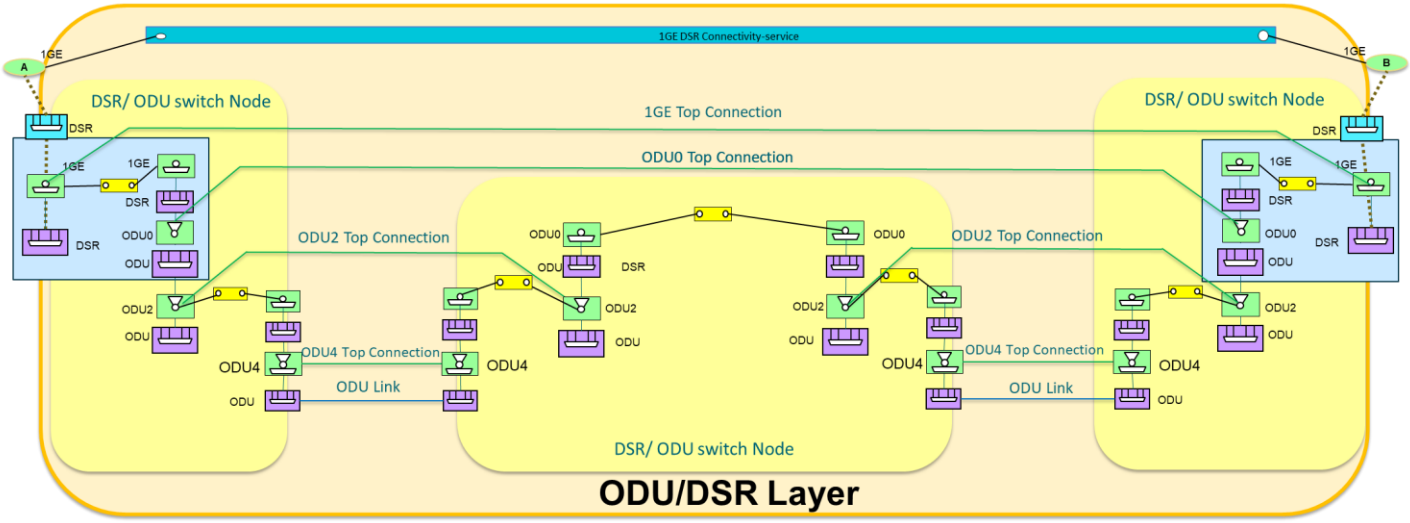


Figure 6-13 Connectivity Service 1GE client signal over ODU0 over ODU2 over ODU4 (Fixed DSR-ODU mapping, flexible ODU allocation) with intermediate ODU0 switching

6.2.1.2.4 Example: 10GE client signal over ODU2 over ODU4 (Fixed DSR-ODU mapping, flexible ODU allocation) with intermediate transponder regeneration.

In this example introduces the intermediate regeneration (3R) of the optical channel (OTSi) into account from the modelling perspective. Please consider the following:

The TAPI Server MAY or MAY NOT expose SIPs for the regeneration board OTSi/OTSiA:

a) The TAPI Server does not expose SIPs for the regeneration board OTSi/ OTSiA interfaces, thus these resources are only available to be consumed by the internal control plane for regeneration of a higher layer (ODU, DSR) client connectivity-service. Currently we assume this approach.
b) The TAPI Server does expose SIPs for the regeneration board OTSi/OTSiA interfaces. So, the TAPI Server exposes to the user the creation of the two OTSi/ OTSiA connectivity services segments independently, to be used by higher layer services. Moreover, these SIPs also can be used to constrain a HO-ODUk or higher layer connectivity-service to use this 3R point. In this case, the user MUST request a creation of a CS including (for a regenerated service with a single regeneration point) four CSEPs with references to aEND, zEND and intermediate regeneration SIPs, in the CS request.

Assuming the option a), the expected connectivity model result in terms of hierarchy of connections within the CS's connection list, SHALL include a single HO-ODUk supported by N-segments of OTSi/OTSiA connections represented as ODU links between every regeneration stage (in this example a single regeneration results into 2 OTSI/OTSiA segments).

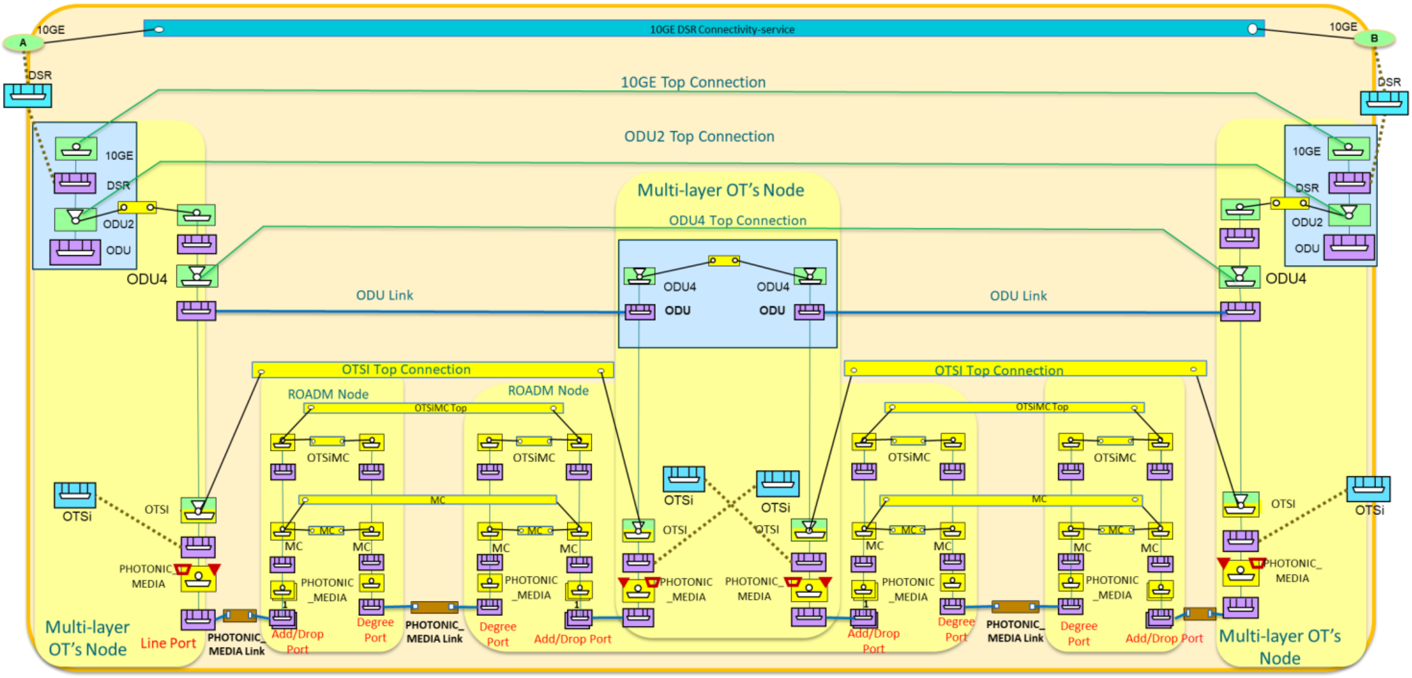


Figure 6-14 Connectivity Service 10GE client signal over ODU2 over ODU4 (Fixed DSR-ODU mapping, flexible ODU allocation) with intermediate transponder regeneration.

6.2.2 Use case 1a: Unconstrained DSR Service Provisioning single wavelength (=<100G).

Number	**UC1a**
Name	**Unconstrained DSR Service Provisioning single wavelength (=<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. **This service can include intermediate regeneration if necessary.** The underlying connection provisioning and management (including lower layer connections e.g., ODU, OTSi, MC, OTSiMC and intermediate regeneration connections if needed) is performed by the SDN Domain controller. The path of each lower layer connection (e.g., ODU or OCh/OTSi, OMS) across the network topology is calculated by the controller and the connections automatically provisioned. The TAPI-Client is not providing technology specific Traffic-Engineering constrains.
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

6.2.2.1 Relevant Parameters

Table 31: 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**
requested-capacity	“total-size”: {value: ,unit:} • "value": "[0-9]{8}", • "unit": ["TB", "TBPS", "GB", "GBPS", "MB", "MBPS", "KB", "KBPS"]	RW	C	• Provided by *tapi-client* • This is mandatory if there is no capacity in each of the CSEPs (it is assumed all CSEPs have the same capacity). • NOTE TAPI v2.3 includes the **layer-protocol-qualifier** so the requested-capacity MAY be omitted if there is no ambiguity.
service-type	"POINT_TO_POINT_CONNECTIVITY"	RW	O	• Provided by *tapi-client* • Current RIA version only considers P2P services.
service-layer	"DSR"	RW	M	• Provided by *tapi-client*

Table 32: 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**
layer-protocol-name	"DSR"	RW	M	• Provided by *tapi-client*
layer-protocol-qualifier	One qualifier derived from "DIGITAL_SIGNAL_TYPE"	RW	M	• Provided by tapi-server • All children identities defined for "DIGITAL_SIGNAL_TYPE" SHOULD be supported (depending on hardware capabilities).
direction	"BIDIRECTIONAL"	RW	O	• Provided by *tapi-client* • Default intended value is BIDIRECTIONAL
role	"SYMMETRIC"	RW	O	• Provided by *tapi-client* • Support only P2P and SYMMETRIC roles.
capacity	“total-size”: {value: unit} • "value": "[0-9]{8}", • "unit": ["TB", "TBPS", "GB", "GBPS", "MB", "MBPS", "KB", "KBPS"]	RW	C	• Provided by *tapi-client* • This is mandatory if there is no capacity in the connectivity service (CSEPs may not have the same capacity).

The ODU CEPs must include the FEC parameters specified in odu-connection-end-point-spec object definition (see below) if provided by the device.

```
augment /tapi-common:context/...owned-node-edge-point/tapi-connectivity:cep-list/tapi-
connectivity:connection-end-point:
    +--ro odu-connection-end-point-spec
       +--ro odu-common
…
       +--ro odu-term-and-adapter
       | …
       |  +--ro fec-parameters
       |  |  +--ro pre-fec-ber?           uint64
       |  |  +--ro post-fec-ber?          uint64
       |  |  +--ro corrected-bytes?       uint64
       |  |  +--ro corrected-bits?        uint64
       |  |  +--ro uncorrectable-bytes?   uint64
       |  |  +--ro uncorrectable-bits?    uint64
       |  +--ro odu-cn-effective-time-slot-list*   uint64
```

6.2.3 Use Case 1b: Unconstrained DSR Service Provisioning multi wavelength (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	The UC1 describes the provisioning of a ***tapi-connectivity:connectivity-service*** instance between SIPs exposed by the TAPI-Server at the DSR networking layer. **This service can include intermediate regeneration if necessary.** This UC follows UC1a, but with the difference that the service triggers the generation of an ODU-Cn top-level connection which, in turn, is realized by one or more OTSi connections.
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

6.2.3.1 Expected results

Please see the graphical description according to the Transitional-link and Multi-Layer models in Figure 6-15 and Figure 6-16 respectively. The two figures below show an example graphical context for a 200GE over ODUC2 connectivity-service over two OTSi/OTSiMC over a single physical (PHOTONIC-MEDIA_OMS) port.

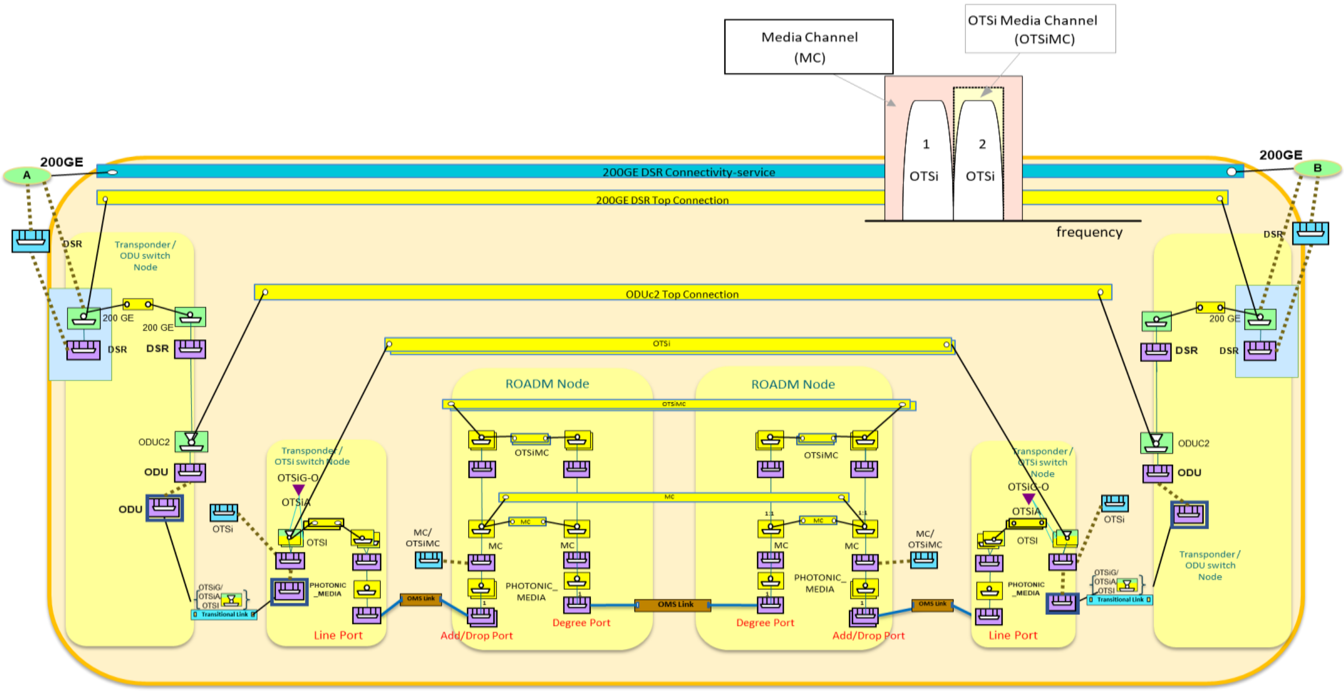


Figure 6-15 200GE over ODUC2 connectivity-service over two OTSi/OTSiMC over a single physical (PHOTONIC-MEDIA_OMS) port (Transitional Link).

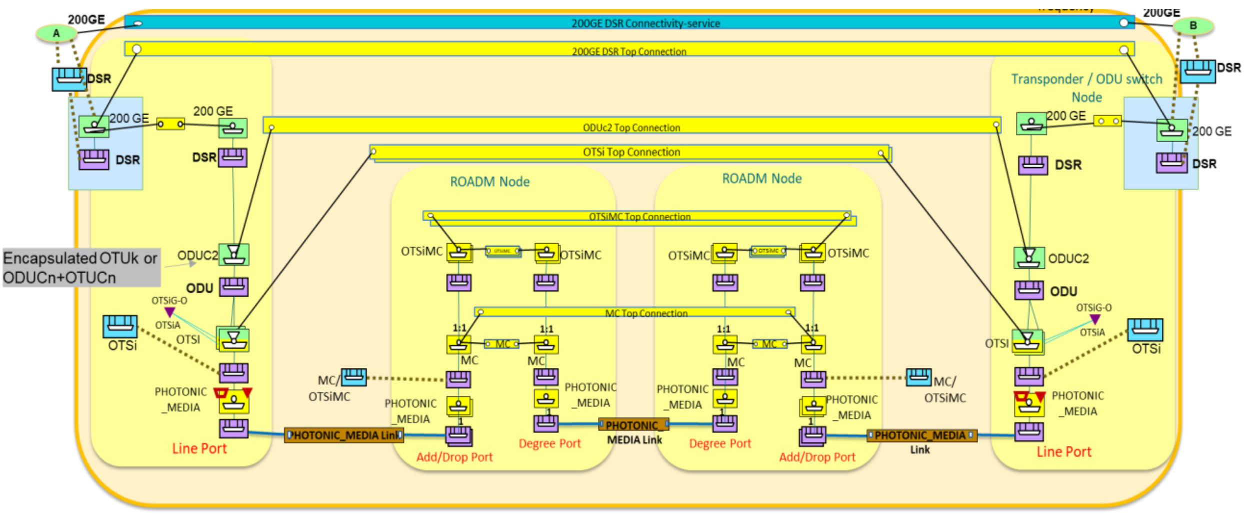


Figure 6-16 200GE over ODUC2 connectivity-service over two OTSi/OTSiMC over a single physical (PHOTONIC-MEDIA_OMS) port (Multi-Layer Node).

The model may include an OTSiA/OTSiG MEP for monitoring purposes, however the monitoring capabilities of the model will be described in a later release of this specification.

Note that the object **otsi-assembly-connection-end-point-spec** is marked as deprecated in TAPI v2.1.3 hence its use is no longer recommended.

6.2.4 Use case 1c: Unconstrained DSR over ODU Service Provisioning

Number	**UC1c**
Name	**Unconstrained 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 OT client ports, including the mapping and or multiplexing of such client signal into the line G.709 OTN frame. This UC MAY require the prior provisioning of a (unconstrained) OTSi / OTSiA service with or without channel selection. 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.
Layers involved	DSR/ODU
Type	Provisioning
Description & Workflow	This UC is implemented following the same workflow described in “Description & Workflow” of UC1.0 with [**server restrictions**]

6.2.4.1 Detailed Workflow

Two cases are considered: Case I (mapping) the ODUk CTP is directly carried by an OTUk TTP object or Case II (multiplexing) the ODUk CTP us carried by a server layer ODU TTP object. The OTUk object is not explicitly modeled in TAPI v2.1.3.

6.2.4.2 Connectivity Service request processing

The provisioning requires the TAPI client to send a connectivity service request with the involved Service Interface Points (SIPs) mapped to the DSR NEPs. The UUID of the connectivity service is provided by the client (`UUID_DSR_CS`). The workflow includes:

- The TAPI client requests the DSR connectivity service using the container **server-connectivity-service-end-point** of the endpoints list element in order to bind the ODU endpoints to the service, in addition to the use of the DSR endpoints.
- The use of 4 endpoints in the connectivity service with a clear client/service relationship by means of the local-id. To illustrate the use of >2 endpoints in a Connectivity service (only source endpoints shown) [**server restrictions**] see the following example:

```
module: tapi-connectivity
  augment /tapi-common:context:
    +--rw connectivity-context
       +--rw connectivity-service* [uuid]
       |  +--rw end-point* [local-id]
       |  |  +--rw layer-protocol-name
       |  |  +--rw layer-protocol-qualifier
       |  |  +--rw service-interface-point
       |  |  |  +--rw service-interface-point-uuid
       |  |  +--rw server-connectivity-service-end-point
       |  |  |  +--rw connectivity-service-uuid
       |  |  |  +--rw connectivity-service-end-point-local-id
       |  |  +--rw local-id
```

With the corresponding JSON

```
  "end-point" : [
     {
         "layer-protocol-name" : "DSR",
         "layer-protocol-qualifier" : <DSR-Layer-Qualifier,
         "service-interface-point" : {"service-interface-point-uuid" : "DSR_SIP"},
         "local-id" : 1,
         "server-connectivity-service-end-point" : {
               "connectivity-service-uuid" : <UUID_DSR_CS> this connectivity service (**)
               "local-id" :  "10"
         }
     },
     {
         "layer-protocol-name" : "ODU",
         "layer-protocol-qualifier" : "ODU-TYPE-ODUj",
         "service-interface-point" : {"service-interface-point-uuid" : "DSR_SIP"}, (**)
         "local-id" : "10",
         "odu-connectivity-service-end-point-spec" : {
           "odu-csep-common-pac" : {
               "odu-type" : "ODU-TYPE-ODUj",
           }
     },
...
   ],
}
```

(**) By agreement, the ODU CSEP will refer to DSR SIP, to indicate that the CSEP instance is used only to specify layer protocol constraints, and the same uuid of the DSR CS is used in ***server-connectivity-service-end-point/connectivity-service-uuid***. Note that Routing constraints can be specified through e.g., include Link or Connectivity Service/Connection coroute inclusion.

The processing depends on whether ODU multiplexing occurs or not. From the point of view of the client both cases are the same since the client does not specify (applicable) multiplexing (see UC2b for that).

```
augment ...connectivity-service/tapi-connectivity:end-point:
    +--rw odu-connectivity-service-end-point-spec
       +--rw odu-csep-ctp-pac
       |  +--rw tributary-port-number?   uint64
       |  +--rw tributary-slot-list*     uint64
       +--rw odu-csep-ttp-pac
       |  +--rw configured-mapping-type?   mapping-type
       |  +--ro configured-client-type?    tapi-dsr:digital-signal-type
       +--rw odu-csep-common-pac
          +--rw odu-type?                  odu-type
          +--rw odu-rate?                  uint64
          +--rw opu-tributary-slot-size?   odu-slot-size
          +--rw number-of-odu-c?           uint64
```

The client DSR interface is modeled as shown in the figure below. The client port has a DSR NEP, with a supported CEP qualifier (e.g., DIGITAL_SIGNAL_TYPE_10_GigE_WAN). Note that to represent maximum switching flexibility, a pair of DSR CEPs with the same qualifier MAY be instantiated with the second NEP on top of a framing ODU TTP CEP. [note: a simplified version is also shown within the figure, where the internal connection between DSR CEPs is not explicit].

NOTE: The ODU TTP MAY or MAY NOT be configurable. When such ODU TTP is not configurable, it must provide the mapping type and the client type.

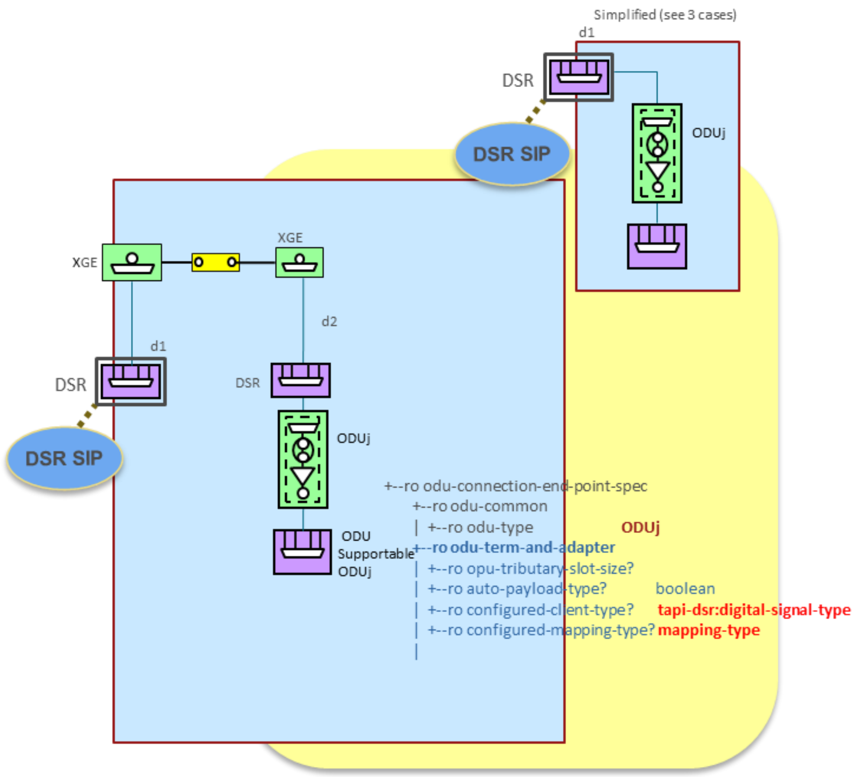


Figure 6-17 UC-1c Client port representation - initial state.

Such ODU TTP CEP MUST include the **tapi-odu:odu-connection-end-point-spec** augment (see below), including the odu-type and the configured-client-type and mapping-type, obtained from the logical channels discovered from the client port logical channel assignments.

```
+--ro odu-connection-end-point-spec
       +--ro odu-common
       |  +--ro odu-type?              odu-type
       |  +--ro odu-rate?              uint64
       |  +--ro odu-rate-tolerance?    uint64
       |  +--ro number-of-odu-c?       uint64
       +--ro odu-term-and-adapter
       |  +--ro opu-tributary-slot-size?           odu-slot-size
       |  +--ro auto-payload-type?                 boolean
       |  +--ro configured-client-type?            tapi-dsr:digital-signal-type
       |  +--ro configured-mapping-type?           mapping-type
       |  +--ro accepted-payload-type
       |  |  +--ro named-payload-type?   odu-named-payload-type
       |  |  +--ro hex-payload-type?     uint64
       |  +--ro fec-parameters
       |  |  +--ro pre-fec-ber?           uint64
       |  |  +--ro post-fec-ber?          uint64
       |  |  +--ro corrected-bytes?       uint64
       |  |  +--ro corrected-bits?        uint64
       |  |  +--ro uncorrectable-bytes?   uint64
       |  |  +--ro uncorrectable-bits?    uint64
```

The transponder line port at the ODU layer is modeled as a ODU NEP that has a supported qualifier corresponding to the ODU-k or ODU-Cn -- see SIP ODU A (line) in the figure --. Such ODU NEP relies on one or more OTSi CEPs that transport the corresponding OTUk or OTU-Cn signal. Note that commonly the flexibility at OTSi layer is not available, i.e., the OTSi CEPs are directly supported by line NEP.

