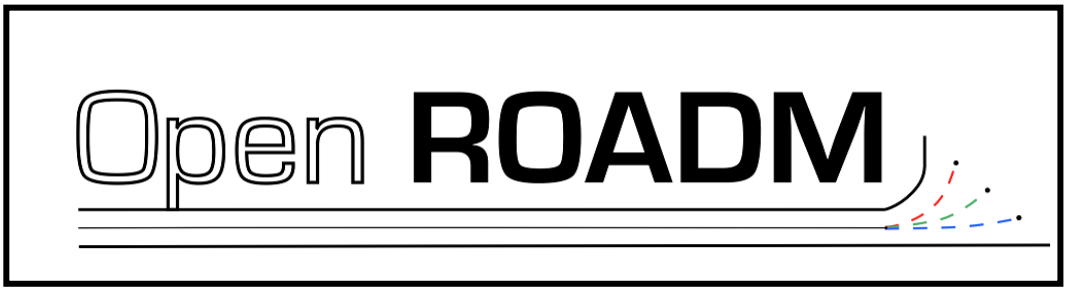


A Description of the

Open ROADM Service Model

Version 13.1 (draft)

March 19, 2024

Document Revision History

Date	Revision	Description
March 19, 2024		Added missing information to align with MSA 13.1 in Tables 2-1, 2-2, 2-3, 5-1, 5-2, 5-3, 5-4, 5-5, and 5-6. The following tables have not been reviewed or updated: tables 5-7 through 5-21 and tables 6-1 through 6-3.
July 27, 2023		Add of RPCs and notifications associated with end-terminal control for Alien Wavelength/IPoWDM use cases (PR883)
September 21, 2021		Add of Operational-mode Catalog functions (PR796), RPCs and notifications associated with optical tunnel creation for Alien Wavelength/IPoWDM use cases (PR 809)
February 21, 2021		Initial draft 7.1
August 31, 2018	0.1	Incorporate edits and comments
June 5, 2018		Initial draft

Table of Contents

List of Tables

1 INTRODUCTION

This white paper intends to provide a description of the Service Yang Model based on Open ROADM MSA version 13.1.

The Open ROADM Service Model consists of service related data stores, RPCs (Remote Procedure Calls), and notifications. It supports the RESTCONF interface between service providers' SDN Controller, OSS or Orchestrator and the ROADM Network Controller (RNC[1]) from vendors/third parties/service providers for making service creation/deletion, performing service changes such as restoration or reroute, and obtaining service related information and notifications. The high-level architecture is shown in Figure 1-1 below.

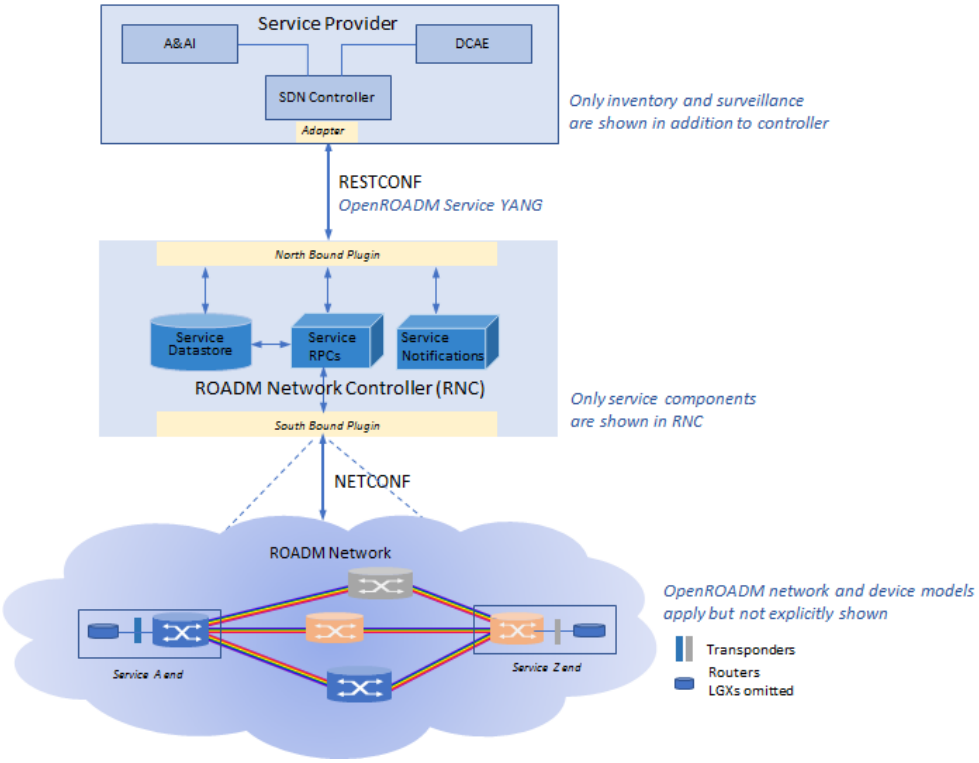


Figure 1-1 High Level Architecture of Open ROADM Service Model Application

2 SERVICE DATA STORE

Service data stores contain service list, versioned service list, and temp service list. These service lists and their parameters can be retrieved and used in various service related RPCs. Table 2-1, Table 2-2 and Table 2-3 document these 3 types of lists in the data store:

- Service list is comprised of a list of services/service names that have been requested or created in the ROADM network and their associated configuration and operational parameters
- Versioned service list adds version number(s) to the service list, while keeping the same service names as in the above-mentioned service list

[1] Also known as Open ROADM Controller.

- Temp service list represents reserved services list to be provisioned in the future. Once transitioning to a normal service, the service will be moved from the temp service list to the service list.

2.1 Service List

Services in the service list can only be created, deleted or modified using special RPCs. Service list will only contain one service with a given name. It does not contain historical (deleted or those past their end time) or temp/draft services. If two services exist with the same name (e.g., with non-overlapping start-end time), this table will contain the current one. If only planned services exist for the name, the one with the earliest start time will be present.

Table 2-1. Service List

	Parameter		***Manda-tory***	***Description***
Service List			Yes	Root of the list.
1	Services		Yes	List, parameters below will be repeated for each service.
2	Service Name		Yes	Service identifier. Unique within the context of a network, e.g., CLFI, CLCI, etc. Used as key for the services. This is reported against the service but may not get reflected in the service in the network. (string)
3	Common ID		No	Service order #, or identifier to be used by the ROADM controller to identify routing constraints received from planning applications. (string). Also used to correlate to an existing temp service when converting the temp service into a normal service.
4	Order ID		No	Service order identifier. May reflect the end customer service order (string).
5	Order note		No	Additional details associated with the service order identified by order-id (string).
6	SDNC Request Header[2]	*Request ID*	No	From original system requesting for the service. Uniquely generated by calling system. (string)
7		*RPC Action*	No	Only the RPC which generated the service will be present in the service list.[3] There are 14 types of RPC actions. Refer to Section 5 for complete list.

[2] Containers in the Yang model are shown as columns with grouped lines subdivided on the righthand side, for example, the SDNC Request Header container consists of {Request ID, RPC action, Notification URL, and Request system ID} from line 4 to line 7.

[3] Some RPCs do not generate service, for example, service feasibility check, temp service create, etc.

	Parameter			***Manda-tory***	***Description***
8		*Notification url*		No	URL for asynchronous response (string)
9		*Request System ID*		No	Identifier of application initiates the request. This identifier is used during call backs from the controller (string)
10	Service Resiliency	*Resiliency*		No	Identity ref with the following types: unprotected, unprotected-diversely-routed, protected, restorable, and external-trigger-restorable