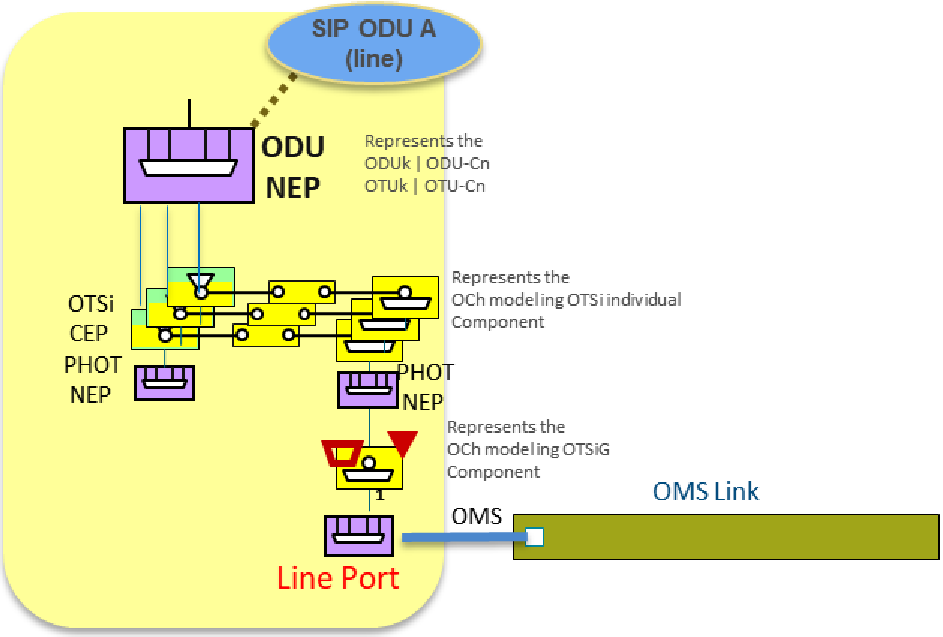


Figure 6-18 UC-1c TAPI context reflecting the line side ODU NEP and SIP over an OTSiA

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 OT line ports. An instance of the ODU Connectivity Service for the ODUk (or ODUCn) is not required, but a TTP ODU CEP representing the ODUk (or ODUCn) connection MUST be instantiated over the ODU NEP. Such CEP must contain the odu-connection-end-point-spec including the following:

a. For an ODUk the **odu-common/odu-type** MUST be ODU_TYPE_ODUk

b. For an ODUCn the **odu-common/odu-type** MUST be ODU_TYPE_ODU_CN and the number-of-odu MUST reflect the number of ODUC slices that constitute the signal. The **odu-cn-effective-time-slot-list** MUST list the ODU-Cn 5GHz available slots.

The resulting view that corresponds to the TAPI context once both client port and line port have been discovered and mapped is shown in the figure below.

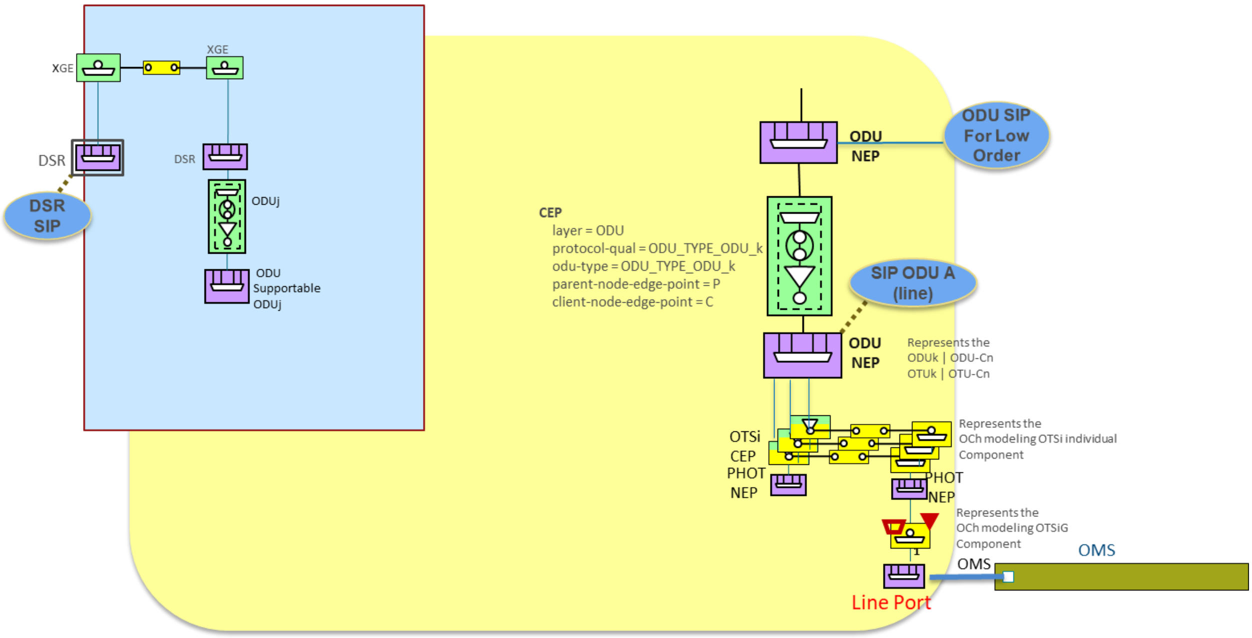


Figure 6-19 UC-1c TAPI context representation of the discovered DSR/ODU multi-layer node CLIENT/LINE ports.

6.2.4.3 Relevant Parameters

Table 33: ODU connectivity-service-end-point (**ODU CSEP**) object definition (UC1c)

ODU CSEP	**/tapi-common:context/tapi-connectivity:connectivity-context/tapi-connectivity:connectivity-service/tapi-connectivity:end-point/tapi-odu:odu-connectivity-service-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
odu-csep-ttp-pac	configured-mapping-type configured-client-type as per tapi-odu.	RW	C	• Provided by *tapi-client.* If the TTP is pre-configured, the TTP aspects are already represented in the TTP CEP that is associated to the client port, that maps the client signal. If it is not, the mapping and client type must be specified.
odu-csep-ctp-pac	NOT USED	N/A		Not used. The TAPI client SHALL NOT specify the (allocated) tributary slot list as defined in the odu-csep-ctp-pac container data node of the odu-connectivity-service-end-point-spec CSEP augment.

odu-csep-common-pac / odu-type	One of [ODU_TYPE_ODU1, 2, 2E, 3, 4, FLEX]	RW	M	• Provided by *tapi-client.* As per the ODUj used
odu-csep-common-pac / odu-rate	ODU rate in Kbit/s. See notes	RW	O	• Provided by *tapi-client.* Indicates the rate of the ODU termination point in Kbits/s. Valid values shall be consistent with the oduType configuration as shown in Table 7-2/G.709 v5. Setting this value for fixed-rate ODUk and ODUCn types (e.g., ODU0), is optional. The default value is derived from the configured oduType, as defined in Table 7-2/G.709 v5.

Table 34: ODU odu-connection-end-point-spec (**ODU CEP**) object definition (UC1c)

ODU CEP	**/tapi-common:context/…/tapi-connectivity:connection-end-point/photonic-media:odu-connection-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
odu-common / odu-type	odu-type: One of [ODU_TYPE_ODU2, 3, 4, _CN]	RO	M	Provided by *tapi-server*
odu-cn-effective-time-slot-list	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).	RO	M	Provided by *tapi-server*

- The SDN Controller SHALL select the tributary slot and tributary port number (for multiplexing), [Note: see UC2b for channel selection, allowing the TAPI client to specify the TS# and TPN]

6.2.4.4 Expected results

After the successful provisioning of the service, the expected results are as follows:

6.2.4.4.1 Case I (mapping) the ODU CTP is directly carried by an OTU TTP object

At the SDN-C TAPI context level:

- The TAPI server has allocated the NEPs and CEPs corresponding to the mapping of a DSR signal into a HO-ODU which is cross-connected to the line side HO-ODU. Internal connections reflect the DSR connections and the ODU connections (see figure below)
- The TAPI server has allocated the TAPI connection objects, with the corresponding CEPs that appear in the ODU NEPs.
- The TAPI server has allocated the DSR Connectivity Service with the requested parameters.

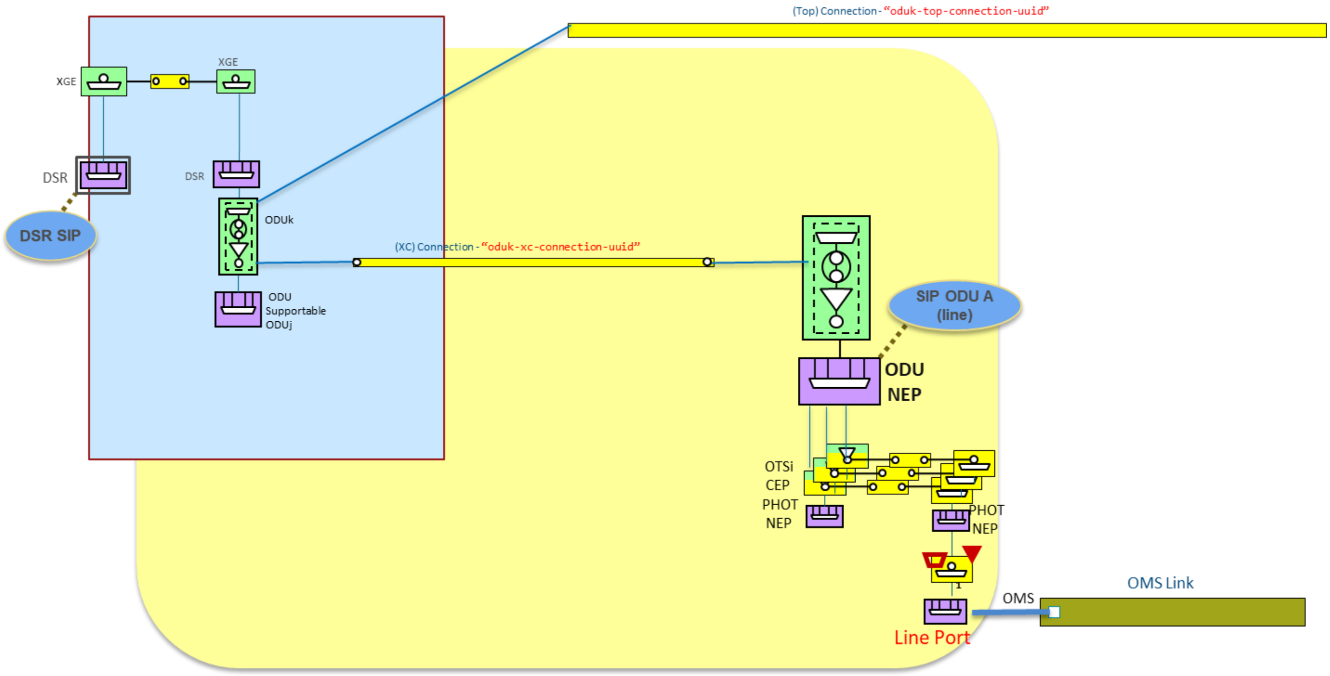


Figure 6-20 UC1c: TAPI context representation of DSR direct mapping over HO-ODUk (ODUj = ODUk)

6.2.4.4.2 Case II (multiplexing) the ODU CTP us carried by a server layer ODU TTP object

At the SDN-C TAPI context level:

- The TAPI server has allocated the NEPs and CEPs corresponding to the mapping of a DSR signal into a LO-ODU which is multiplexed into an HO-ODU. Internal connections reflect the DSR connections and the ODU connections (see figure below)
- The TAPI server has allocated the TAPI connection objects, with the corresponding CEPs that appear in the ODU NEPs, both for the low order and high order.
- The TAPI server has allocated the DSR Connectivity Service with the requested parameters.

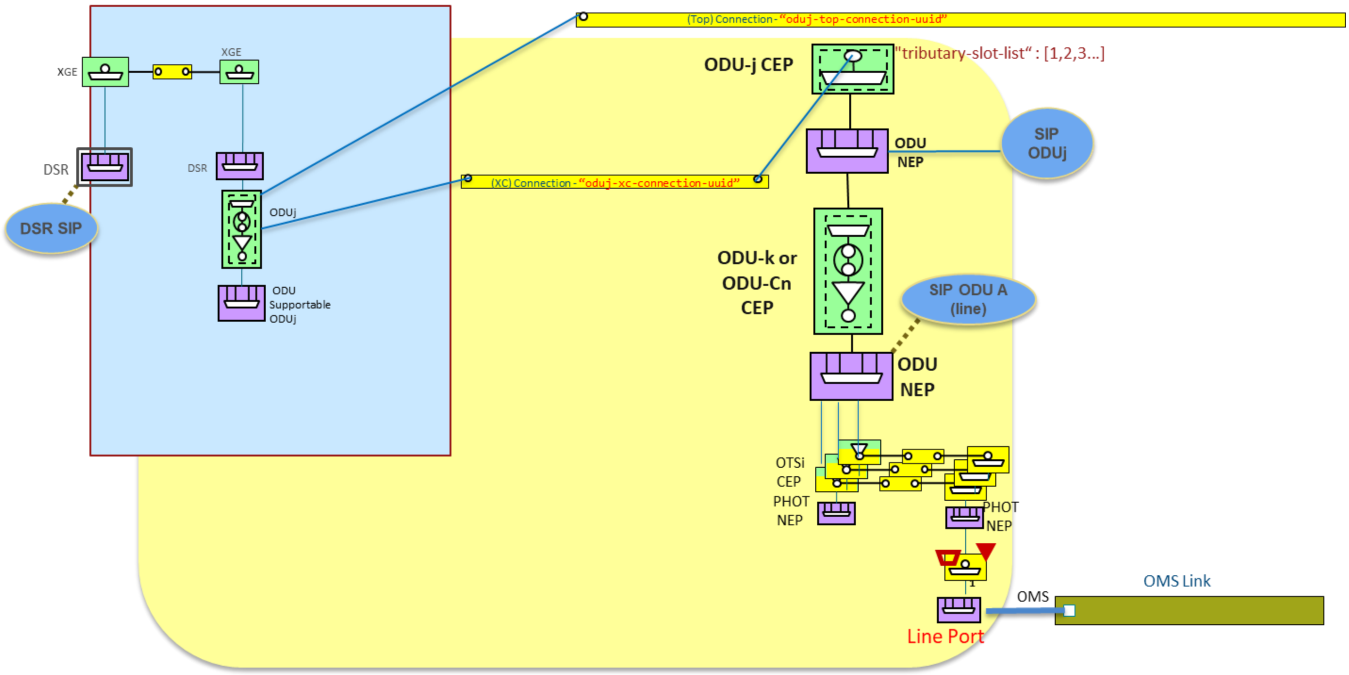


Figure 6-21 UC1c: TAPI context representation of DSR mapping over LO-ODUj multiplexed over HO-ODUk

6.2.5 Use case 1d: Unconstrained PHOTONIC_MEDIA/OTSi Service Provisioning

Number	**UC1d**
Name	**Unconstrained PHOTONIC_MEDIA/OTSi Service Provisioning**
Technologies involved	Optical
Process/Areas Involved	Planning and Operations
Brief description	The UC1d describes the provisioning of a ***tapi-connectivity:connectivity-service*** instance between SIPs exposed by the TAPI-Server at the PHOTONIC_MEDIA networking layer. The underlying connection provisioning and management (including lower layer connections e.g., OTS/OMS, OTSiMC, MC including intermediate regeneration connections if needed) is performed by the SDN Domain controller. The path of each lower layer connection (e.g., OTS/OMS, OTSiMC, MC) across the network topology is calculated by the controller and the connection automatically provisioned. Moreover, the TAPI-Client is not providing technology specific Traffic-Engineering constrains such as optical-spectrum selection for the routing of OTSi connections.
Layers involved	PHOTONIC_MEDIA
Type	Provisioning
Description & Workflow	This UC is implemented following the same workflow described in “Description & Workflow” of UC1.0

6.2.5.1 Relevant Parameters

No specific OTSiA CSEP parameters are required. Note that UC2a allows channel selection. Relevant OTSi CEP parameters are provided in
Table 24: otsi-connection-end-point-spec (**CEP**) object definition with UC1.0.

6.2.5.2 Expected results

This use case requires the relevant SIPs attached to the corresponding OTSI NEPs are available and exposed by the TAPI server. The connection generation MUST follows the rules detailed in section 5.1.1., and expected results depend on whether a transitional link is used or not, as shown next.

6.2.5.2.1 Example: OTSi Single Wavelength Connectivity service provisioning [Transitional link approach]

In this case, it is assumed that the ODU and PHOTONIC_MEDIA layers are connected through a transitional link. Please see the detail graphical description in Figure 6-22:

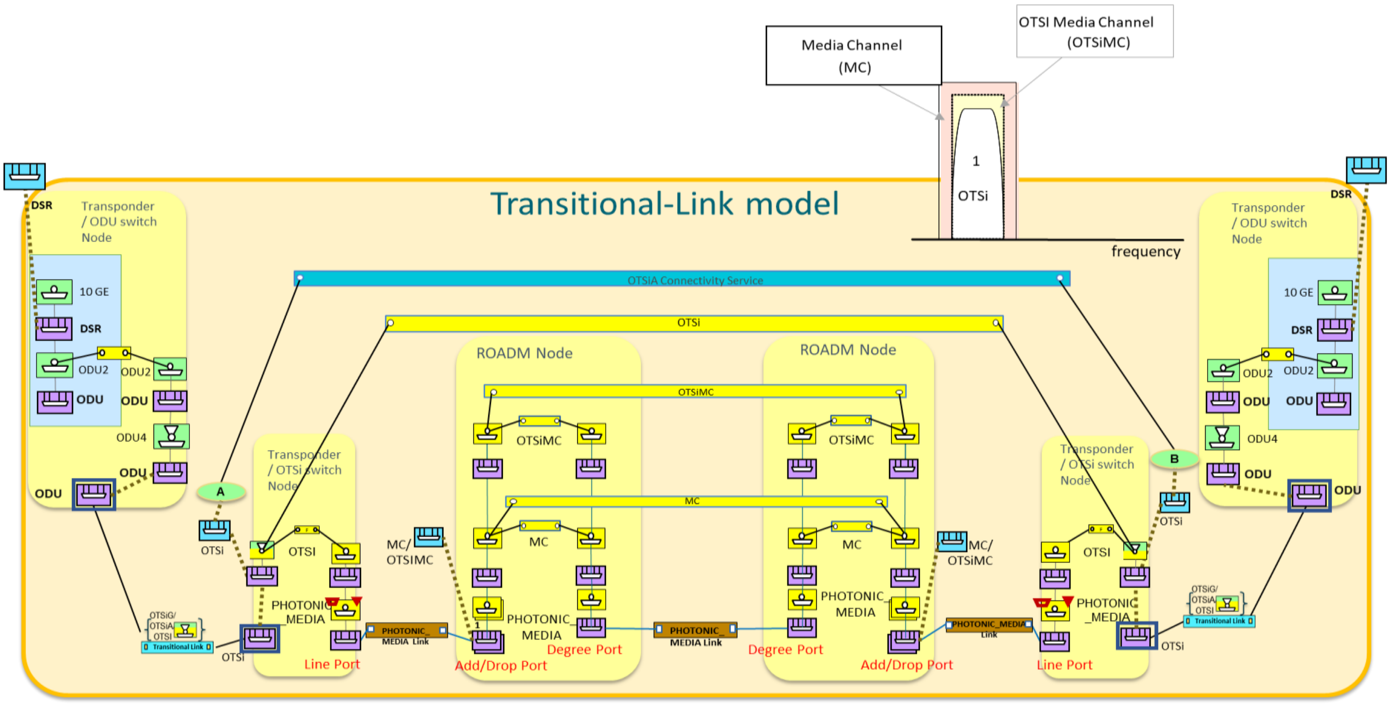


Figure 6-22 OTSi single lambda connectivity-service - Transitional link model.

6.2.5.2.2 Example: OTSi Single Wavelength Connectivity service provisioning [Multi-layer node approach]

In this case, it is assumed that originally, the ODU and PHOTONIC_MEDIA layers integrated in a Multi-layer Node. Please see the detail graphical description in Figure 6-23:

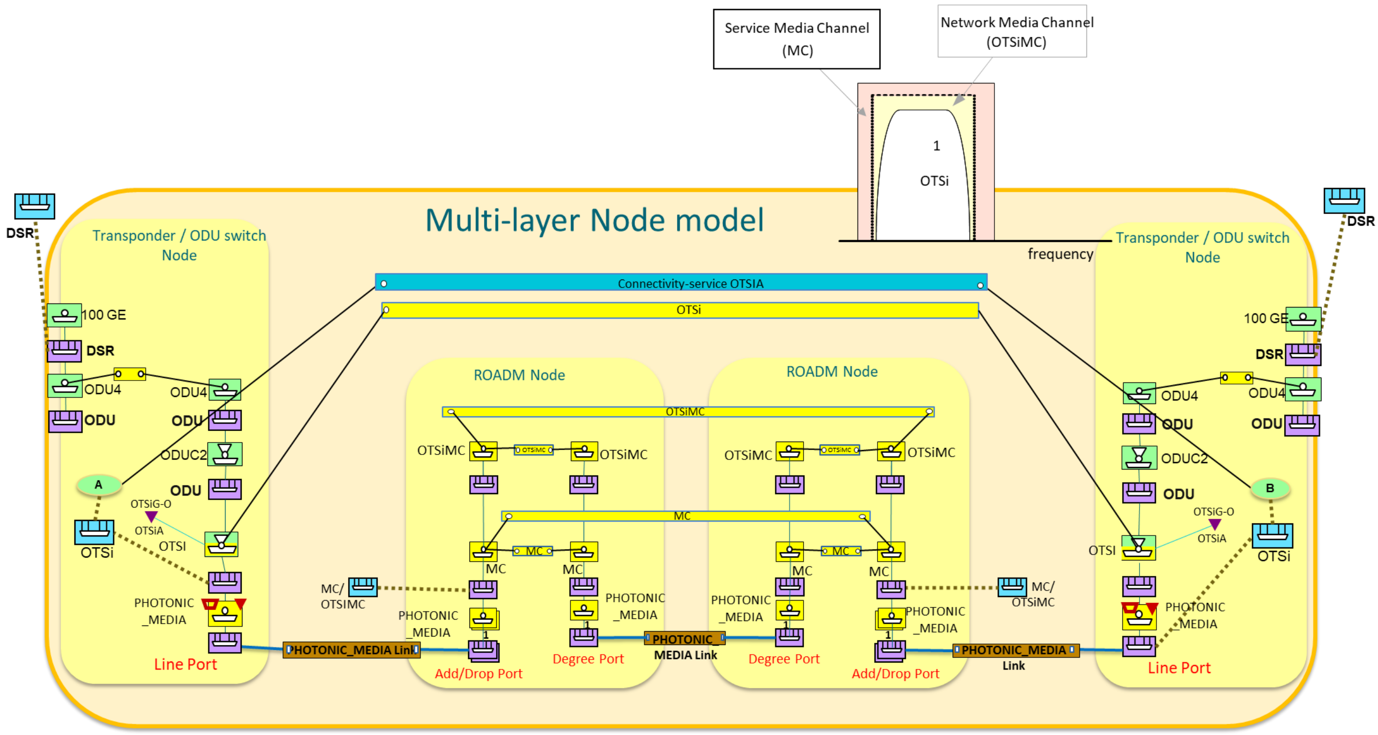


Figure 6-23 OTSi single lambda connectivity-service - Multi-layer node approach.

6.2.6 Use case 1e: Unconstrained PHOTONIC_MEDIA/OTSiA Service Provisioning

Number	**UC1e**
Name	**Unconstrained 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 with OTSiA qualifier. The TAPI-Client is not providing technology specific Traffic-Engineering constraints such as optical-spectrum selection for the routing of OTSi connections. **Note:** the number of OTSi components is assumed known and implicit.
Layers involved	PHOTONIC_MEDIA
Type	Provisioning
Description & Workflow	This UC is implemented following the same workflow described in "Description & Workflow" of UC1.0

6.2.6.1 Relevant Parameters

No specific OTSiA CSEP parameters are required. Note that UC2a allows channel(s) selection. Relevant OTSi CEP parameters are provided in
Table 24: otsi-connection-end-point-spec (**CEP**) object definition with UC1.0.

6.2.6.2 Expected results

This use case requires the relevant SIPs attached to the corresponding OTSi NEPs are available and exposed by the TAPI server. The connection generation follows the rules detailed in section 5.1.1. This case requires the generation of N number of OTSi Top Connections required to transport the service. N Top OTSi Connections are thus generated over the same parent NEP (which only includes PHOTONIC_LAYER_QUALIFIER_OTSi within its ***supported-layer-qualifier*** list). Please see the detail graphical description in Figure 6-24 and Figure 6-25.

6.2.6.2.1 Example: OTSiA multi-wavelength connectivity-service provisioning [Transitional link approach]

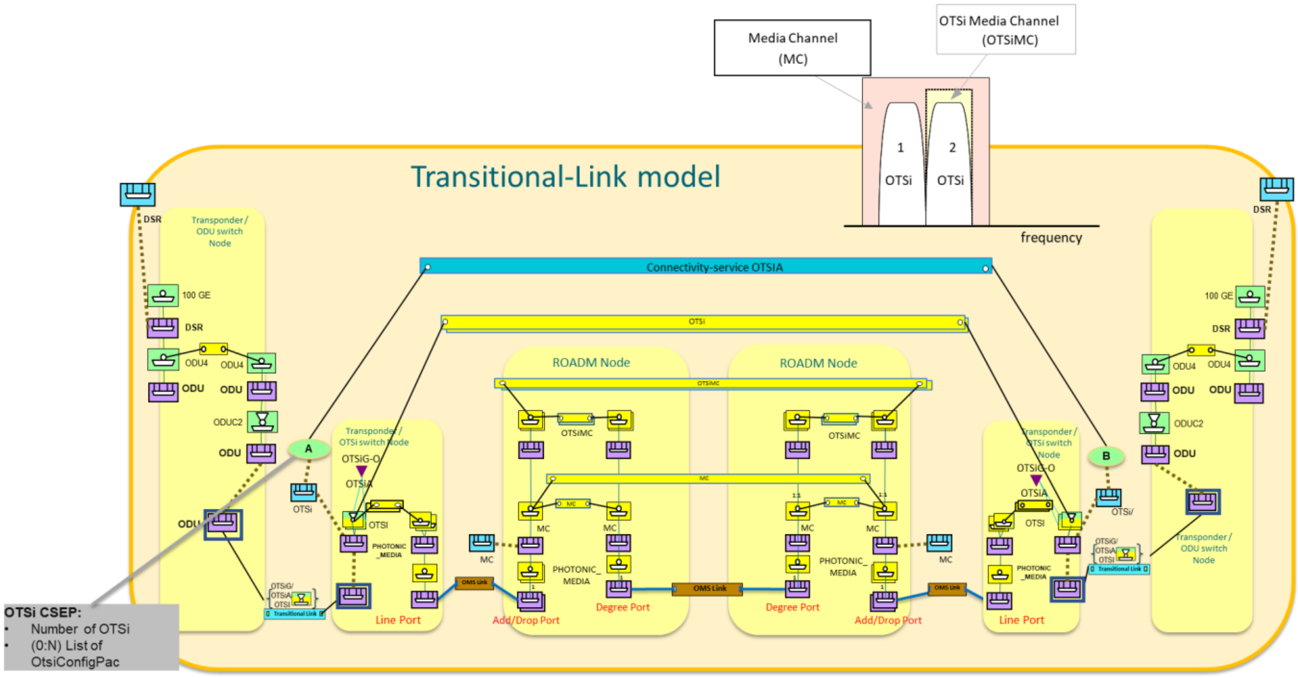


Figure 6-24 OTSiA multi-wavelength connectivity-service (transitional link model abstraction).

6.2.6.2.2 Example: OTSiA multi-wavelength connectivity-service provisioning [Multi-layer node approach]

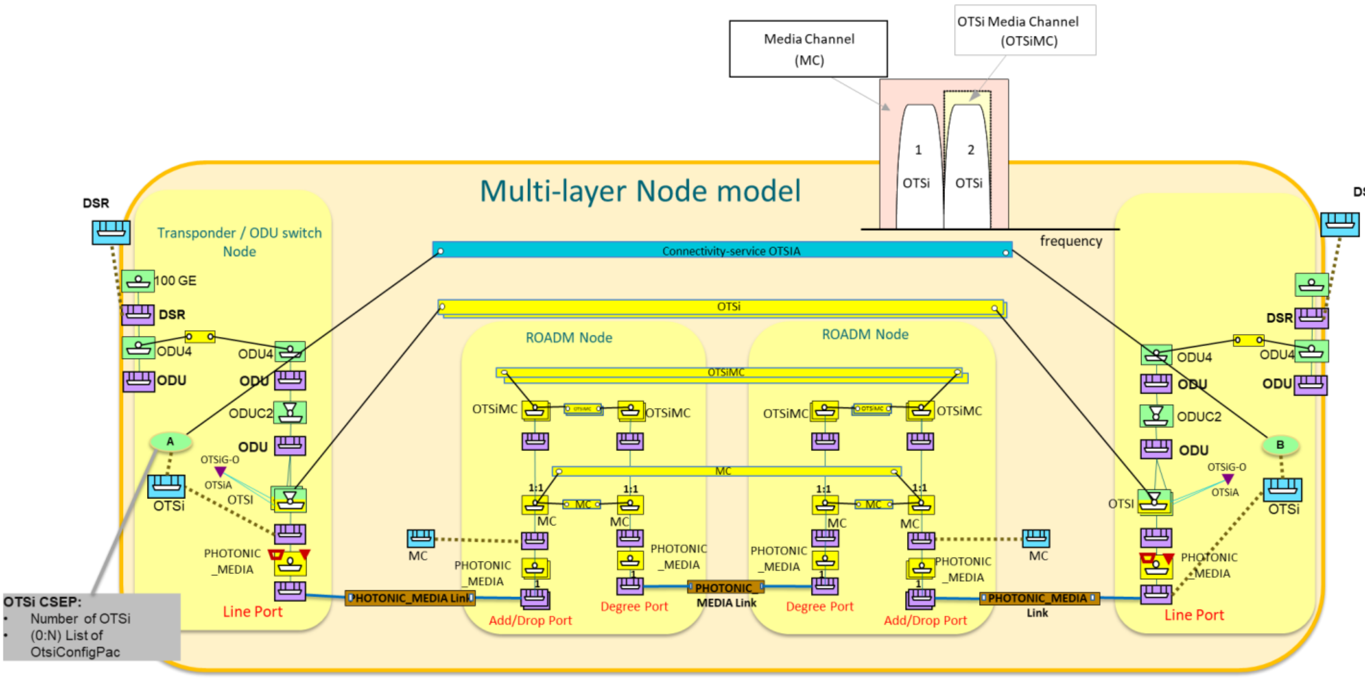
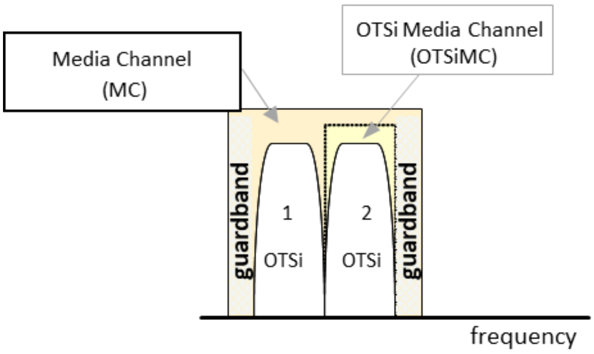


Figure 6-25 OTSiA multi-wavelength connectivity-service (multi-layer node model abstraction).

6.2.7 Use case 1f: Unconstrained PHOTONIC_LAYER_QUALIFER_MC Service Provisioning

Number	**UC1f**
Name	**Unconstrained PHOTONIC_LAYER_QUALIFER_MC Service Provisioning**
Technologies involved	Optical
Process/Areas Involved	Planning and Operations
Brief description	The UC1f describes the provisioning of a ***tapi-connectivity:connectivity-service*** instance between SIPs with the PHOTONIC_LAYER_QUALIFER_MC qualifier. **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. This MC is generally wider than the OTSi(A) occupied spectrum (for example, due to guard bands). Multiple OTSi signals MAY be included in the MC:  The TAPI-Client is not providing technology specific Traffic-Engineering constrains such spectrum-band selection for the MC connections. The Reconfigurable Optical Add Drop Multiplexers (ROADMs) Add/Drop ports MUST be represented as UNI interfaces with associated SIPs. The UNI interfaces MUST be represented according to one of the following alternatives: • Bidirectional UNI representation. • Unidirectional UNI representation.
Layers involved	PHOTONIC_MEDIA
Type	Provisioning
Description & Workflow	This UC is implemented following the same workflow described in “Description & Workflow” of UC1.0

6.2.7.1 Relevant Parameters

No specific MC CSEP parameters are required. Note that UC2c allows spectrum selection. Relevant MC CEP parameters are provided in Table 26: Media-channel-connection-end-point-spec (**MC CEP**) object definition with UC1.0.

6.2.7.2 Expected results

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

6.2.7.2.1 Model 1: Full Bidirectional - UNI and PHOTONIC_MEDIA Topology

This choice corresponds to a solution exposed by the TAPI server where the relation between the Add/Drop directions of UNI interfaces is known by the TAPI server and the unidirectionality of the rest of the PHOTONIC_MEDIA layer is abstracted as a full-bidirectional topology.

In this approach the MC UNI interfaces are represented as bidirectional SIPs associated to Add/Drop PHOTONIC_MEDIA NEPs. MC Connectivity-Services are bidirectional too with a single bidirectional Top Connection representing the end-to-end route across the PHOTONIC_MEDIA layer.

Moreover, the MC Top Connection includes within the ***tapi-connectivity:lower-connection*** attribute, the reference to the bidirectional Cross-Connections (XCs) between the bidirectional PHOTONIC_LAYER_QUALIFIER_MC CEPs over the bidirectional MC NEPs.

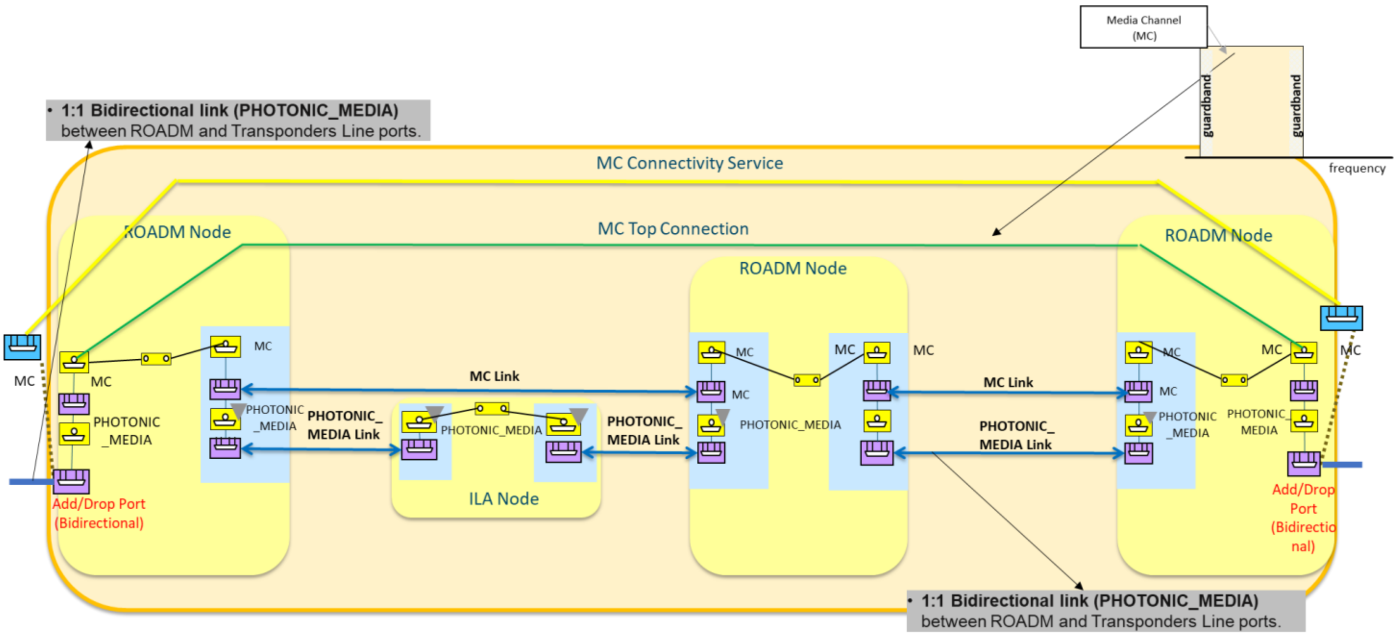


Figure 6-26 Full Bidirectional - UNI and OMS bidirectional scenario.

6.2.7.2.2 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.

It is assumed that unidirectional MC links are in place.

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 to the two point-to-multipoint 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.

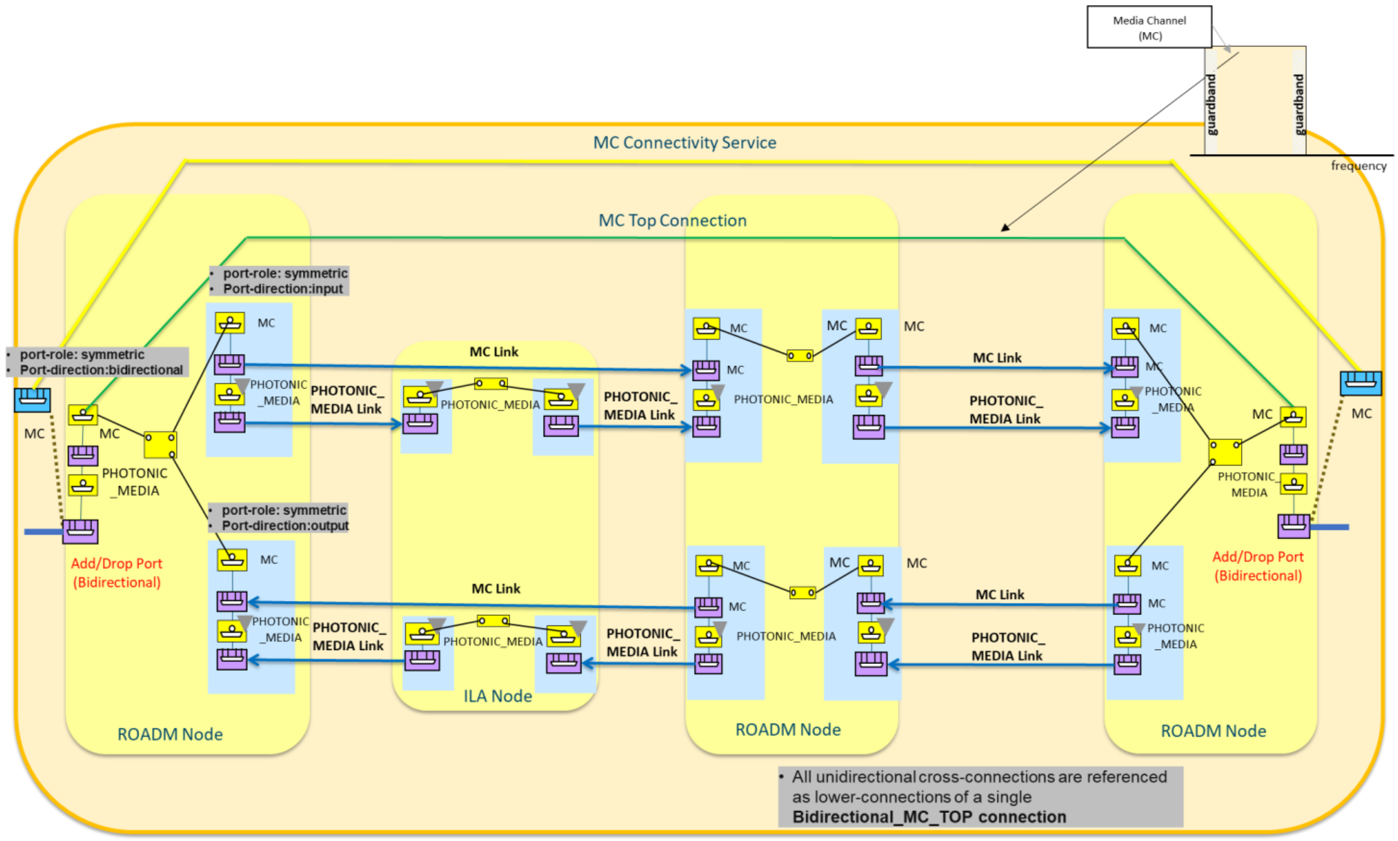


Figure 6-27 Mixed Scenario - UNI bidirectional and OMS unidirectional.

6.2.7.2.3 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 OT line ports (UC not described in this RIA[6]) *or* the OTs are not managed/controlled by the TAPI server (hence the relationship between OT's Line and OLS Add/Drop ports is also unknow by the TAPI server).

In this modelling approach the MC UNI interfaces are represented as unidirectional SIPs associated to unidirectional Add/Drop OTS/OMS NEPs.

MC Connectivity-services are bidirectional with four CSEPs (thus including four references to the unidirectional Add/Drop SIPs). 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.

[6] For example, once the TAPI Client correlates the PHOTONIC_MEDIA NEPs of OT's Line ports (supporting OTSI layer-protocol-qualifier CEPs) and Open Line System (OLS) Add/Drop MC, it exposes the connectivity through a 1:2 asymmetric ***tapi-topology:link*** relationship at the PHOTONIC_MEDIA layer. For further study

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.

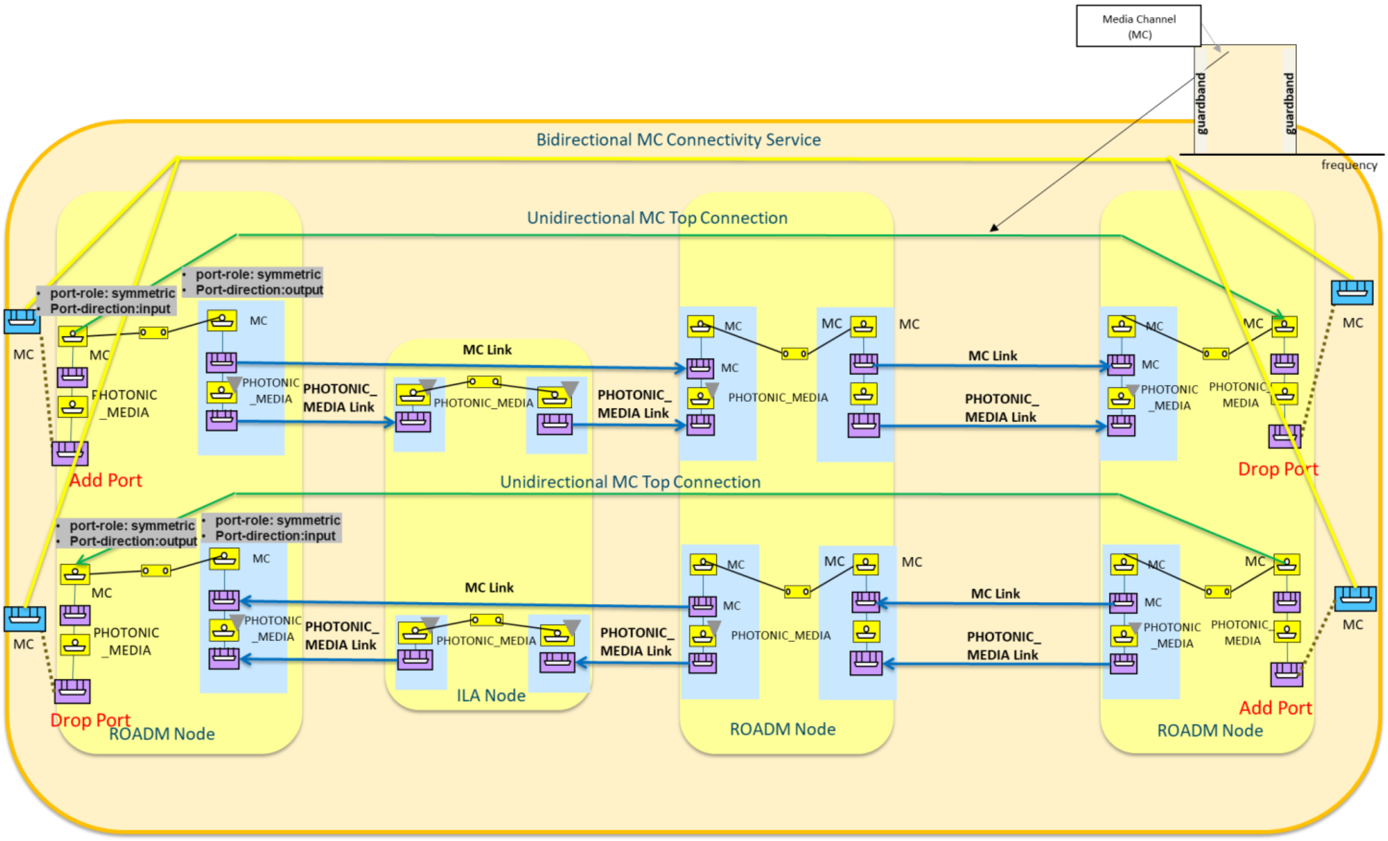
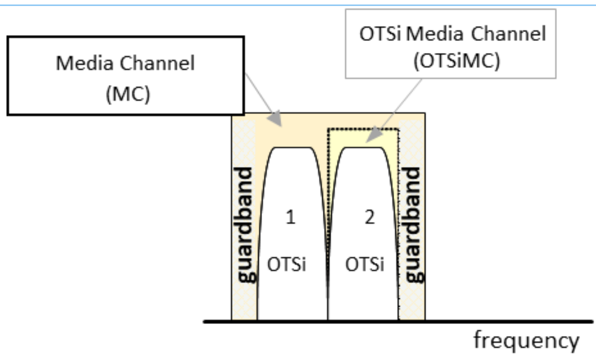


Figure 6-28 Full Unidirectional - UNI and OMS unidirectional scenario.

6.2.8 Use case 1g: Unconstrained PHOTONIC_MEDIA/OTSiMC over MC Service Provisioning

Number	**UC1g**
Name	**Unconstrained PHOTONIC_MEDIA/OTSiMC over 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 spectrums/channels occupied by the OTSi signals through the specification of the PHOTONIC_LAYER_QUALIFER_OTSiMC connectivity-service constraints. 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**] Notes: • The server layer restrictions are specified within the *additional* CSEP(s) with MC/MCA related constraints such as MC spectrum(s). • TAPI 2.1.3 does not support OTSiMCA specific class augmenting CSEP. The McaConnectivityServiceEndPointSpec is reused, augmenting the CSEP with layerProtocolQualifier = OTSiMCA. This allows the specification of the OTSiMC spectrum(s), through MediaChannelConfigPac(s). The connectivity-service request send by the TAPI Client MUST contain at least the end-point specification for both the **PHOTONIC_LAYER_QUALIFER_OTSiMC** and the **PHOTONIC_LAYER_QUALIFER_MC.**

6.2.8.1 Relevant Parameters

Table 35: Connectivity-service-end-point server-connectivity-end-point constraint definition

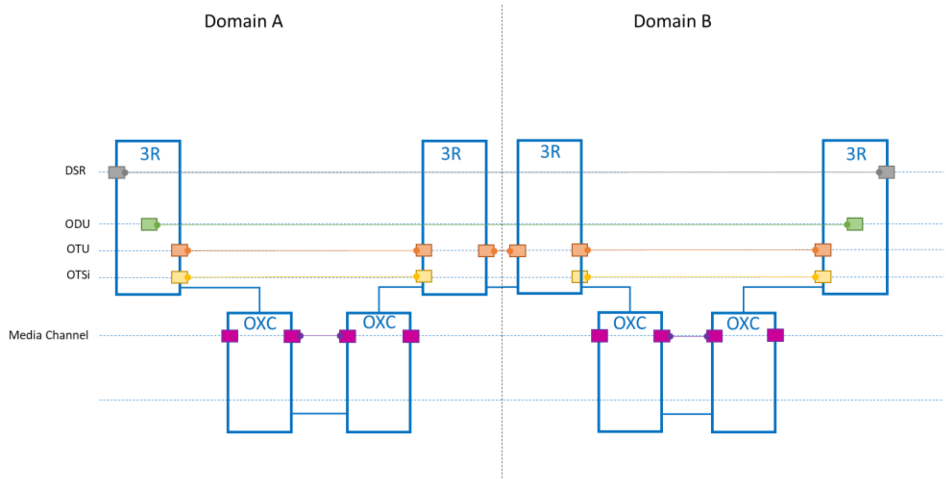
CSEP	**/tapi-common:context/tapi-connectivity:connectivity-context/tapi-connectivity:connectivity-service/tapi-connectivity:end-point**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
server-connectivity-service-end-point	**connectivity-service-end-point ref** -> - {/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/uuid , /tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/end-point/uuid	RW	M	Provided by tapi-client

Relevant OTSiMC and MC CEP parameters (i.e., both layers) are provided in Table 26: Media-channel-connection-end-point-spec (**MC CEP**) object definition with UC1.0.

6.2.8.2 Expected results

This use case accepts different variations according to the model directionality chosen to represent the PHOTONIC_MEDIA layer as in UC1f. For a full description of the model guidelines please see Section 6.2.7.2.

6.2.9 Use case 1h: Unconstr. asymmetric DSR Service Provisioning, DSR UNI to OTUk E-NNI grey interface.

Number	**UC1h**
Name	Unconstrained 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 connectivity-service between UNI and E-NNI CSEPs. Note that the proposed definition for these two interfaces (UNI, E-NNI) is to be further consolidated in TAPI, a possible solution is to include MEF extensions to augment current SIP definitions. The intention is to establish services which starts in one network domain and handover to another network domain managed by a different (TAPI) Server:  UNI side, the value for the attribute layer-protocol-qualifier in this case is typically a DIGITAL_SIGNAL_TYPE base qualifier. The capacity (**tapi-connectivity:connectivity-service/end-point/capacity**) of the port and the protocol qualifier (**tapi-connectivity:connectivity-service/end-point/layer-protocol-qualifer**) MUST be provided by the TAPI client (e.g., 100_GigE, 10_GigE or STM-16) as part of the CSEP definition. 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/ODU
Type	Provisioning

Description & Workflow	This UC is implemented following the same workflow described in "Description & Workflow" of UC1.0

6.2.9.1 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.

6.2.9.2 Connectivity Service request processing

The connectivity-service request shall include the CSEPs definition which shall point to the UNI (DSR) and E-NNI SIPs and specify the **layer-protocol-qualifier** for the UNI and E-NNI endpoints. The proposed connectivity-service provisioning request should be according to the following example:

```
{
"tapi-connectivity:connectivity-service": [
        {
            "end-point": [
                {
                    "direction": "BIDIRECTIONAL",
                    "role": "SYMMETRIC",
                    "layer-protocol-qualifier": "tapi-dsr:DIGITAL_SIGNAL_TYPE_10_GigE",
                    "layer-protocol-name": "DSR",
                    "local-id": "end_point_1",
                    "service-interface-point": {
                        "service-interface-point-uuid":  "UUID_X1" (DSR SIP)
                    }
                },
                {
                    "direction": "BIDIRECTIONAL",
                    "role": "SYMMETRIC",
                    "layer-protocol-qualifier": "tapi-odu:ODU_TYPE_ODU2", (*)
                    "layer-protocol-name": "ODU",
                    "local-id": "end_point_2",
                    "service-interface-point": {
                        "service-interface-point-uuid": "UUID_Y1"(OTN SIP)
                    }
                }
            ],
            "service-layer": "DSR",
            "uuid": "UUID_CS",
        }
    ]
}
```

(*) The E-NNI CSEP Protocol/Layer Qualifier can be ODU2 or DSR, the asymmetric nature of the connection being indicated by the OTN SIP.

6.2.9.3 Expected results

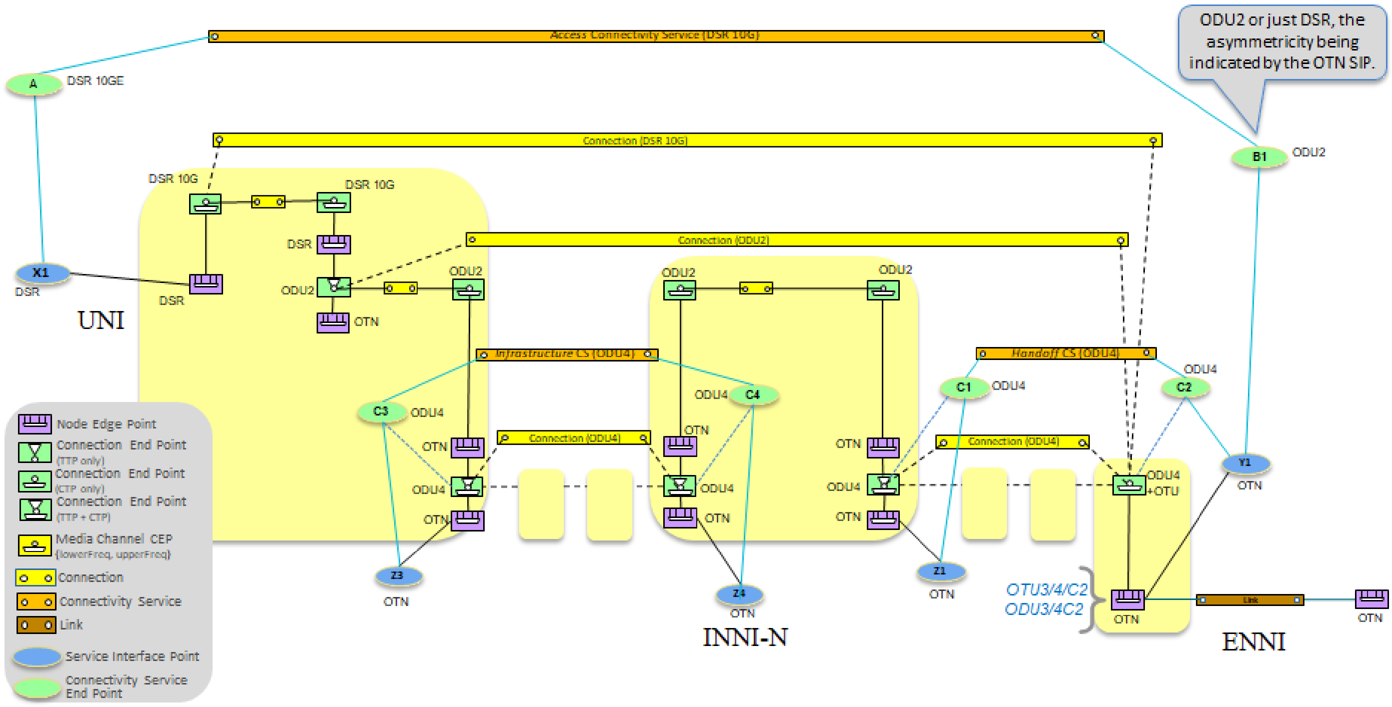


Figure 6-29 Asymmetric scenario.

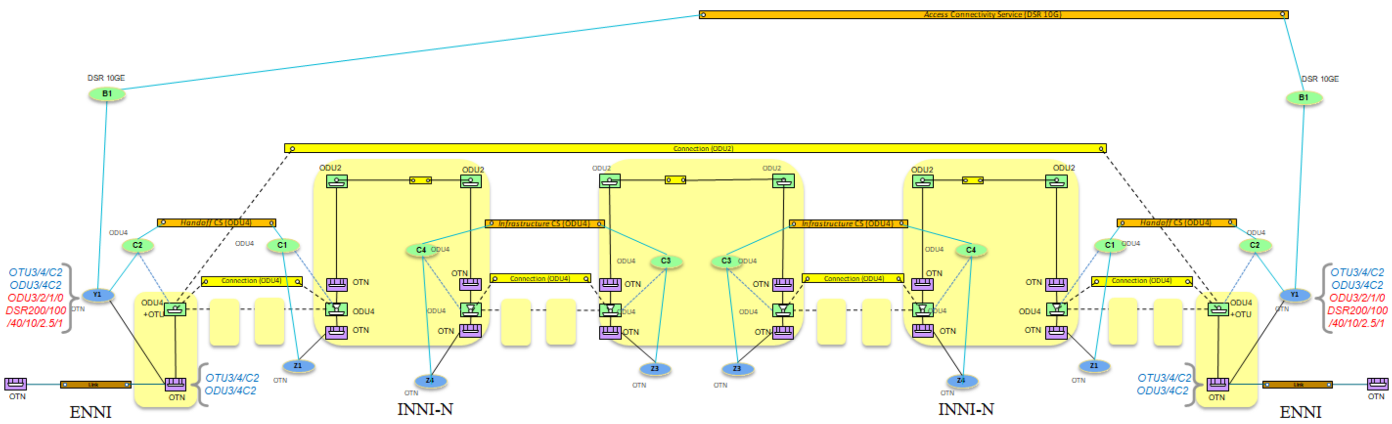


Figure 6-30 Asymmetric use case: TAPI Symmetric transit scenario with client monitoring.

Following rules apply:

- In the "access scenario", the "semi-terminated" Top Connection lists the CEP where the Connection is terminated (UNI side) and the available server layer CEP at ENNI side ("end point projection") where the connection is delivered within a server container. Doing so the Top Connection provides the best topological information. The ENNI CEP is intended as the point in the topology where the Connection is delivered to the external domain. The ENNI CEP, e.g., the ODU4 CEP, is the one which supports the Connection bandwidth (ODU2 time slot).

- In the transit scenario, the Connectivity Service (and its CSEPs) could be specified at any layer protocol name/qualifier, 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. Transit scenario, the "unterminated" Top Connection shall be represented if there is at least one monitoring point in the transit managed domain.

- Note that the ENNI CSEP could indicate the DSR layer qualifier, because CSEP is just providing intent information regarding the portion of (DSR) bandwidth that the ENNI SIP/NEP shall support.

- Note that ODU related information (e.g., preferred time slot) can be provisioned as server layer constraint.

6.2.10 Use case 2a: Unconstr. PHOTONIC_MEDIA/OTSi or OTSiA Service Provisioning with channel selection

Number	**UC2a**
Name	Unconstrained PHOTONIC_MEDIA/OTSi or OTSiA Service Provisioning with channel selection
Technologies involved	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, for the PHOTONIC_MEDIA_OTSI connectivity-services. The UC relies on the **tapi-photonic-media:otsia-connectivity-service-end-point** augmentation (the **tapi-photonic-media:otsi-connectivity-service-end-point** is deprecated in favor of otsia also in the case of n=1**).** of **tapi-connectivity:connectivity-service/end-point** objects. While the UC enables the direct provisioning of an OTSiA service, note that TAPI 2.1.3 does not specify OTU parameters. As a consequence, such parameters can be specified as value/pair parameters of the CS. TAPI 2.3 implementations will consider the use of otu-connectivity-service-end-point-spec instead, and directly request the provisioning of an OTU service.The TAPI Server MUST provide the RESTCONF Response according to the criteria provided in Table 37.
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 Relevant Parameters

The Relevant Parameters for this use case are included in UC1.0. In addition, the following PHOTONIC_MEDIA extensions are required.

Table 36: otsi-connectivity-service-end-point-spec (**CSEP**) object definition

otsi-connectivity-service-end-point-spec	**/tapi-common:context/tapi-connectivity:connectivity-context/tapi-connectivity:connectivity-service/tapi-connectivity:end-point/tapi-photonic-media:otsi-connectivity-service-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
central-frequency	• central-frequency: "[0-9]{9}", • "frequency-constraint": {adjustment-granularity, grid-type} ○ “adjustment-granularity”: [“UNCONSTRAINED”, “G_3_125GHZ”, “G_6_25GHZ”, “G_12_5GHZ”, “G_25GHZ”, “G_50GHZ”, “G_100GHZ”,] ○ “grid-type”: [“GRIDLESS”, “FLEX”, “CWDM”, “DWDM”]	RW	M	• Provided by *tapi-client* • The central-frequency of the laser specified in MHz. It is the oscillation frequency of the corresponding electromagnetic wave. • Adjustment-granularity in Gigahertz. As per ITU-T G.694.1, it is used to calculate nominal central frequency". • The grid-type specifies the reference set of frequencies used to denote allowed nominal central frequencies
spectrum	{lower-frequency, upper-frequency, frequency-constraint} • “upper/lower-frequency”: "[0-9]{9}", • "frequency-constraint": {adjustment-granularity, grid-type} ○ “adjustment-granularity”: [“UNCONSTRAINED”, “G_3_125GHZ”, “G_6_25GHZ”, “G_12_5GHZ”, “G_25GHZ”, “G_50GHZ”, “G_100GHZ”,] ○ “grid-type”: [“GRIDLESS”, “FLEX”, “CWDM”, “DWDM”]	RW	M	• Provided by *tapi-client* • The upper lower frequency boundaries of the band specified in MHz • Adjustment-granularity in Gigahertz. As per ITU-T G.694.1, it is used to calculate nominal central frequency. • The grid-type specifies the reference set of frequencies used to denote allowed nominal central frequencies
application-identifier	{application-identifier-type, application-code} • “application-identifier-type”:[“PROPRIETARY”, “ITUT_G959_1”, “ITUT_G698_1”, “ITUT_G698_2”, “ITUT_G696_1”, “ITUT_G695”,]	RW	M	• Provided by *tapi-client*

	“application-code”:”[0-9a-zA-Z_]{64}"			
modulation	[“RZ“, “NRZ”, “BPSK”, “DPSK”, “QPSK”, “8QAM”, “16QAM”]	RW	M	• Provided by tapi-client
transmit-power	{total-power, power-spectral-density} • “total-power”:"[0-9].[0-9]{7}", • “power-spectral-density”: "[0-9].[0-9]{7}",	RW	M	• Provided by *tapi-client*
laser-control	One of [FORCED-ON, FORCED-OFF, AUTOMATIC-LASER-SHUTDOWN, UNDEFINED]	RW	O	• Provided by *tapi-client*
total-power-warn-threshold-upper	[0-9].[0-9]{7}	RW	O	• Provided by *tapi-client*
total-power-warn-threshold-lower	[0-9].[0-9]{7}	RW	O	• Provided by *tapi-client*
local-id	"[0-9a-zA-Z_]{32}"	RW	M	• Provided by *tapi-client*

6.2.10.2 TAPI Server response behavior.

Please consider this list as preliminary. It will be updated based on received feedback.

Table 37: UC2a expected response behavior.

HTTP Response status code	Error-tag	Error-message	Condition description
200			Success
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..

6.2.11 Use case 2b: Unconstrained DSR service provisioning with ODU channel selection

Number	**UC2b**
Name	**Unconstrained DSR service provisioning with ODU channel selection**
Technologies involved	Optical, ODU 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 service with the selection of the ODU tributary slot (*channel selection*) to be used in the multiplexing of the mapped client signal into the OTN frame. NOTE: this UC assumes that the TAPI client requests a DSR service which, in turn, is mapped into a LO ODU container and multiplexed into a HO ODU container.
Layers involved	ODU
Type	Provisioning
Description & Workflow	See the detailed workflow UC1.0 with [**server-restrictions**].

6.2.11.1 Relevant Parameters

The following ODU extensions are required.

Table 38: ODU connectivity-service-end-point (**ODU CSEP**) object definition (UC2b)

ODU CSEP	**/tapi-common:context/tapi-connectivity:connectivity-context/tapi-connectivity:connectivity-service/tapi-connectivity:end-point/tapi-odu:odu-connectivity-service-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
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	M	Provided by *tapi-client.* In case the ODU server layer is an ODUCn, each entry in the list is an integer value (P) representing the time slot name TS#A.B (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)
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.	RW	O	Provided by *tapi-client*.

	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			

6.2.11.2 Connectivity Service request processing

The workflow involves:

- Request the DSR connectivity service using the container **server-connectivity-service-end-point** in order to specify the ODU endpoints and select the tributary slot and port (for multiplexing),
- Specify the tributary slot list as defined in the **odu-csep-ctp-pac** container data node of the **odu-connectivity-service-end-point-spec** CSEP augment.

The server endpoints (ODU CEP) MUST use the following augment:

```
augment /tapi-common:context/tapi-connectivity:connectivity-context/tapi-
connectivity:connectivity-service/tapi-connectivity:end-point:
    +--rw odu-connectivity-service-end-point-spec
       +--rw odu-csep-ctp-pac
       |  +--rw tributary-port-number?   uint64
       |  +--rw tributary-slot-list*     uint64
       +--rw odu-csep-ttp-pac
       |  +--rw configured-mapping-type?   mapping-type
       |  +--ro configured-client-type?    tapi-dsr:digital-signal-type
       +--rw odu-csep-common-pac
          +--rw odu-type?                  odu-type
          +--rw odu-rate?                  uint64
          +--rw opu-tributary-slot-size?   odu-slot-size
          +--rw number-of-odu-c?           uint64
```

For example, the endpoints list of the Connectivity service (only source endpoints shown)

```
  "end-point" : [
     {
         "layer-protocol-name" : "DSR",
         "layer-protocol-qualifier" : "GBEX",
         "service-interface-point" : {"service-interface-point-uuid" : "DSR_SIP"},
         "local-id" : 1,
         "server-connectivity-service-end-point" : {
               "connectivity-service-uuid" : <uuid of this connectivity service>
               "local-id" :  "10"
         }
     },
     {
         "layer-protocol-name" : "ODU",
         "layer-protocol-qualifier" : "ODU-TYPE-ODUj",
         "service-interface-point" : {"service-interface-point-uuid" : DSR SIP},
         "local-id" : "10",
         "odu-connectivity-service-end-point-spec" : {

           "odu-csep-ctp-pac" : {
                 "tributary-port-number? Uint64 → may be left for default
                 "tributary-slot-list" : [1,2,3...]
           },
           "odu-csep-common-pac" : {
               "odu-type" : "ODU-TYPE-ODUj",
           }
```

```
      },
...
   ],
}
```

6.2.12 Use case 2c: Unconstr. PHOTONIC_LAYER_QUALIFER_MC/MCA serv. prov. with spectrum selection

Number	**UC2c**
Name	Unconstrained PHOTONIC_LAYER_QUALIFER_MC / MCA 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. The UC relies on the **tapi-photonic-media:mca-connectivity-service-end-point** augmentation (the **tapi-photonic-media:media-channel-connectivity-service-end-point** is deprecated in favor of mca also in the case of 1, i.e., mca-connectivity-service-end-point-spec/number-of-mc = 1) of **tapi-connectivity:connectivity-service/end-point** objects. UC MUST support number-of-mc=1. Two different mutually exclusive methods of constraints are supported: the explicit spectrum band and the capacity:

```
  augment /tapi-common:context/tapi-connectivity:connectivity-
context/tapi-connectivity:connectivity-service/tapi-connectivity:end-
point:
    +--rw mca-connectivity-service-end-point-spec
       +--rw mc-config* [local-id]
       |  +--rw spectrum
       |  |  +--rw upper-frequency?        uint64
       |  |  +--rw lower-frequency?        uint64
       |  |  +--rw frequency-constraint
       |  |     +--rw adjustment-granularity?   adjustment-granularity
       |  |     +--rw grid-type?                grid-type
       |  +--rw power-management-config-pac
       |  |  +--rw intended-maximum-output-power
       |  |  |  +--rw total-power?              decimal64
       |  |  |  +--rw power-spectral-density?   decimal64
       |  |  +--rw intended-minimum-output-power
       |  |  |  +--rw total-power?              decimal64
       |  |  |  +--rw power-spectral-density?   decimal64
       |  |  +--rw expected-maximum-input-power
       |  |  |  +--rw total-power?              decimal64
       |  |  |  +--rw power-spectral-density?   decimal64
       |  |  +--rw expected-minimum-input-power
       |  |     +--rw total-power?              decimal64
       |  |     +--rw power-spectral-density?   decimal64
       |  +--rw local-id                          string
       |  +--rw name* [value-name]
       |     +--rw value-name    string
       |     +--rw value?        string
       +--rw number-of-mc?   uint64
```

	Spectrum constraints: • Used to explicitly define the spectrum band to be occupied by the MC connectivity-service *by using the upper-frequency and lower-frequency fields* **Capacity constraints:** • Used to implicitly define the spectrum band. The spectrum capacity MUST be expressed in GHz units by using the *requested-capacity* at the connectivity service object level. The power management constraints are modeled by the power-management-config-pac object. This allows the TAPI Client to define the expected-max/min input power which will be injected into the end-points of the MC connectivity-service and also, it allows to request to the TAPI Server to provide a target intended-max/min-output-power. The TAPI Server MUST provide the RESTCONF Response according to the criteria provided in Table 40.
Layers involved	PHOTONIC_MEDIA
Type	Provisioning
Description & Workflow	This UC is implemented following the same workflow described in "Description & Workflow" of UC1.0

6.2.12.1 Relevant Parameters

Table 39: mca-connectivity-service-end-point-spec object definition

MCA CSEP	**/tapi-common:context/tapi-connectivity:connectivity-context/tapi-connectivity:connectivity-service/tapi-connectivity:end-point/tapi-photonic-media:mca-connectivity-service-end-point-spec**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
number-of-mc	[0-9]{9}	RW	M	• Provided by tapi-client • Fixed to 1
mc-config	List of {otsi-config [local-id]} • otsi-config: {spectrum, power-management-config-pac }	RW	M	• Provided by tapi-client
spectrum	{lower-frequency, upper-frequency, frequency-constraint} • "upper/lower-frequency": "[0-9]{9}", • "frequency-constraint": {adjustment-granularity, grid-type} ○ "adjustment-granularity": ["UNCONSTRAINED", "G_3_125GHZ", "G_6_25GHZ", "G_12_5GHZ", "G_25GHZ", "G_50GHZ", "G_100GHZ",]	RW	M	• Provided by tapi-client • The upper/lower-frequency boundaries of the band specified in MHz. • Adjustment-granularity in Gigahertz. As per ITU-T G.694.1, it is used to calculate nominal central frequency". • The grid-type specifies the reference set of frequencies used to denote allowed nominal central frequencies

	○ "grid-type": ["GRIDLESS", "FLEX", "CWDM", "DWDM"]			
power-management-config-pac	{intended-maximum-output-power, intended-minimum-output-power, expected-maximum-input-power, expected-minimum-input-power} • "intended-maximum-output-power":{total-power, power-spectral-density} ○ "total-power":"[0-9].[0-9]{64}", ○ "power-spectral-density":"[0-9].[0-9]{64}" • "intended-minimum-output-power":{total-power, power-spectral-density} ○ "total-power":"[0-9].[0-9]{64}", ○ "power-spectral-density":"[0-9].[0-9]{64}" • "expected-maximum-output-power":{total-power, power-spectral-density} ○ "total-power":"[0-9].[0-9]{64}", ○ "power-spectral-density":"[0-9].[0-9]{64}" • "expected-minimum-output-power":{total-power, power-spectral-density} ○ "total-power":"[0-9].[0-9]{64}", ○ "power-spectral-density":"[0-9].[0-9]{64}"	RW	M	• Provided by tapi-client
local-id	"[0-9a-zA-Z_]{32}"	RW	M	Provided by tapi-client

6.2.12.2 TAPI Server response behavior.

Please consider this list as preliminary. It will be updated based on received feedback.

Table 40: 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.

6.3 Constrained Provisioning

[**constrained**] The term indicates that the TAPI-Client may add **routing constraint(s)** in the service request.

NOTE:

i) Since there currently is no mechanism to indicate whether a set of constraints MUST or SHOULD be applied, by default these topology 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 topology constraints are applied to a service with restoration capabilities, any reroute action SHOULD account for any inclusion/exclusion topology constraints policy defined if possible but, as a general rule, the restoration MUST always take place even if the topology constraints enter in conflict with the new route.

6.3.1 Use case 3a: Include/exclude a node or group of nodes.

Number	**UC3a**
Name	**Constrained provisioning: Include/exclude a node or group of nodes**
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 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 ODU/OTSi/OMS/OTS. 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/ODU/PHOTONIC_MEDIA
Type	Provisioning
Description & Workflow	This UC is implemented following the same workflow described in “Description & Workflow” of UC1.0

6.3.1.1 Relevant Parameters

Table 41 complements the information included in the unconstrained service provisioning use cases.

Table 41: Connectivity-service node topology-constrains object definitions.

connectivity-service	**/tapi-common:context/tapi-connectivity:connectivity-context/tapi-connectivity:connectivity-service**

Attribute	Allowed Values/Format	Mod	Sup	Notes
include-node	List of valid node uuids. The canonical representation uses lowercase	RW	M	• Unordered and partial list • *Declarative* routing constraints not in the scope.
exclude-node	List of valid node uuids. The canonical representation uses lowercase	RW	M	

6.3.2 Use case 3b: Include/exclude a link or group of links.

Number	UC3b
Name	**Constrained provisioning: 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/ODU/PHOTONIC_MEDIA
Type	Provisioning
Description & Workflow	This UC is implemented following the same workflow described in "Description & Workflow" of UC1.0

6.3.2.1 Relevant Parameters

Table 42 complements the information included in the unconstrained service provisioning use cases.

Table 42: Connectivity-service link topology-constrains object definitions.

connectivity-service	**/tapi-common:context/tapi-connectivity:connectivity-context/tapi-connectivity:connectivity-service**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
include-link	List of valid link uuids.	RW	M	• Unordered and partial list • *Declarative* routing constraints not in the scope.
exclude-link	List of valid link uuids.	RW	M	

6.3.3 Use case 3c: Include/exclude the route used by another service.

Number	**UC3c**
Name	Constrained provisioning: Include/exclude the route used by another service.
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 or exclusion of the resources used by another connectivity service(s). **Coroute-Inclusion:** Implementations SHOULD perform path computation 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 perform path computation 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:* - A DSR service that includes an ODU service means that implementations SHOULD encapsulate the new DSR in the ODU service - 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. 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

6.3.3.1 Relevant Parameters

Table 43 complements the information included in the unconstrained service provisioning use cases.

Table 43: Connectivity-service coroute-inclusion and diversity-exclusion object definitions.

connectivity-service	**/tapi-common:context/tapi-connectivity:connectivity-context/tapi-connectivity:connectivity-service**			
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	M	• Provided by *tapi-client*
coroute-exclusion	List of {connectivity-service-uuid: **connectivity-service-ref** - */tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/uuid* }	RW	M	• Provided by *tapi-client*

6.3.4 Use case 3d: Diverse Routing in SRG failure.

Number	**UC3d**
Name	Diverse Routing in SRG failure
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.** Assumption: SRGs are predefined (in links, nodes, etc). This use case regards the provisioning of Shared Risk Groups (SRG) policies and provide route disjointness upon these policies. This service is subject to be requested to the SDN-C including **an SRG disjoint-policy and also includes any protection-policy implying a second protection path routing computation**, the SDN-C MUST compute that both routes (Nominal and Backup) do not share any SRG present in the network. In other words, the network resources employed to route these two connections (nominal and protection) should not share any risk group element.
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

6.3.4.1 Relevant Parameters

Table 44: Connectivity-service diversity-policy for SRGs. complements the information included in the unconstrained service provisioning use cases

Table 44: Connectivity-service diversity-policy for SRGs.

Connectivity-service	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivityservice**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
diversity-policy	One of ["SRLG","SRNG", "SNG", "NODE", "LINK"]	RW	M	• Provided by *tapi-client* [mandatory for this use case: SRLG or SRNG values]

6.3.5 Use case 3e: Constrained Provisioning based on min hops policy

Number	**UC3e**
Name	Constrained Provisioning based on min hops policy
Technologies involved	Optical, ODU, 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 shall enforce the TAPI Server to minimize the number of links of the lowest server layer and qualifier in the context (e.g., links between NEPs with layer-protocol-name PHOTONIC_MEDIA and supported-cep-layer-protocol-qualifier including LAYER_PROTOCOL_QUALIFIER_UNSPECIFIED). This use case can be combined with the following set of constrains defined by the TAPI Client: exclude/include node, exclude/include link, exclude/include service's route (exclude/include SRGs FFS). In these cases, the previous constrains MUST be applied before the MIN_WORK_ROUTE_HOP route-objective-function is applied. In case of applying this use case for protection services, the TAPI client MAY 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/ODU/PHOTONIC_MEDIA
Type	Provisioning
Description & Workflow	This UC is implemented following the same workflow described in "Description & Workflow" of UC1.0

6.3.5.1 Relevant Parameters

Table 45: Connectivity-service route-objective-function (UC3e). complements the information included in the unconstrained service provisioning use cases.

Table 45: Connectivity-service route-objective-function (UC3e).

Connectivity-service	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivityservice**			
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

6.3.6 Use case 3f: Constrained Provisioning based on min latency policy

Number	**UC3f**
Name	Constrained Provisioning based on min latency policy
Technologies involved	Optical, ODU, 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. This use case can be combined with the following set of constrains defined by the TAPI Client: exclude/include node, exclude/include link, exclude/include service's route (exclude/include SRGs FFS). In these cases, the previous constrains MUST be applied before the MIN_WORK_ROUTE_LATENCY route-objective-function is applied. In case of applying this use case for protection services, the TAPI client MAY 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/ODU/PHOTONIC_MEDIA
Type	Provisioning
Description & Workflow	This UC is implemented following the same workflow described in "Description & Workflow" of UC1.0

6.3.6.1 Relevant Parameters

The table below complements the information included in the unconstrained service provisioning use cases.

Table 46: Connectivity-service route-objective-function (UC3f)

connectivity-service	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivityservice**			
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*

6.4 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:

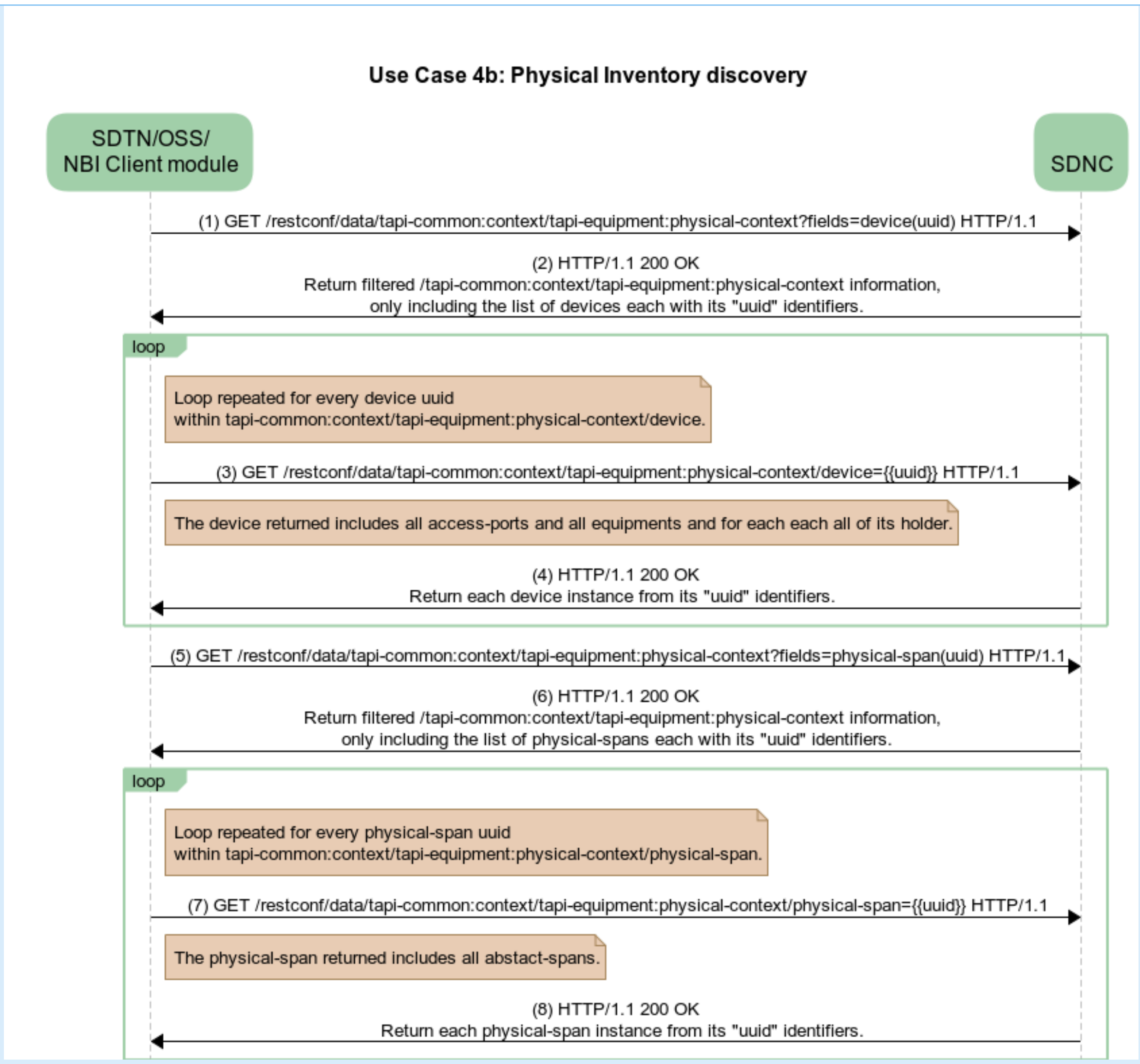
- EQUIPMENT_CATEGORY_RACK,
- EQUIPMENT_CATEGORY_SUBRACK,
- EQUIPMENT_CATEGORY_CIRCUIT_PACK,
- EQUIPMENT_CATEGORY_SMALL_FORMFACTOR_PLUGGABLE,
- EQUIPMENT_CATEGORY_ STAND_ALONE_UNIT.

6.4.1 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.

6.4.2 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.



6.4.2.1 Relevant Parameters

The parameters listed in Table 47 MUST be included in the structure of the equipment. *Note: the basic structure of the equipment does not include the rack position as a mandatory field within the NBI. This Rack position should be added to the database of OSSs once the information is provided by the installers or from an external database.*

Table 47: Parameters for equipment model required.

Conceptual parameter	**TAPI xPath reference** **/tapi-common:context/tapi-equipment:physical-context/device/equipment/...**
Part number	actual-equipment/common-actual-properties/asset-instance-identifier
Serial number	actual-equipment/common-actual-properties/serial-number

Name	name
Description	actual-equipment/common-equipment-properties/equipment-type-description
Component Version	actual-equipment/common-equipment-properties/equipment-type-version
Type	category
Relative position of the component into the network element	contained-holder/actual-holder/common-holder-properties/holder-location
Removable	contained-holder/actual-holder/common-holder-properties/is-guided
Manufacturer	actual-equipment/common-equipment-properties/manufacturer-name
Operator_ID_type	contained-holder/actual-holder/common-holder-properties/asset-type-identifier
Operational state[7]	/tapi-common:context/tapi-topology:topology-context/topology/node/owned-node-edge-point/operational-state
Admin state[7]	/tapi-common:context/tapi-topology:topology-context/topology/node/owned-node-edge-point administrative-state

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**

Table 48: 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}	RO	M	• Provided by *tapi-server*
uuid	Device uuid.	RO	M	• As per RFC 4122 • 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} • equipment-uuid **name** MUST include	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 AccessPort

[7] NOTE: Since TAPI 2.1.3 does not include admin and operational state yang leaves for physical context objects (such as equipment), they SHOULD be reflected into the relevant topological element (node/NEP)

	"value_name": "PORT_NUMBER", "value": " [0-9a-zA-Z_]{64}"			• **equipment-uuid(s):** 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.*

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*
actual-equipment	{actual-non-field-replaceable-module, common-actual-properties, common-equipment-properties}	RO	M	• Provided by *tapi-server*
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 49: 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 50: 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 51: 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 manufacturer'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 52: 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 53: 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*
access-port	Including: device-uuid, access-port-uuid	RO	M	• Provided by *tapi-server*
abstract-strand	Including, optionally: List of adjacent strands List of spliced strands List of connector-pin List of strand-media-characteristics	RO	O	• Provided by *tapi-server* • Depends on the composition of the physical-strand • Strand media characteristics MAY encode properties of e.g. fiber, etc. and the current format is unspecified

6.4.2.2 Relative location of component with TAPI 2.1.3 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:

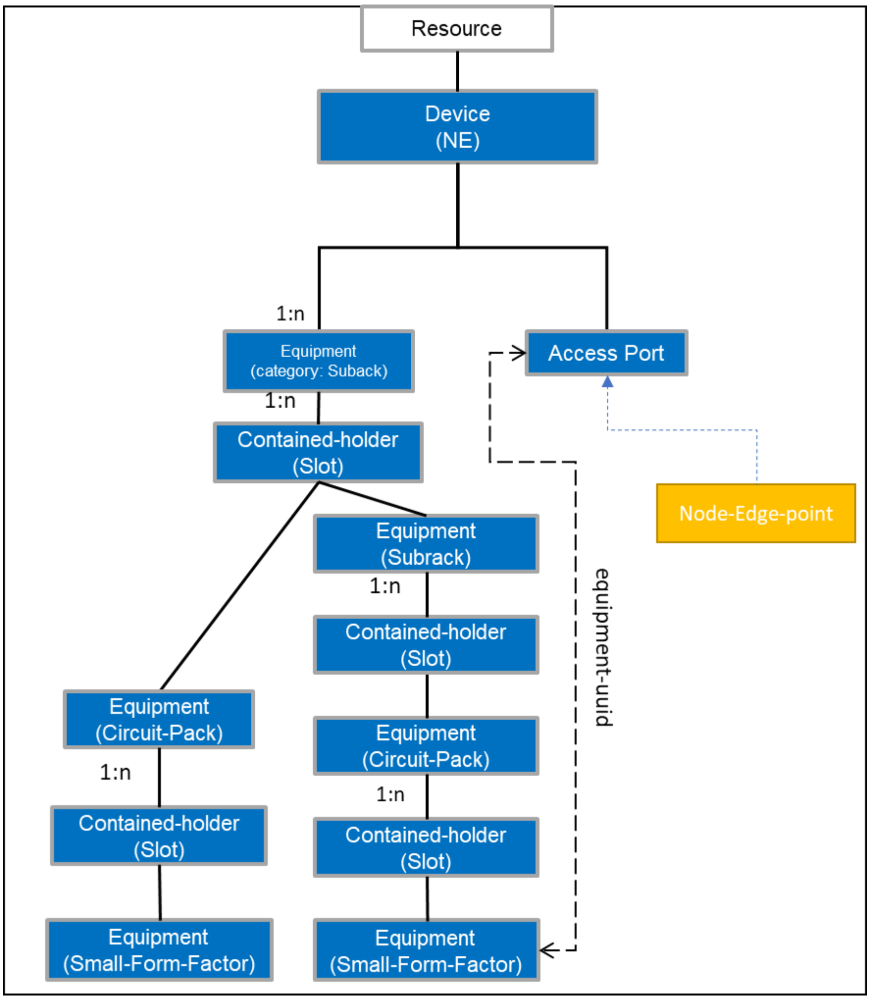


Figure 6-31 UC-4b Hierarchical arrangement of equipment objects with TAPI 2.1.3.

- 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.

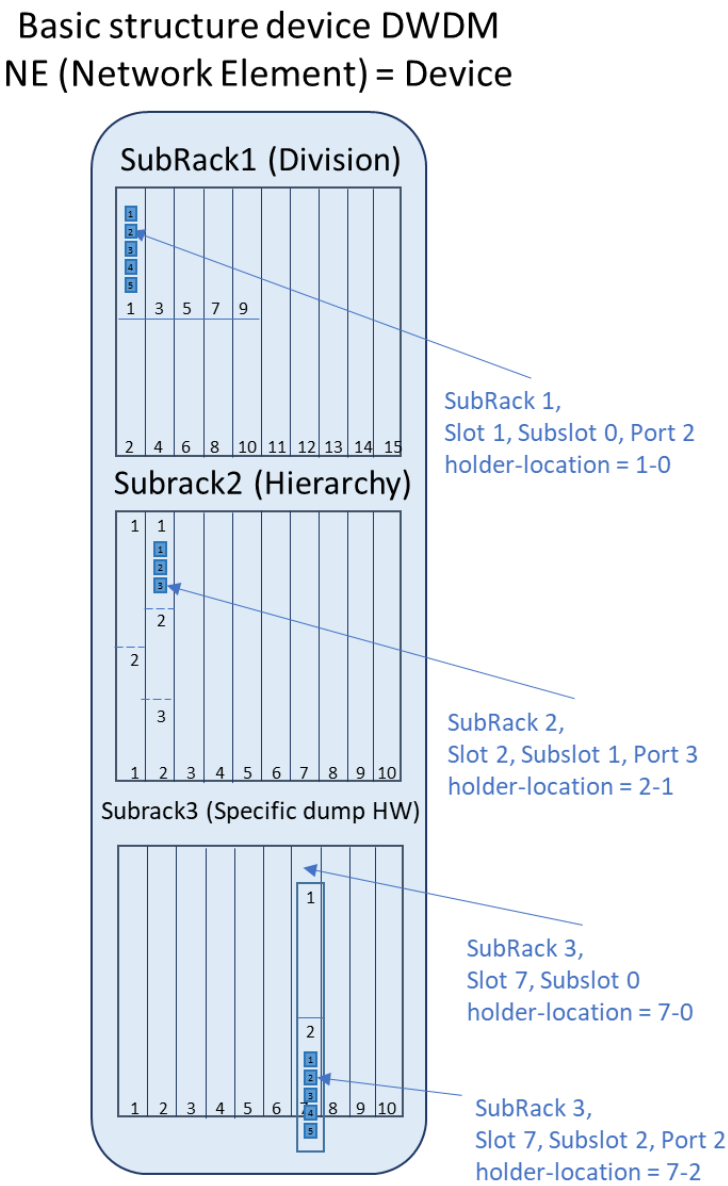


Figure 6-32 UC-4b Network Element Subracks container-holder location examples.

To complete the picture, the examples illustrated in Figure 6-32 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"}]

6.5 Resiliency

This section deals with use cases covering resiliency (i.e., protection and restoration). 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 FC being a connection and resource referring to a route):

- NON-REVERTIVE, according to TAPI description: "An FC switched to a lower priority (non-preferred) resource will not revert to a higher priority (preferred) resource when that recovers."
- REVERTIVE, according to TAPI description: "An FC switched to a lower priority (non-preferred) resource will revert to a higher priority (preferred) resource when that recovers (potentially after some hold-off time)."

Additionally,

- WAIT-TO-REVERT-TIME according to TAPI description: "If the protection system is revertive, this attribute specifies the time, in minutes, to wait after a fault clears on a higher priority (preferred) resource before reverting to the preferred resource.". Note that in TAPI v2.1.3 this value defaults to 15 minutes.

The specific TAPI path to which these parameters refer is:

```
module: tapi-connectivity
  augment /tapi-common:context:
    +--rw connectivity-context
       +--rw connectivity-service* [uuid]
       |  +--rw reversion-mode?                         reversion-mode
       |  +--rw wait-to-revert-time?                    uint64
```

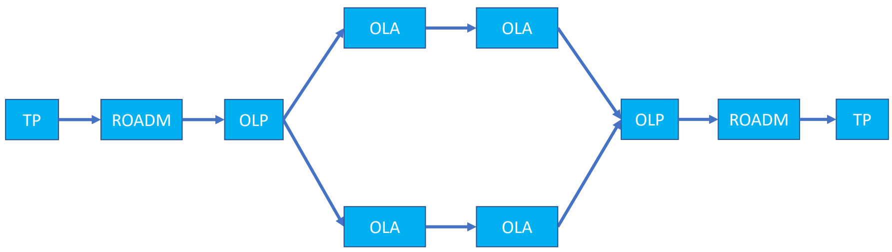
For the resilience use cases, the following parameters apply, Table 54: Connectivity-service parameters for reversion.

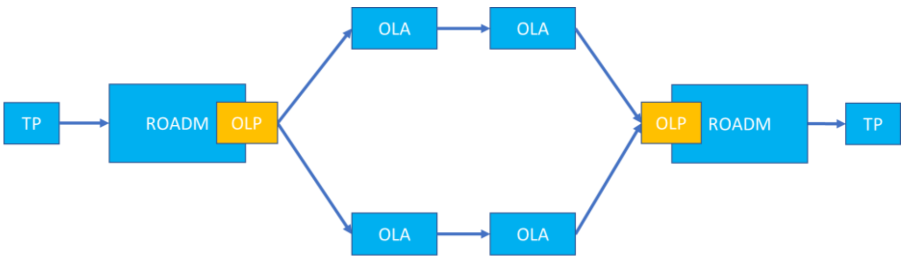
Table 54: Connectivity-service parameters for reversion

connectivity-service	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
reversion-mode	One of ["REVERTIVE", "NON-REVERTIVE"]	RW	M	• Provided by *tapi-client*
wait-to-revert-time	"[0-9]{4}"	RW	C	• Provided by *tapi-client* • This attribute is mandatory in connection objects when the reversion-mode REVERTIVE and it is different of the default value (15). • The supported values MAY be additionally constrained by the underlying hardware. A config operation with unsupported values MUST fail.

6.5.1 Use case 5a: OLP OMS/OTS Protection Discovery

<table>
<tr><th>Number</th><th>UC5a</th></tr>
<tr><td>Name</td><td>OLP OMS/OTS Protection Discovery</td></tr>
<tr><td>Technologies involved</td><td>Optical</td></tr>
<tr><td>Process/Areas Involved</td><td>Planning and Operations</td></tr>
<tr><td>Brief description</td><td>
This use case covers the use of Optical Line Protection elements for protected services at OMS/OTS layers at the degree side. The following figures represent the usage of an OLP for OMS/OTS protection in the cases: 1) OLP is a standalone node and 2) OLP is part of a ROADM. In 1), the OLP appears as a node whose NEPs are PHOTONIC_MEDIA and UNSPECIFIED CEP qualifier (see [TAPI-TOP-MODEL-REQ-18][TAPI-TOP-MODEL-REQ-19]). This version of the RIA only mandates case 1). Case 2) is left for further study (and could be represented with a floating NEP). This UC covers 1:1 and 1+1.





OMS/OTS OLP protection is not intended to be configured by the user, but to be represented by the TAPI server as part of the PHOTONIC_MEDIA layer topology. Hence the OMS/OTS protection is not provisioned by a connectivity-service. The TAPI server is responsible of the automatic discovery of the OMS protection scheme and its representation.
An OMS/OTS protection MUST be represented as described in [TAPI-TOP-MODEL-REQ-23]. The Link object representing the OMS protected resource (the MC link) MUST be present and MUST contain the /tapi-topology:link/tapi-topology:resilience-type/protection-type attribute specify which type of protection service is provided.
The OMS/OTS protected link resource MUST be realized by a (top-level) connection which MUST be reported within the /tapi-connectivity:connectivity-context/connection list. NOTE: As seen in the figure, the link between the ROADM and the OLP is not protected yet the MC-link claims to be protected. Further refinements of the UC and the model are to be considered in newer versions of this RIA.
</td></tr>
</table>

	Depending on the type of protection the MC link attribute MUST be set with the following values: • **ONE_PLUS_ONE_PROTECTION:** Dual transmitting and selective receiving. • **ONE_FOR_ONE_PROTECTION:** Selective transmitting and selective receiving. The protection process MUST be triggered automatically and the TAPI server MUST notify the TAPI client about the service condition changes through the tapi-notification service (as defined in UCs 15a and 15b).
Layers involved	PHOTONIC_MEDIA
Type	Resilience
Description & Workflow	This type of protection (OMS/OTS OLP) is not provisioned but only discovered.

6.5.1.1 Expected result

The expected representation of the OTS/OMS OLP protection schema is shown in the following TAPI topology.

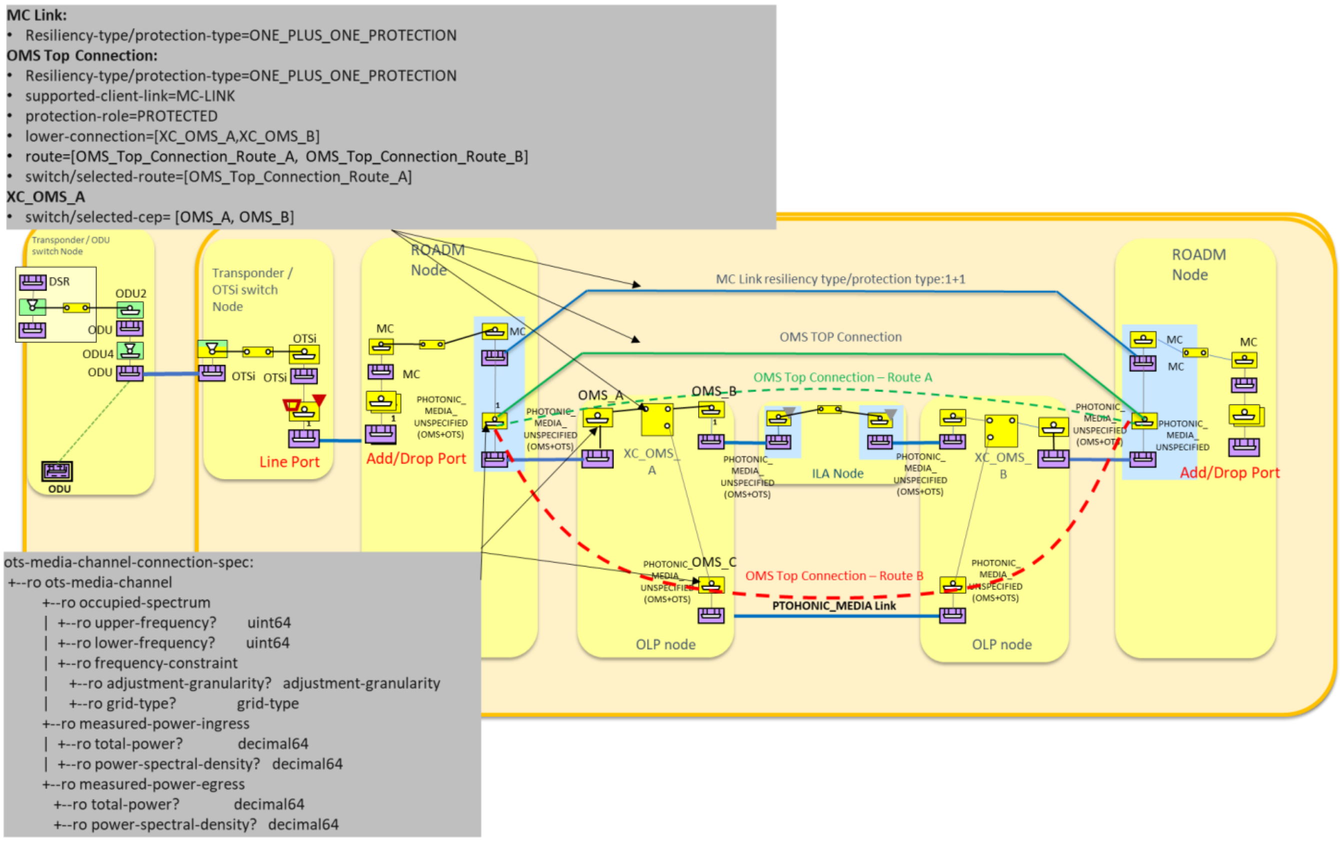
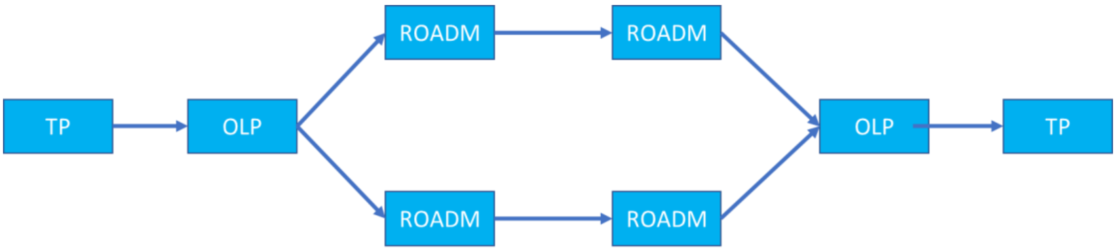


Figure 6-33 UC-5a OLP protection TAPI representation. Note that Route A and Route B represented as dashed lines are not connections (only refer to routes of the single OMS top connection).

Note 1: Transponder representation in the Z side has been omitted.

Note 2: The Top-Connection switch control is under discussion thus the selected-route attribute shall be considered optional when implementing OLP protection UCs. In contrast, the switch-control at the XC level MUST be present, representing/reflecting the actual switch state.

6.5.2 Use case 5b: OLP-based Transponder to Transponder Protection with Diverse Service Provisioning

Number	**UC5b**
Name	**OLP-based Transponder to Transponder Protection with Diverse Service Provisioning**
Technologies involved	Optical
Process/Areas Involved	Planning and Operations
Brief description	This use case covers the use of OLP elements for protected services at the add-drop ports, where an OLP is placed between a transponder line port and two add/drop ports of the ROADM(s) (see figure). The protection is configured upon a user request (configured by a connectivity-service provisioning with protection constraints). The TAPI server MUST notify the TAPI client about the service condition changes. For example, as defined in UCs 15a and 15b.  This UC covers two scenarios: • UC5b-1 : the provisioning of a protected OTSi Connectivity service • UC5b-2 : the provisioning of protected MC Connectivity service The Connectivity Service object sent to the TAPI Server MUST include the ***tapi-connectivity:connectivity-service/tapi-topology:resilience-type/protection-type*** attribute to specify which type of protection service is requested. Depending on the type of protection this attribute MUST be set to one the following values: • **ONE_PLUS_ONE_PROTECTION:** Dual transmitting and selective receiving. • **ONE_FOR_ONE_PROTECTION:** Selective transmitting and selective receiving. This use case does not detail intermediate regeneration, this capability is left for future use case specification
Layers involved	PHOTONIC_MEDIA
Type	Resilience

Description & Workflow	This protection scheme requires the reservation of two disjoint routes along the PHOTONIC_MEDIA layer for the provisioning of connections. The TAPI Client MAY delegate the protection role selection to the TAPI Server during the CS provisioning process. Figure 6-34 (using a transitional ODU/OTSi link) **UC5b-1 : the provisioning of a protected OTSi Connectivity service** For this subcase, the connectivity service is PHOTONIC_MEDIA/OTSi. The two main CSEPs refer to the OTSi SIPs (e.g., 52.D1). If the client wishes to specify the protection roles, it MAY add two additional MC/ PHOTONIC_MEDIA CSEPs referring to the involved SIPs (73.D1 and 72.D1) SIPs within the CS request. **UC5b-2 : the provisioning of protected MC Connectivity service** For this subcase, the connectivity service is PHOTONIC_MEDIA/MC. The TAPI Client MAY include the three SIPs (per end) mapped to the MC/PHOTONIC_MEDIA NEPs representing the switching ports of the OLP component (MC1, MC2, MC3 as represented in the Figure), for the appropriate provisioning of Protection Roles (within the CSEPs definition). These protection roles MAY be in the example PROTECTED (71.D1 - MC1), WORK (72.D1 - MC2), PROTECT (73.D1 - MC3). **NOTE** for both UC5b-1 and 5b-2: with TAPI 2.1.3 it is only possible to specify protection roles if the corresponding SIPs are defined in the context. In TAPI 2.3 it shall be possible to specify routing constraints based on protection roles for this purpose when internal SIPs are not defined. The TAPI server MAY implement the switch control only at the top-level connection (by using one switch and the selected-route list) or by using the subordinate switch control as in Example 1. This UC is implemented following the same workflow described in “Description & Workflow” of UC1.0

6.5.2.1 Expected results

As described in UC1f (Section 6.2.7) the PHOTONIC_MEDIA layer may be modelled unidirectionally or bidirectionally. In this use case two alternative models are presented:

1. Full-bidirectional - both UNI MC and PHOTONIC_MEDIA models are bidirectional
2. Full-unidirectional - both UNI and PHOTONIC_MEDIA models are unidirectional

6.5.2.1.1 UC5b Example 1: Bidirectional OLP Line protection modelling

In Figure 6-34, The expected result after the creation of the OLP protected PHOTONIC_LAYER_QUALIFER_OTSI connectivity service is represented over the TAPI topology scenario included. The bidirectional 71.D1 UNI MC SIP is associated to its bidirectional MC NEP, similarly the bidirectional 72.D1 and 73.D1 “unreliable” SIPs. Please note, OTSiMC layer is intentionally left out from the diagram for simplicity but it MAY be included on top of every MC CEP described in the example.

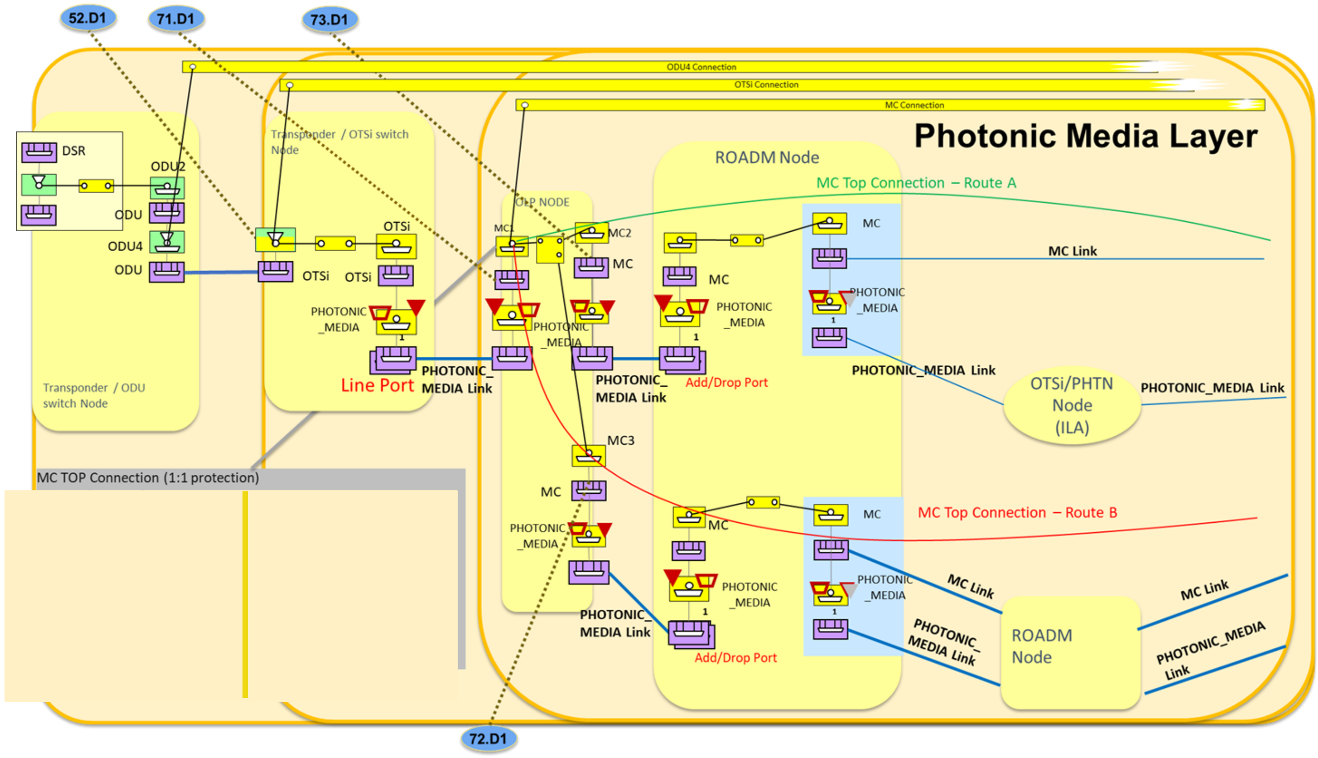


Figure 6-34 UC-5b OLP TAPI Connectivity-Service low-level description.

As an example, the provisioning of a PHOTONIC_LAYER_QUALIFER_OTSI Connectivity Service triggers the creation of:

- **An OTSi Top Connection**.
- **An MC Top Connection**: which has two routes and includes one switch-control instance. Such switch-control instance recursively includes two subordinate switch-controls (sub-switch-control) and no switch. Each of the subordinate switch controls references (*points to*) a switch-control included by its respective lower-connection (by using a connection-uuid and switch-control-uuid).
    - Each of the aEnd and zEnd lower-connection switch-control includes one switch instance.
    - **aEnd OLP site MC lower connection**: includes the switch control including the switch between aEnd working and protection CEPs (MC2, MC3) [no sub-switch-control].
    - **N >= 0 intermediate MC lower connections**: without switch control (connections along the ROADM nodes)
    - **zEnd OLP site MC lower connection**: includes the switch control between zEnd working and protection CEPs (symmetric respective CEPs to MC2, MC3 in the other aEnd OLP) [no sub-switch-control].

In case of ONE_FOR_ONE_ PROTECTION for the aEnd / zEnd switches, both fields:

- **selected-connection-end-points:** either MC2 or MC3 (and the corresponding CEP in zEnd) is selected according to the conditions.
- **(optionally) selected-route:** either Route-A or Route-B is selected according to the conditions.

In case of ONE_PLUS_ONE_PROTECTION:

- **selected-connection-end-points:** either MC2 or MC3 is selected, *in the receive direction*, according to the conditions.
    - There is no correlation between the switches at the protection scheme ends.
- **(optional) selected-route:** both Route-A and Route-B are always selected.

6.5.2.1.2 UC5b Example 2: Unidirectional OLP Line protection modelling

In this case the connectivity model is split in the two directions (Add/Drop).

- The Connectivity Service is assumed to be bidirectional.
- SIPs are assumed unidirectional.
- The Connectivity Service request in this case MUST include four CSEPs referencing the Add and Drop SIPs at aEnd and zEnd points.
- Once established, the Connectivity Service MUST include two MC Top Connections (in its connection list), one for each direction (e.g., *Add direction* and *Drop direction* in the Figure below).
- The two unidirectional Top Connections MUST expose the management of their respective switch-control in the same way as described in the previous example. Thus, the switch control for each direction is independent and the bidirectional connectivity-service operational-state is impacted by the status of both. Note: In the case of 1+1 only the receiving side connection has a switch-control instance.

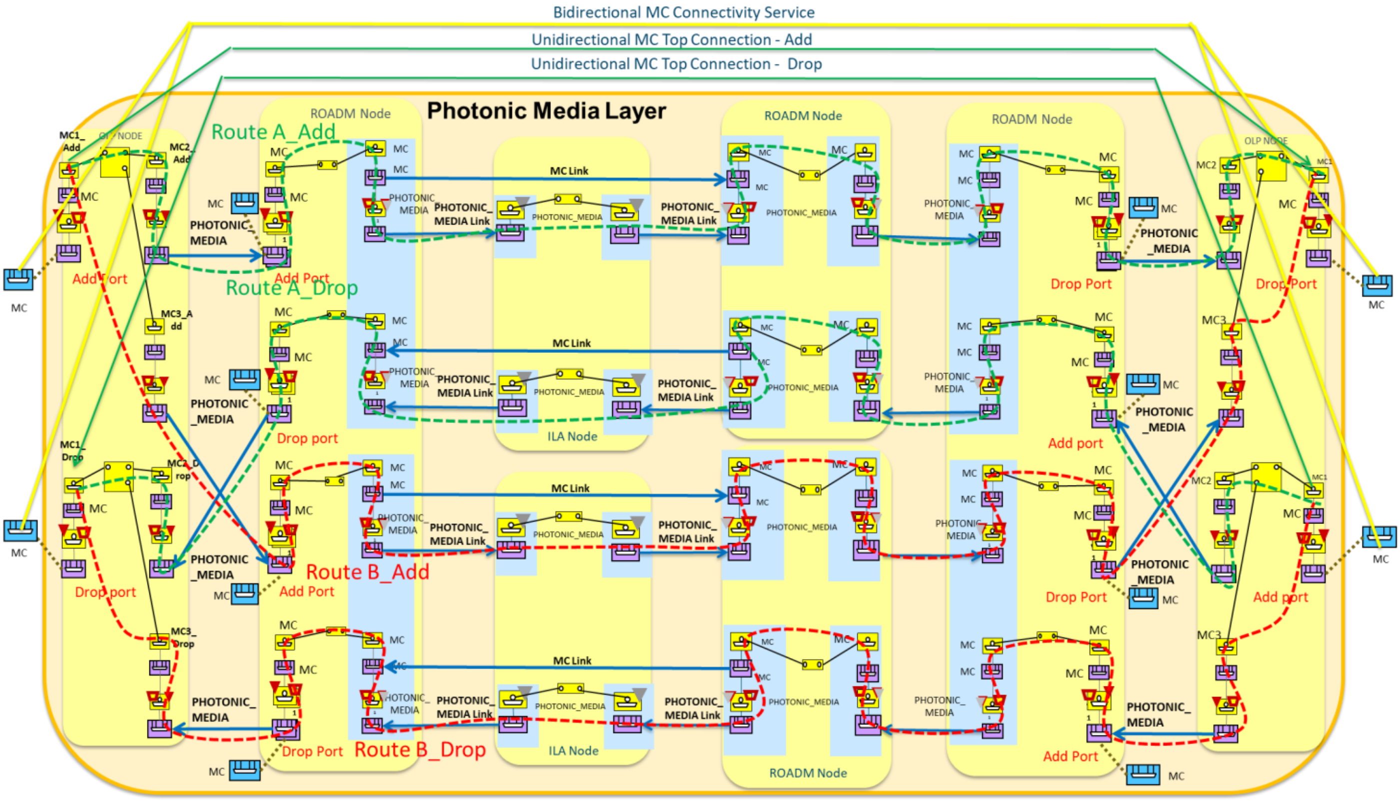


Figure 6-35 UC-5b OLP TAPI Connectivity-Service low-level description. Unidirectional OLS modelling.

6.5.2.2 Relevant Parameters

Tables in this section complement the information included in the unconstrained service provisioning use cases.

Table 55: Connectivity-service parameters for 1+1 UC5b.

connectivity-service	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
resilience-type	"**protection-type**": one of ["ONE_FOR_ONE_PROTECTION", "ONE_PLUS_ONE_PROTECTION"]	RW	M	• Provided by *tapi-client*
preferred-restoration-layer	PHOTONIC_MEDIA	RW	M	• Provided by *tapi-client*
hold-off-time	"[0-9]{4}"	RW	O	• Provided by *tapi-client*
max-switch-times	"[0-9]{2}"	RW	O	• Provided by tapi-client
is-coordinated-switching-both-ends	One of [true, false]	RW	O	• Provided by tapi-client
is-lock-out	One of [true, false]	RW	O	• Provided by tapi-client
is-frozen	One of [true, false]	RW	O	• Provided by tapi-client

Table 56: Connectivity-service-End-Points parameters for UC5b.

connectivity-service-end-point	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service/end-point**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
protection-role	One of ["WORK", "PROTECT", "PROTECTED"]	RW	O	• Provided by *tapi-client*

Table 57: Connection parameters for UC5b.

connection	**/tapi-common:context/tapi-connectivity:connectivity-context/connection**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
switch-control	List of { **switch-control** }	RO	C	• Provided by *tapi-server* • Must appear as defined above.

Table 58: Switch-control parameters for UC5b.

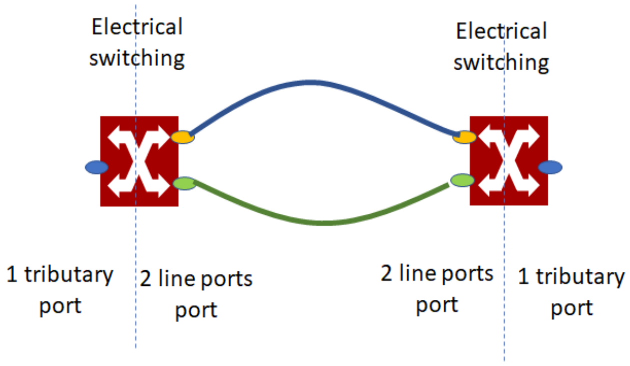
switch-control	**/tapi-common:context/tapi-connectivity:connectivity-context/connection/switch-control**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
uuid	Switch control uuid.	RO	M	• As per RFC 4122 • Provided by ***tapi-server***
name	List of {value-name: value} • "value-name": "SWC_NAME" "value": " [0-9a-zA-Z]{64}"	RO	M	• Provided by ***tapi-server***
resilience-type	One of {"**protection-type**": [ONE_PLUS_ONE_PROTECTION", "ONE_FOR_ONE_PROTECTION"]}	RO	M	• Provided by ***tapi- server***
reversion-mode	One of ["REVERTIVE", "NON-REVERTIVE"]	RO	M	• Provided by ***tapi- server***
hold-off-time	"[0-9]{4}"	RO	O	• Provided by ***tapi- server***
wait-to-revert-time	"[0-9]{4}"	RO	C	• Provided by *tapi-server* • This attribute is mandatory in connection objects when the reversion-mode REVERTIVE and it is different from the default value (15 min)
max-switch-times	"[0-9]{2}"	RO	O	• Provided by ***tapi- server***
is-coordinated-switching-both-ends	One of [true, false]	RO	O	• Provided by ***tapi- server***
is-lock-out	One of [true, false]	RO	O	• Provided by ***tapi- server***
is-frozen	One of [true, false]	RO	O	• Provided by ***tapi- server***
preferred-restoration-layer	List of ["PHOTONIC_MEDIA"]	RO	M	• Provided by **tapi- server**
sub-switch-control	List of {"*/config/context/connection/{uuid}/switch-control/{switch_control_uuid}/*"}	RO	C	• Provided by ***tapi-server*** • MUST appear in Top Level connections as previously described.
switch	List of { **switch** }	RO	C	• Provided by ***tapi-server*** • MUST appear in OLP connections as previously described.

Table 59: Switch parameters for UC5b.

switch	**/tapi-common:context/tapi-connectivity:connectivity-context/connection/switch-control/switch**

Attribute	Allowed Values/Format	Mod	Sup	Notes
local-id	"[0-9a-zA-Z_]{32}"	RO	M	• Provided by ***tapi-server***
name	List of {value-name: value} • "value-name": "SW_NAME" "value": " [0-9a-zA-Z_]{64}"	RO	M	• Provided by ***tapi-server***
switch-direction	One of ["BIDIRECTIONAL", "INPUT", "OUTPUT"]	RO	M	• Provided by ***tapi-server***
selection-control	One of ["LOCK_OUT", "NORMAL", "MANUAL", "FORCED"]	RO	M	• Provided by ***tapi-server***
selection-reason	One of ["LOCKOUT", "NORMAL", "MANUAL", "FORCED", "WAIT_TO_REVERT", "SIGNAL_DEGRADE", "SIGNAL_FAIL"]	RO	M	• Provided by ***tapi-server***
selected-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***
selected-route	List of {"*/tapi-common:context/tapi-connectivity:connectivity-context/connection/{uuid}/route/{local_id}/* "}	RO	C	• Provided by ***tapi-server*** • This is mandatory when using the "single-switch at the Top-Connection" mode.

6.5.3 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, ODU
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 ODU layers. Cross-connections on the digital OTN layer are used to implement dual feeding and selective receiving and protection switching is triggered by 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 15a and 15b.
Layers involved	ODU
Type	Resilience
Description & Workflow	The connectivity-service is requested between two DSR/ODU 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 (which can be potentially at the DSR layer or at any ODU layer rate) (cfr. Figure 6-36). The TAPI Client MAY delegate the protection role selection to the TAPI Server during the CS provisioning process (only specifying the main CSEPs, such as DSRs). **Note**: In TAPI v.2.1.3 the means for the client to specify protection roles is to provide additional CSEPs, referring to the relevant available SIPs. In TAPI 2.3 it shall be possible to specify routing constraints based on protection roles for this purpose when internal SIPs are not defined. In consequence, a client MAY specify additional CSEPs and associate to them different Protection Roles (e.g., including two pairs of SIPs 67.D1, 68.D1 mapped to the OTSi NEPs representing the line ports of the transponders). It is assumed that there is a 1:1 rate relationship between the client (e.g., DSR) and the OTSi (no multiplexing) and that the OTSi top-connections are not pre-existing (there is no mechanism to specify a protection role over an existing connection).

	The Connectivity Service object sent to the TAPI Server MUST include the ***tapi-connectivity:connectivity-service/tapi-topology:resilience-type/protection-type*** attribute with **ONE_PLUS_ONE_PROTECTION** attribute value. This UC is implemented following the same workflow described in "Description & Workflow" of UC1.0

6.5.3.1 Expected result [with bidirectional transitional link example]

The expected result after the creation of the eSNCP DSR/ODU Connectivity Service is represented in Figure 6-36.

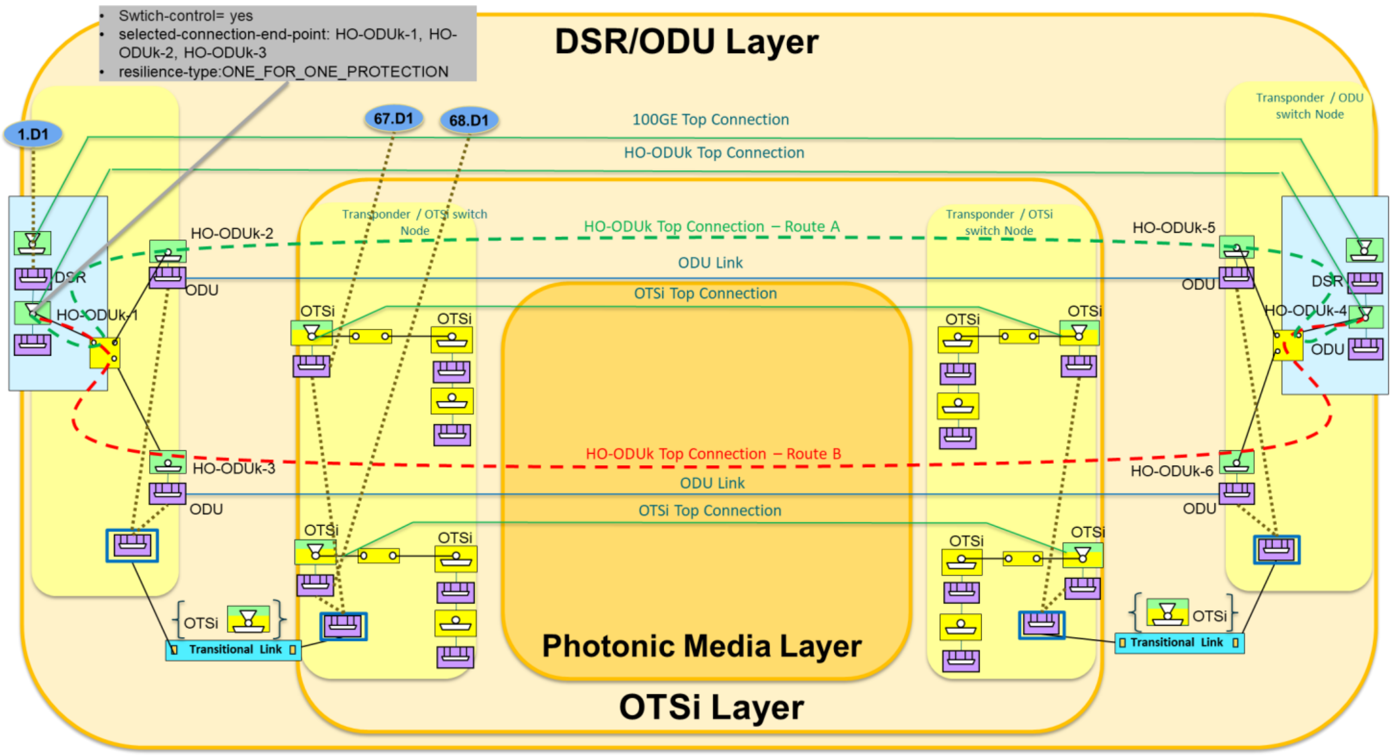


Figure 6-36 UC5c: eSNCP protection schema for HO-ODUk Top Connection [transitional link].

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:** As per "*UC5b with MC Top connection*", replacing MC by ODU, OLP by eSNCP, and only addressing the ONE_PLUS_ONE case with no intermediate ODU cross-connections considered.

  In case of ONE_PLUS_ONE_PROTECTION:

  - **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.

- **(optionally) selected-route:** Route-A (HO-ODU-Top-Connection-A) and Route-B (HO-ODU-Top-Connection-B)

Server layer connections (e.g., PHOTONIC_MEDIA Layer OTSi/MC) MAY be generated as necessary (without switch control)

6.5.3.2 Relevant Parameters

Table 60 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 60: Connectivity-service parameters for UC5c.

connectivity-service	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
resilience-type	{"**protection-type**": "ONE_PLUS_ONE_PROTECTION"}	RW	M	• Provided by *tapi-client*
preferred-restoration-layer	[ODU]	RW	M	• Provided by *tapi-client*
hold-off-time	"[0-9]{4}"	RW	O	• Provided by *tapi-client*
max-switch-times	"[0-9]{2}"	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*

6.5.4 Use case 5d: 1+1 DSR/ODU prot. with Diverse Service Provisioning (eSNCP) in Asymmetric scenarios

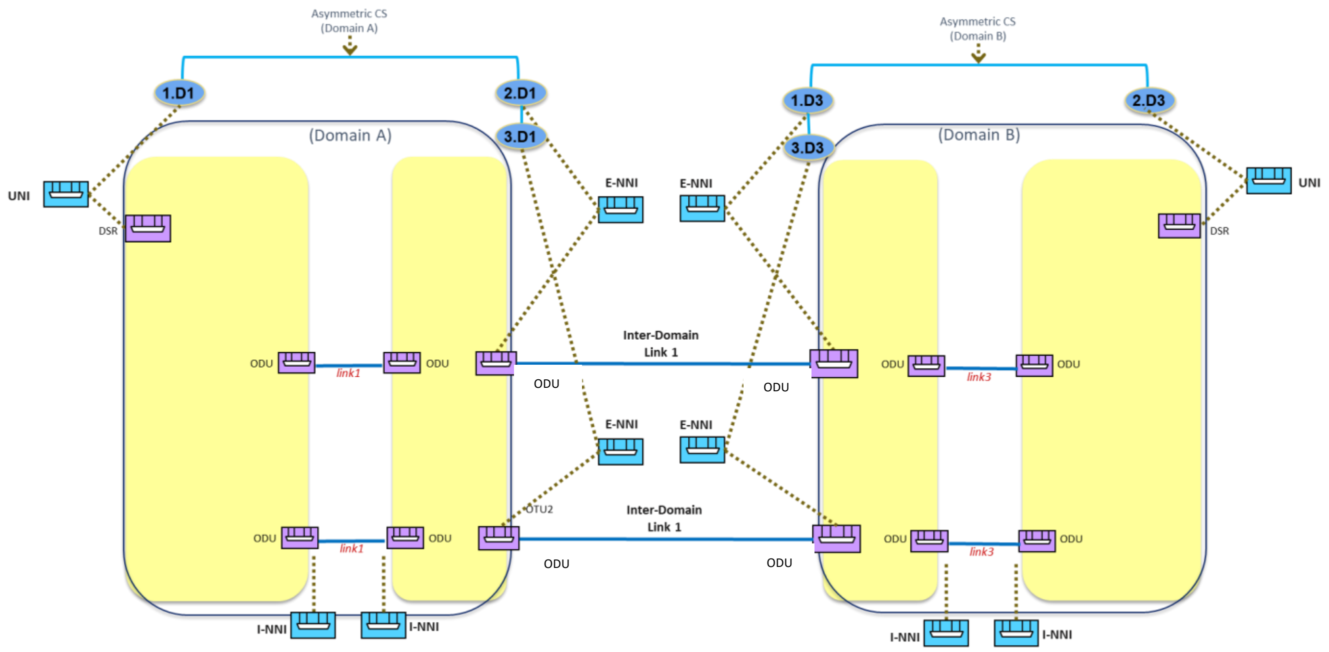
Number	**UC5d**
Name	1+1 DSR/ODU protection with Diverse Service Provisioning (eSNCP) in Asymmetric scenarios
Technologies involved	Optical
Process/Areas Involved	Planning and Operations
Brief description	**Disclaimer: This use case is in a draft state, the final definition will be completed based on 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 handover 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 15a and 15b).
Layers involved	ODU
Type	Resilience
Description & Workflow	The connectivity-service is requested between one DSR UNI CSEP and two DSR/ODU E-NNI CSEPs 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 also by implementing 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 TAPI Client MAY add two additional ODU CSEPs (referring to ODU SIPs) to define the I-NNI interfaces to implement the protection at the ODU layer (e.g., constraining the line ports of the transponders). The TAPI Client MAY associate them different Protection Roles. The attribute **tapi-connectivity:connectivity-service/is-coordinated-switching-both-ends** (boolean) MUST be included in order to track the expected behavior given the asymmetric nature of the connectivity-service. When this variable is set to false, there is not expectation of any coordinated action between the near/far end domains. The connectivity-service object MUST include the **tapi-connectivity:connectivity-service/tapi-topology:resilience-type/protection-type** attribute with **ONE_PLUS_ONE_PROTECTION** attribute value.

6.5.4.1 Detailed Workflow

The scenario assumes the boundary interfaces between network domains to be E-NNI OTUk interfaces which shall be modeled as ODU NEPs with the "inter-domain-plug-id" identifier as described in UC0d. The relationship between SIPs, NEPs and CSEPs is show in the figure below (as an example).

Disclaimer: The proposed definition for these two interfaces (UNI, E-NNI) shall be consolidated in TAPI, a possible solution is to include MEF extensions to augment current SIP definitions, however this still need to be discussed.

Thus, the initial view of the network before provisioning is depicted in the following figure:



6.5.4.2 Connectivity Service request processing

The expected TAPI Client request MUST include the relevant parameters as shown. Note that the the WORK/PROTECT CSEPs have "layer-protocol-name": "DSR" and "layer-protocol-qualifier": "tapi-dsr:DIGITAL_SIGNAL_TYPE_OTU_k" while the actual SIP (and NEP) are ODU.


```
{
"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"  (1.D1)
                    }
                },
               {
                    "direction": "BIDIRECTIONAL",
                    "protection-role": "WORK",
```

```
                        "layer-protocol-qualifier": "tapi-dsr:DIGITAL_SIGNAL_TYPE_OTU_k", (*)
                        "layer-protocol-name": "DSR",
                        "local-id": "end_point_2",
                        "service-interface-point": {
                            "service-interface-point-uuid":  "UUID2" (2.D1, ODU SIP)

                        }
                    },
                    {
                        "direction": "BIDIRECTIONAL",
                        "protection-role": "PROTECT",
                        "layer-protocol-qualifier": "tapi-dsr:DIGITAL_SIGNAL_TYPE_OTU_k", (*)
                        "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" (3.D1, ODU SIP)

                        }
                    }
                ],
                "service-layer": "DSR",
                "is-coordinated-switching-both-ends": false,
                "resilience-type":{
                     "protection-type":ONE_PLUS_ONE_PROTECTION,
                }
                "uuid": "UUID_CS",
        }
    ]
}
```

(*) The E-NNI CSEPs Protocol/Layer Qualifiers can be either ODU or DSR, the asymmetric nature of the connection being indicated by the OTN SIP.

6.5.4.3 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.

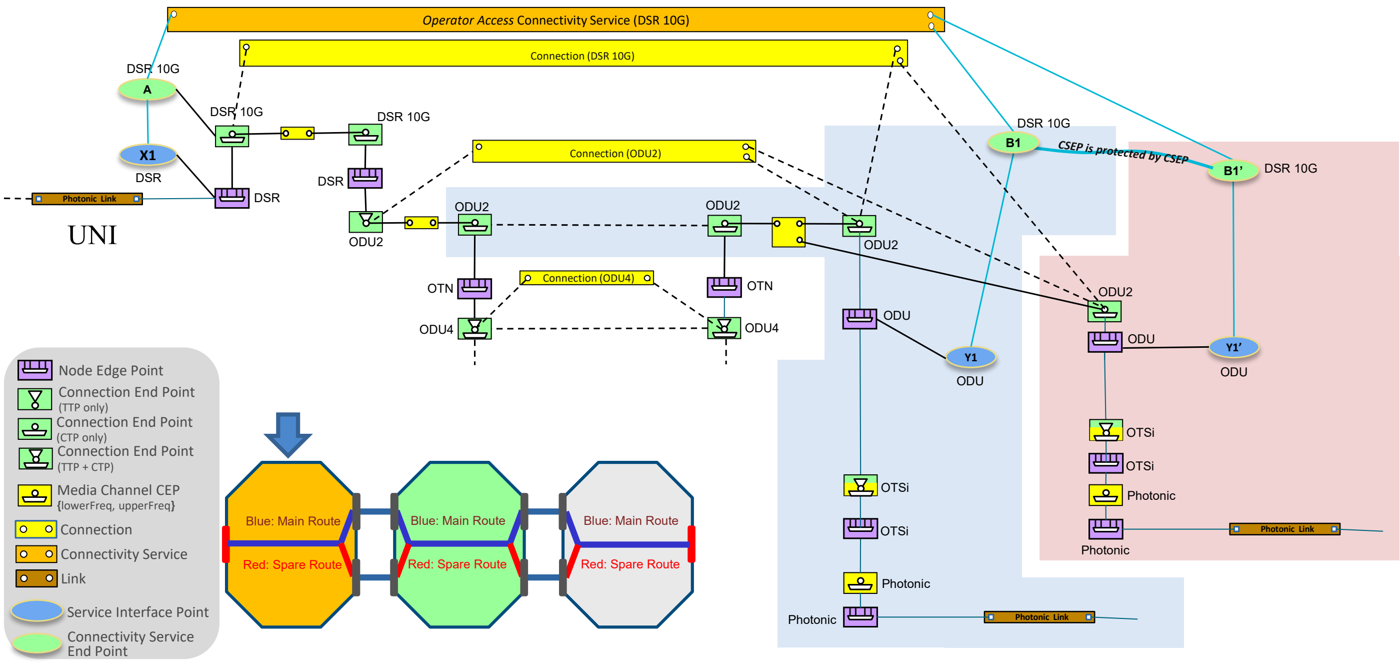


Figure 6-37 TAPI context after asymmetric connectivity-service with 1+1 Protection with Diverse Service Provisioning (eSNCP) provisioning between UNI DSR and E-NNI OTUk interfaces.

The proposed example covers an asymmetric connectivity-service request between a 10GE DSR CSEP (representing the UNI client interface in Domain A) and two ODU/DSR CSEP (representing the E-NNI OTU2, inter-domain interface at the boundary between Domain A and B). Note that the connectivity service MAY trigger the creation of the connections at the INNI-INNI interfaces (e.g., HO-ODU4, which may include underlying optical layer provisioning if required).

The requested DSR/ODU 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 by its lower-connection (by using a connection-uuid and switch-control-uuid).
    - The aEnd lower-connection switch-control includes one switch instance.
    - **aEnd ODU2 lower connection**: includes the switch control including the switch between aEnd working and protection CEPs [no sub-switch-control].
    - **N >= 0 intermediate ODU2 lower connections** without switch control.
    - **The 2 zEnds 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.
- **(optional) selected-route:** both blue main route (A) and red spare route (B) are always selected.

The TAPI server MAY implement the switch control only at ODU2 top-level connection (by using one switch and the selected-route list).

6.5.5 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	ODU, PHOTONIC_MEDIA
Type	Resilience
Description & Workflow	The connectivity service is requested between two CSEPs. The TAPI Client MAY include the ***tapi-connectivity: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/tapi-topology: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 Resiliency workflow:

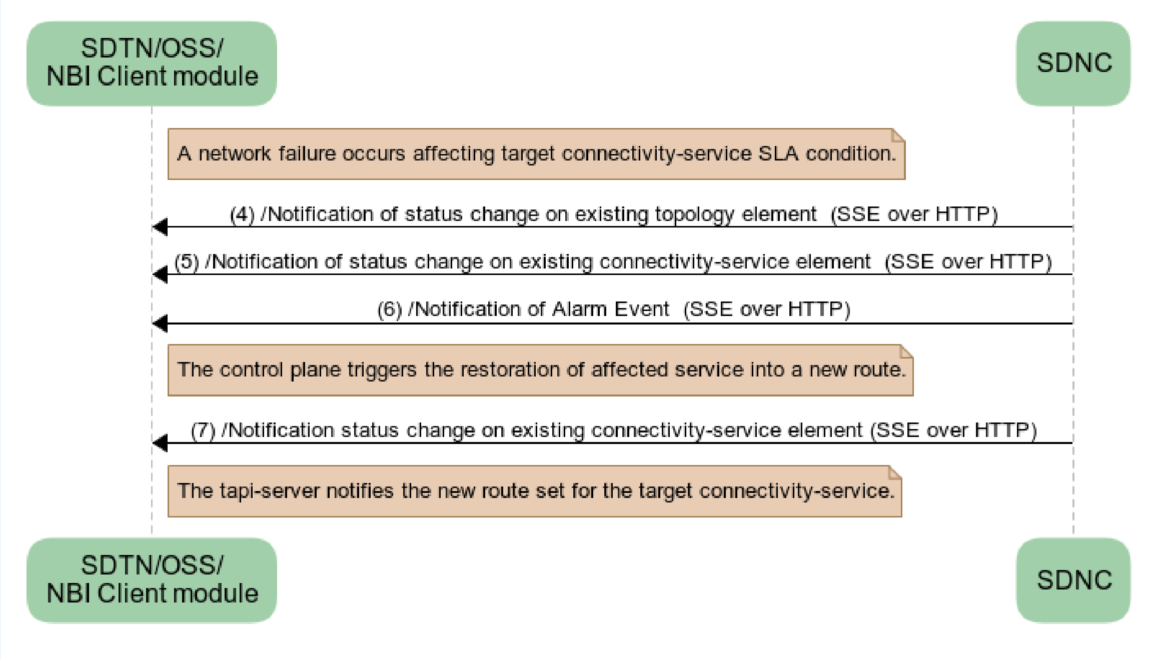


Figure 6-38 UC-6a: Resiliency workflow (note, the triggering of the restoration MAY happen prior to the notifications)

6.5.5.1 Relevant Parameters

Table 61 complements the information included in the Use Case 1.0 and Use Case 5b definitions, with the Connectivity-Service parameters required to implement this use case.

Table 61: Connectivity-service parameters for UC6a.

connectivity-service	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
resilience-type	"protection-type": "DYNAMIC_RESTORATION"	RW	M	• Provided by *tapi-client*
preferred-restoration-layer	One of [ODU, PHOTONIC_MEDIA]	RW	O	• Provided by *tapi-client*
reversion-mode	One of ["REVERTIVE", "NON-REVERTIVE"]	RW	O	• Provided by tapi-client Reversion modes for restoration (e.g., returning to the nominal path) is not specified.

6.5.6 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	Draft disclaimer: 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-topology: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 has to modify the existing Connectivity-Service object by adding the pre-computed restoration path, into the **tapi-connectivity:connectivity-service/tapi-topology:include-path** leaf-list attribute. The TAPI user MUST use the PUT operation to modify the existing object. 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.

6.5.6.1 Relevant Parameters

Table 62 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 62: Connectivity-service parameters for UC6a.

connectivity-service	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
resilience-type	"protection-type": "DYNAMIC_RESTORATION"	RW	M	• Provided by *tapi-client*
preferred-restoration-layer	One of [ODU, PHOTONIC_MEDIA]	RW	M	• Provided by *tapi-client*
reversion-mode	One of ["REVERTIVE", "NON-REVERTIVE"]	RW	M	• Provided by tapi-client
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*

6.5.7 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 PHOTONIC_MEDIA layer as the OLP Protection scheme defined in UC5b • Over the ODU 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, 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). 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, ODU, 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

6.5.7.1 Relevant Parameters

Table 63 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 63: Connectivity-service parameters for UC7a.

connectivity-service	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service**			
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	One of: [ODU, PHOTONIC_MEDIA]	RW	M	• Provided by *tapi-client*
hold-off-time	"[0-9]{4}"	RW	O	• Provided by *tapi-client*
max-switch-times	"[0-9]{2}"	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*

6.5.8 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**: TAPI 2.1.3 does not 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 ODU 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, ODU, 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

6.5.8.1 Relevant Parameters

Table 64 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 64: Connectivity-service parameters for UC7b.

connectivity-service	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service**			
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	[ODU, PHOTONIC_MEDIA]	RW	M	• Provided by *tapi-client*
hold-off-time	"[0-9]{4}"	RW	O	• Provided by *tapi-client*
max-switch-times	"[0-9]{2}"	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*

6.5.9 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.
Layers involved	DSR, ODU, PHOTONIC_MEDIA
Type	Resilience
Description & Workflow	The Connectivity Service object MUST include the tapi-connectivity:connectivity-service/tapi-topology:resilience-type/protection-type attribute with PERMANENT_ONE_PLUS_ONE_PROTECTION attribute value.

6.5.9.1 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 65: Connectivity-service parameters for UC8 (same as of 7a).

connectivity-service	**/tapi-common:context/tapi-connectivity:connectivity-context/connectivity-service**			
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
resilience-type	"**protection-type**": "PERMANENT_ONE_PLUS_ONE_PROTECTION"	RW	M	• Provided by *tapi-client*

6.5.10 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. The specific TAPI path to which these parameters refer is: `module: tapi-connectivity` `  augment /tapi-common:context:` `    +--rw connectivity-context` `       +--rw connectivity-service* [uuid]` `	+--rw reversion-mode? reversion-mode` `	+--rw wait-to-revert-time? uint64`

Layers involved	DSR, ODU, 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.

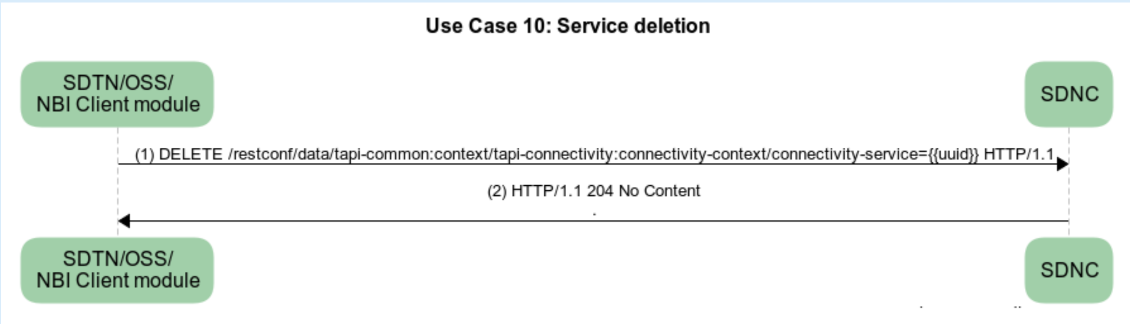
6.5.10.1 Relevant Parameters

See the introduction to the Section 6.5 for the relevant parameters.

6.6 Maintenance

6.6.1 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: - As detailed in e.g., UC1c, UC1g, UC2b, 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**]. - A connection C is "*supporting*" a connectivity service S if any of the connection CEPs (namely P) appears in the NEP/CEP hierarchy that starts at the S top-level connection. In other words, P would be modified/adjusted as part of the workflow associated to the instantiation, modification, or deletion of S [**supporting connection]**. - Because of the connectivity service(s) instantiation, a number of *supporting connections* and the corresponding related NEPs and CEPs will have been created/configured. Further connectivity service(s) provisioning/deletion MAY modify such connections. - The deletion of a connectivity service (either the client provisioned one or the server allocated ones) MUST NOT trigger the deletion of any other connectivity services. A connectivity service lifetime is always ended with a user driven delete operation **[independent deletion]**. - The deletion of a connectivity service S MUST cause the deletion of all supporting connections C that: i) are exclusively supporting S at the time of deletion [**exclusively ownership]** and ii) are not server-owned. This implies that there are no orphan connections if they were created upon the provisioning of a connectivity service [**no orphan connections**].

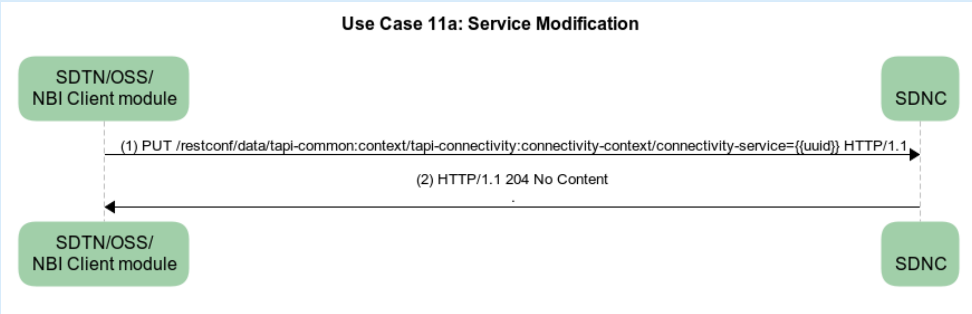
	o For example, the provisioning of a connectivity service ODU2-S1 MAY cause either 1) the instantiation of a top-level connection ODU2-C and a supporting connection ODU4-C or 2) the instantiation of a top-level connection ODU2-C, a supporting connection ODU4-C and a server allocated connectivity service 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. - The deletion of connections does not apply to those not created or deleted upon a connectivity service provisioning (e.g., OMS/OTS connections are assumed pre-existing) [**TAPI server-owned connections**]. - 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`). 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, ODU, 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 6-39 UC-10: Service Deletion workflow.

6.6.2 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.
	
	The usage of HTTP PATCH is for further study.

Layers involved	DSR, ODU, 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 removed.  Figure 6-40 UC-11a: Service Modification.

6.6.3 Use Case 11b: Modification of service nominal path to secondary (prot.) path for maintenance operations

<table>
<tr><td>Number</td><td>UC11b</td></tr>
<tr><td>Name</td><td>Modification of service nominal path to secondary (protection) path for maintenance operations.</td></tr>
<tr><td>Technologies involved</td><td>Optical</td></tr>
<tr><td>Process/Areas Involved</td><td>Planning and Operations</td></tr>
<tr><td>Brief description</td><td> 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:
<pre>module: tapi-connectivity
  augment /tapi-common:context:
    +--rw connectivity-context</pre></td></tr>
</table>

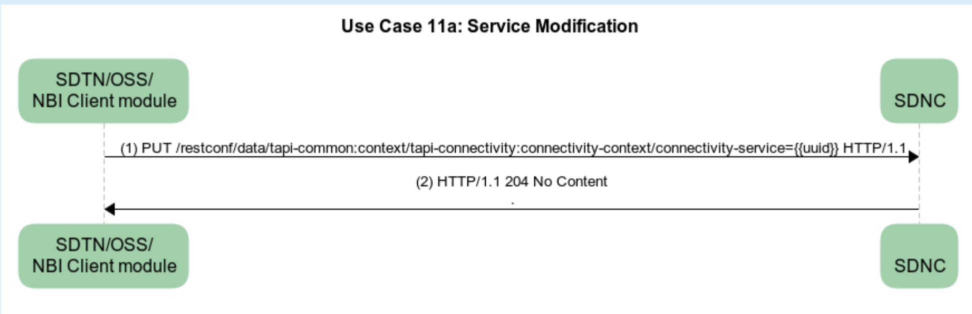
<table>
<tr><td></td><td>
<pre>
        +--rw connectivity-service* [uuid]
        | +--rw end-point* [local-id]
        | | +--rw protecting-connectivity-service-end-point
        | | | +--rw connectivity-service-uuid?   -> /tapi-
common:context/tapi-connectivity:connectivity-context/connectivity-
service/uuid
        | | | +--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
</pre>
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.
</td></tr>
<tr><td>Layers involved</td><td>DSR, ODU, PHOTONIC_MEDIA</td></tr>
<tr><td>Type</td><td>Maintenance</td></tr>
<tr><td>Description & Workflow</td><td>
The TAPI client MUST specify the tapi-connectivity:connectivity-service/uuid attribute in the RESTCONF PUT request to identify the service to be modified.


Figure 6-41 UC-11a: Service Modification.
</td></tr>
</table>

6.7 Planning

6.7.1 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, OTSi/OTSiA, 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 2.1.3, 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, ODU, PHOTONIC_MEDIA
Type	Planning

Description & Workflow

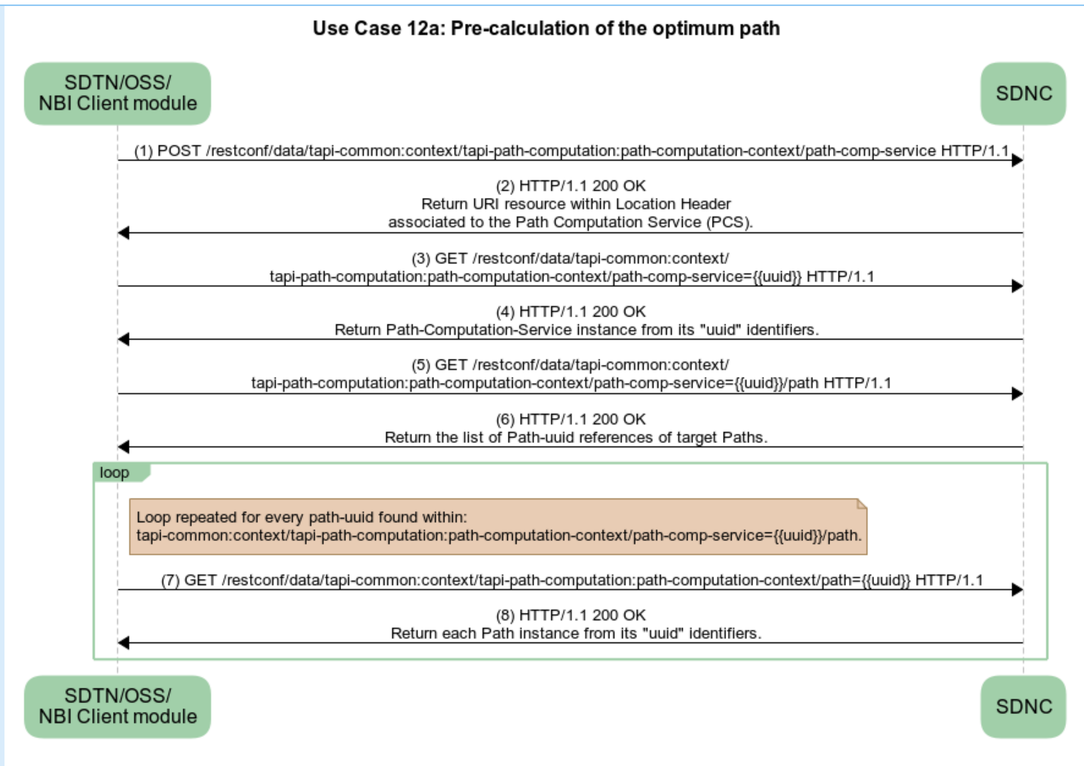


Figure 6-42 UC-12a: Pre-calculation of the optimum path workflow.

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)
```

6.7.1.1 Relevant Parameters

Table 66: 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 67: 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* • TAPI 2.1.3 specifies min and max elements 2, so only point-to-point is supported.
routing-constraint	{ **routing-constraint** }	RW	M	• Provided by *tapi-client* • For details, see Table 70
topology-constraint	{ **topology-constraint** }	RW	M	• Provided by *tapi-client* • For details, see Table 69
objective-function	{**objective-function**}	RW	M	• Provided by tapi-client • For details, see Table 71
optimization-constraint	{**optimization-constraint**}	RW	O	• Provided by tapi-client • For details, see Table 72
uuid	As per RFC4122	RW	M	• Provided by *tapi-client*
name	"value-name": "PATH_COMP_REQ_NAME" "value": "[0-9a-zA-Z_]{64}"	RW	M	• Provided by *tapi-client*

Table 68: 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	One of ["DSR", "ODU", "PHOTONIC_MEDIA"]	RW	M	• Provided by *tapi-client*
layer-protocol-qualifier	One of ["DIGITAL_SIGNAL_TYPE", "ODU_TYPE", "PHOTONIC_LAYER_QUALIFIER"]	RW	M	• Provided by *tapi-client* • All children identities defined for ["DIGITAL_SIGNAL_TYPE", "ODU_TYPE", "PHOTONIC_LAYER_QUALIFIER"] 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 69: 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 [ODU, PHOTONIC_MEDIA]	RW	M	• Provided by *tapi-client*

Table 70: Routing constraint object's parameters.

routing-constraint				
Attribute	**Allowed Values/Format**	**Mod**	**Sup**	**Notes**
cost-characteristic	{ cost-name, cost-value, cost-algorithm} • "cost-name": "string", • "cost-value": "string", • "cost-algorithm": "string",	RW	O	• Provided by *tapi-client*
latency-characteristic	{ traffic-property-name, fixed-latency-characteristic, queing-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* • NOTE: queing spelling is fixed in 2.3.1
risk-diversity-characteristic	{ 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",	RW	M	• Provided by tapi-client

	"MIN_SUM_OF_WORK_AND_PROTECTION_ROUTE_COST", "MIN_SUM_OF_WORK_AND_PROTECTION_ROUTE_LATENCY", "LOAD_BALANCE_MAX_UNUSED_CAPACITY"			
route-direction	One of ["BIDIRECTIONAL", "INPUT", "OUTPUT"]	RW	M	• Provided by *tapi-client*
is-exclusive	Boolean	RW	O	• Provided by *tapi-client*

Table 71: 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 72: 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*

6.7.2 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, ODU, 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.

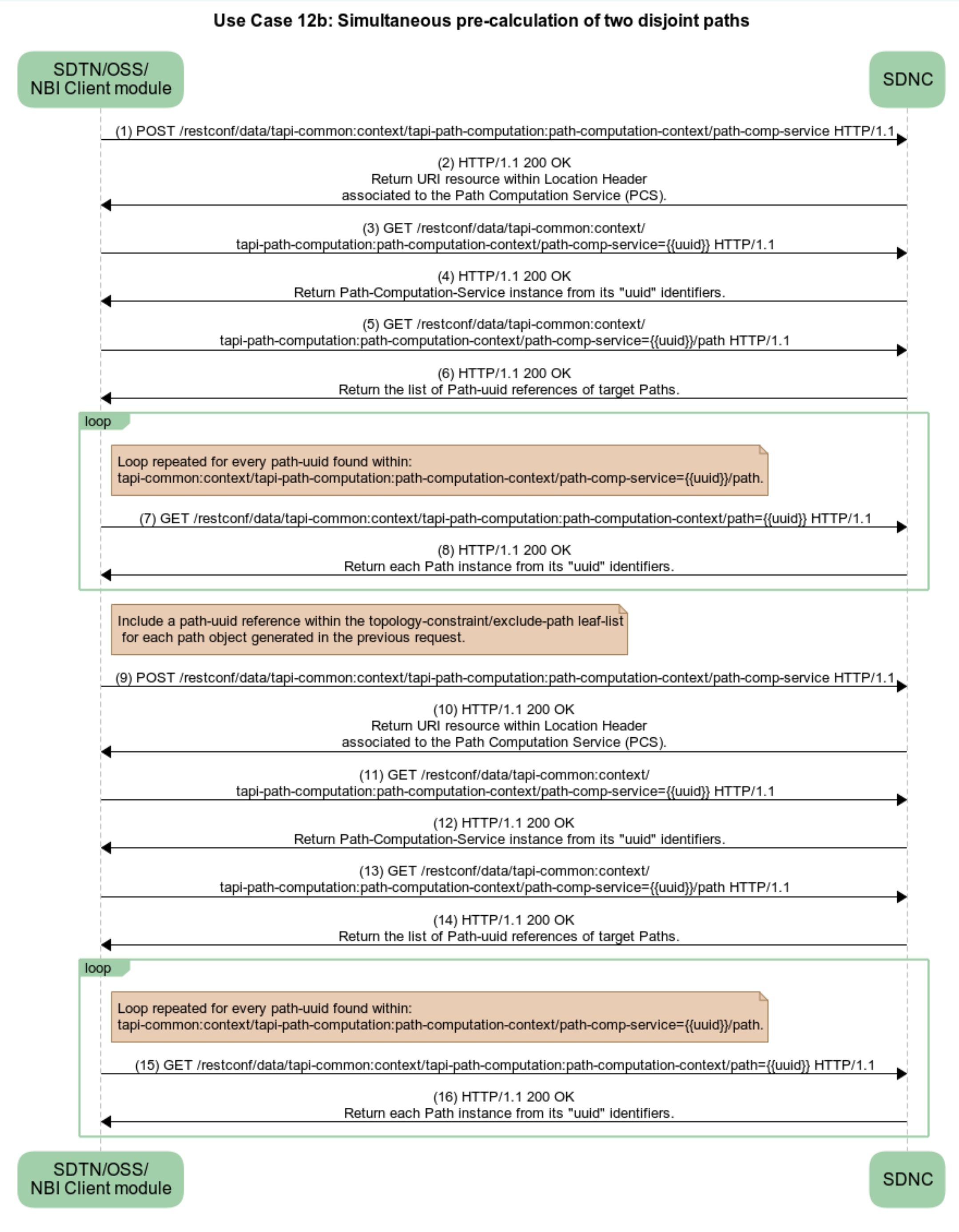


Figure 6-43 UC-12b: Simultaneous pre-calculation of two disjoint paths

6.7.3 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, ODU, 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.

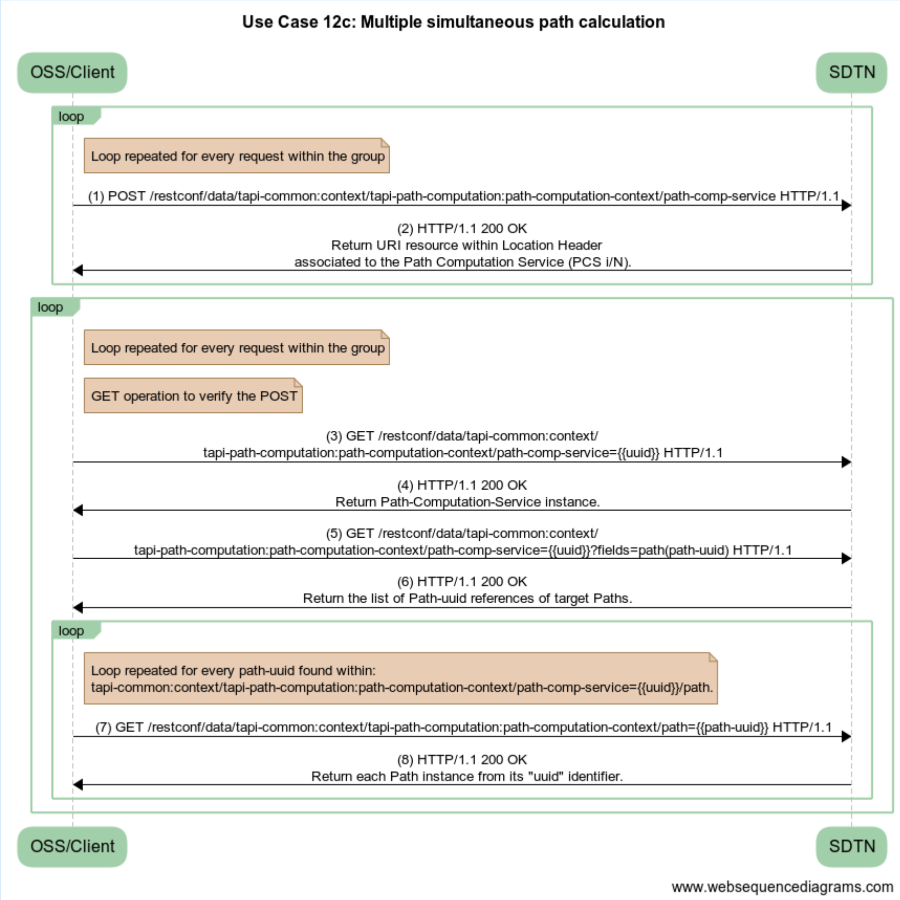


Figure 7 18 UC-12c: Multiple simultaneous path computation

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 73: 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.

6.8 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].

6.8.1 Use case 13a: Subscription to Notification service

Number	**UC 13a**
Name	**Subscription to Notification service**
Technologies involved	All
Process/Areas Involved	Planning and Operations
Brief description	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:

```
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 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):

TAPI based (creation of a "filtered stream")	**RESTCONF based (subscription)**
Creation of a filtered stream (in addition to the existing default one)	Creation of a channel (upon subscription)
notif-subscription/subscription-filter subtree	Filter query parameters

Filtering parameters:
`requested-notification-types,`
`requested-object-types,`
`requested-layer-protocols,`
`requested-object-identifier lists`
	<filter-expression> (which may include, but not limited to:
	`notification`
	`+--ro notification-type?  notification-type`
	`+--ro target-object-type? object-type`
	`+--ro target-object-identifier? tapi-common:uuid`

A new stream appears in the list of RESTCONF streams	Can be applied to the default tapi-notification stream.
POST method on the tapi-notification:notification-context including the notif-subscription obect.	GET method on the /stream/<stream-name> where stream name is either "tapi-notification" or a uuid of a TAPI created filtered stream.
Requires RESTCONF subscription	May not require to interact with TAPI notification context.

Either by creating a new stream or by applying a RESTCONF filter, the server MUST support the filtering of notifications by a combination of:

- object-type (i.e., Connectivity-Service, Connection…),
- networking layer,
- notification-types, supporting OBJECT_CREATION, ATTRIBUTE_VALUE_CHANGE, OBJECT_DELETION, ALARM_EVENT, 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.

```
/tapi-common:context/tapi-notification: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:
	`/tapi-notification:notification/notification-type='OBJECT_CREATION'`
	
	`GET /streams/tapi-notification?filter=%2Ftapi-notification%3Anotification%2Fnotification-type%3D'OBJECT_CREATION'`
	
	Example 2 filter:
	`(/tapi-notification:notification/notification-type='ATTRIBUTE_VALUE_CHANGE' and /tapi-notification:notification/target-object-type='NODE')`
	
	`GET /streams/tapi-notification?filter=%2Ftapi-notification%3Anotification%2Fnotification-type%3D'ATTRIBUTE_VALUE_CHANGE'%20and%20%2Ftarget-object-type%3D'NODE'`
	
	Example 3 filter:
	`/tapi-notification:notification/target-object-name['INVENTORY_ID']/value[contains(.,'/ne=MadridNorte')]`
	
	`GET /streams/tapi-notification?filter=%2Ftapi-notification%3Anotification%2Ftarget-object-name['INVENTORY_ID']%2Fvalue[contains(.%2C'/ne=MadridNorte')]`

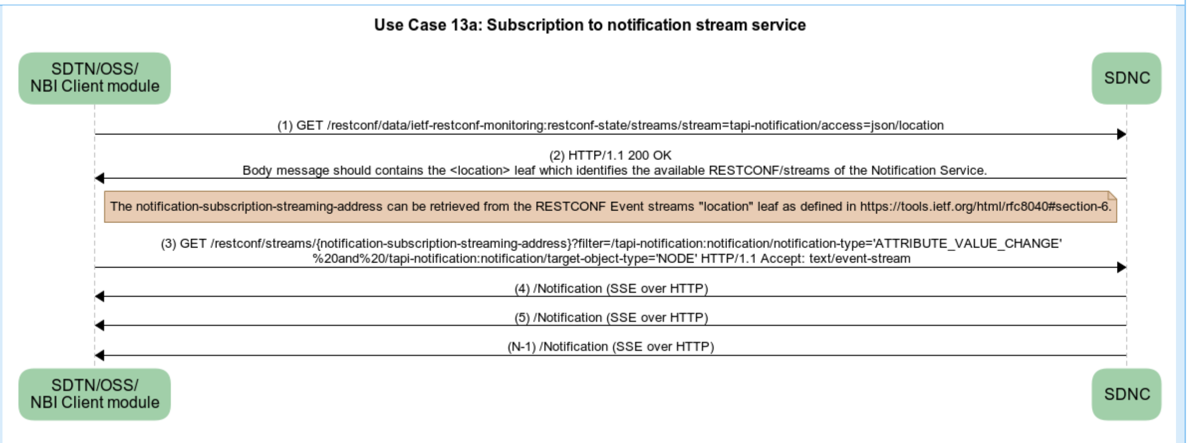
Layers involved	DSR, ODU, PHOTONIC_MEDIA
Type	Notifications and Alarms
Description & Workflow	

	Figure 6-44 UC-13a: Subscription to notification stream service

6.8.2 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 (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 information upon the use case to be implemented. **For TAPI v2.1.3, implementations MAY support additional RESTCONF filtering in the notification/alarm-info** subtree (e.g., perceived-severity, probable-cause). For TAPI v.2.3+ implementations MUST support filtering on the relevant parameters specified in UC 16a.
	Without loss of generality, for the examples please assume all notifications are defined within the custom "tapi-notification" stream.
	Example 1:
	`filter = /tapi-notification:notification/notification-type='ALARM_EVENT'`
	`GET /streams/tapi-notification?filter=%2Ftapi-notification%3Anotification%2Fnotification-type%3D'ALARM_EVENT'`
	Example 2:
	`filter = (`
	`/tapi-notification:notification/notification-type='ALARM_EVENT' and`
	`/tapi-notification:notification/target-object-type='EQUIPMENT'`
	`)`
	`GET /streams/tapi-notification?filter=%2Ftapi-notification%3Anotification%2Fnotification-type%3D'ALARM_EVENT'%20and%20%2Ftarget-object-type%3D'EQUIPMENT'`
	Example (TAPI 2.1.3)
	`filter = (`

	`/tapi-notification:notification/notification-type='ALARM_EVENT' and`
	`/tapi-notification:notification/alarm-info/perceived-severity-type='CRITICAL'`
	`)`
	`GET /streams/tapi-notification?filter=%2Ftapi-notification%3Anotification%2Fnotification-type%3D'ALARM_EVENT'%20and%20%2Falarm-info%2Fperceived-severity-type%3D'CRITICAL'`

Layers involved	DSR, ODU, PHOTONIC_MEDIA
Type	Notifications and Alarms
Description & Workflow	This UC is implemented following the same workflow described in "Description & Workflow" of UC13a

6.8.3 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 (THRESHOLD_CROSSING_ALERT)
	- 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 information upon the use case to be implemented. **For TAPI v2.1.3, implementations MAY support additional RESTCONF filtering in the notification/tca-info** subtree (e.g., threshold-crossing, threshold-parameter, perceived-severity). For TAPI v.2.3+ implementations MUST support filtering on the relevant parameters specified in UC 16b.
	Without loss of generality, for the examples please assume all notifications are defined within the custom "tapi-notification" stream.
	Example 1
	`filter = /tapi-notification:notification/notification-type='THRESHOLD_CROSSING_ALERT'`
	`GET /streams/tapi-notification?filter=%2Ftapi-notification%3Anotification%2Fnotification-type%3D'THRESHOLD_CROSSING_ALERT'`

	Example 2:
	`filter = (`
	`/tapi-notification:notification/notification-type='THRESHOLD_CROSSING_ALERT' and`
	`/tapi-notification:notification/tca-info/threshold-crossing='THRESHOLD_ABOVE'`
	`)`
	`GET /streams/tapi-notification?filter=%2Ftapi-notification%3Anotification%2Fnotification-type%3D'THRESHOLD_CROSSING_ALERT'%20and%20%2Ftca-info%2Fthreshold-crossing%3D'THRESHOLD_ABOVE'`
	
	Example 3:
	`filter = (`
	`/tapi-notification:notification/notification-type='THRESHOLD_CROSSING_ALERT' and`
	`/tapi-notification:notification/tca-info/threshold-crossing='THRESHOLD_ABOVE' and`
	`/tapi-notification:notification/tca-info/perceived-severity ='CLEAR'`
	`)`
	`GET /streams/tapi-notification?filter=%2Ftapi-notification%3Anotification%2Fnotification-type%3D'THRESHOLD_CROSSING_ALERT'%20and%20%2Ftca-info%2Fthreshold-crossing%3D'THRESHOLD_ABOVE'%20and%20%2Ftca-info%2Fperceived-severity%3D'CLEAR'`

Layers involved	DSR, ODU, PHOTONIC_MEDIA
Type	Notifications and Alarms
Description & Workflow	This UC is implemented following the same workflow described in “Description & Workflow” of UC13a

6.8.4 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* including OBJECT_CREATION, OBJECT_DELETION

	- *object-type* including TOPOLOGY, NODE, LINK, NODE_EDGE_POINT
Layers involved	DSR, ODU, PHOTONIC_MEDIA
Type	Notifications and Alarms
Description & Workflow	This UC follows the same workflow as UC13a. See Section 2.7.1.6 for relevant parameters

6.8.5 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* including OBJECT_CREATION, OBJECT_DELETION - *object-type* including CONNECTIVITY_SERVICE, CONNECTION, CONNECTION-END-POINT, SERVICE-INTERFACE-POINT
Layers involved	DSR, ODU, PHOTONIC_MEDIA
Type	Notifications and Alarms
Description & Workflow	This UC follows the same workflow as UC13a. See Section 2.7.1.6 for relevant parameters

6.8.6 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* includig OBJECT_CREATION, OBJECT_DELETION - *object-type* including CONNECTIVITY_SERVICE, CONNECTION, CONNECTION-END-POINT, SERVICE-INTERFACE-POINT
Layers involved	DSR, ODU, PHOTONIC_MEDIA
Type	Notifications and Alarms
Description & Workflow	This UC follows the same workflow as UC13a. See Section 2.7.1.6 for relevant parameters

6.8.7 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:
	`tapi-notification:notification:`
	`        +--ro source-indicator?         source-indicator`
	This UC includes UC13a where implementations MUST support the subscription including a combination of:
	• *notification-type* including ATTRIBUTE_VALUE_CHANGE
	• *object-type including* TOPOLOGY, NODE, LINK or NODE_EDGE_POINT
	The server MUST include the changed-attributes parameter in the notification.

Layers involved	DSR, ODU, PHOTONIC_MEDIA
Type	Notifications and Alarms
Description & Workflow	This UC follows the same workflow as UC13a. See Section 2.7.1.6 for relevant parameters

6.8.8 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:
	
	`tapi-notification:notification:`
	`        +--ro source-indicator?          source-indicator`
	
	This UC includes UC13a where implementations MUST support the subscription including a combination of:
	
	• *notification-type* including ATTRIBUTE_VALUE_CHANGE
	• *object-type* including CONNECTIVITY_SERVICE, CONNECTIVITY_SERVICE _END_POINT, CONNECTION, CONNECTION-END-POINT, SERVICE-INTERFACE-POINT
	
	The server MUST include the changed-attributes parameter in the notification.

Layers involved	DSR, ODU, PHOTONIC_MEDIA
Type	Notifications and Alarms
Description & Workflow	This UC follows the same workflow as UC13a. See Section 2.7.1.6 for relevant parameters

6.8.9 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 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:
	`tapi-notification:notification:`
	`        +--ro source-indicator?          source-indicator`
	
	This UC includes UC13a where implementations MUST support the subscription including a combination of:
	• *notification-type* including ATTRIBUTE_VALUE_CHANGE
	• *object-type* including ROUTE, SWITCH
	
	The server MUST include the changed-attributes parameter in the notification

Layers involved	DSR, ODU, PHOTONIC_MEDIA
Type	Notifications and Alarms
Description & Workflow	This UC follows the same workflow as UC13a. See Section 2.7.1.6 for relevant parameters

6.8.10 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 UC13a/UC13b where implementations MUST support the subscription of: • *notification-type* including ALARM_EVENT Additionally, the user may add filters based on any field of the alarm-info as detailed in Section 3.2.7.
Layers involved	DSR, ODU, PHOTONIC_MEDIA
Type	Planning and Operations
Description & Workflow	This Use Case relies in the operations and filters by notification-type (ALARM_EVENT), as defined in UC13b.

6.8.11 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 threshold crossing notifications. This UC includes UC13a/UC13b where implementations MUST support the subscription as Additionally, the user may add filters based on any field of the tca-info as detailed in Section 3.2.8 • *notification-type* including THRESHOLD_CROSSING_ALERT *NOTE: This version of the RIA does not specify how the thresholds are configured.*
Layers involved	DSR, ODU, PHOTONIC_MEDIA
Type	Planning and Operations
Description & Workflow	This Use Case relies in the operations and filters by notification-type (THRESHOLD_CROSSING_ALERT), as defined in UC13b.

6.9 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.

Use cases related to Performance and OAM are postponed to a further release of this specification and covering TAPI 2.3+

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>.
[OpenAPI]	OpenAPI Specification Version 3.0.2, <https://swagger.io/specification/>
[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://www.opennetworking.org/wp-content/uploads/2018/12/TR-512_v1.4_OnfCoreIm-info.zip
[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 Terms defined in this TR

Connection

A Connection represents an enabled (provisioned) potential for forwarding (incl. all circuit/packet forms) between two or more Node Edge Points of a Node. The bounding Node of a Connection may be explicit or be conceptually implicit.

Connection is a container for provisioned connectivity that tracks the state of the allocated resources and is distinct from the Connectivity Service request.

Connection End Point

The Connection End Point encapsulates information related to a Connection at the ingress/egress points of every Node that the Connection traverses in a Topology. Thus, they represent the ingress/egress port functions (including termination, encapsulation, processing, mapping, etc.) of the Connection.

Connectivity Service

A Connectivity Service represents an “intent-like” request for connectivity between two or more Service Interface Points. Connectivity Service is a container for connectivity request details and is distinct from Connection that realizes the request.

Connectivity Service End Point

The Connectivity Service End Point encapsulates information related to a Connectivity Service at the ingress/egress points of that Connectivity Service.

Context

The Context provides a scope of control, naming and information exchange between particular instances of controllers.

Link

A Link is an abstract representation of the effective adjacency between two or more Nodes (specifically Node Edge Points) in a Topology.

Node

The 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) belonging to those Network Resources and the potential to enable forwarding of information between those edge ports.

Node Edge Point

The Node Edge Point represents the ingress-egress edge-port functions that access the forwarding capabilities provided by the Node. Hence it provides an encapsulation of addressing, mapping, termination, adaptation and OAM functions of one or more transport layers (including circuit and packet forms) performed at the entry and exit points of the Node.

Path

The Path is described by an ordered list of Links.

Route

The Route of a Connection is modeled as a collection of Connection End Points.

Service Interface Point

A Service Interface Point represents the network-interface-facing aspects of the edge-port functions that access the forwarding capabilities provided by the Node. Hence it provides a limited, simplified view of interest to external clients (e.g. shared addressing, capacity, resource availability, etc.), that enable the clients to request connectivity without the need to understand the provider network internals.

Topology

The 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 particular set of Network Resources.

Transitional Link

Link that is formed by abstracting one or more LTPs (in a stack) to focus on the flow and deemphasize the protocol transformation. This abstraction is relevant when considering multi-layer routing.

8.3 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

End of Document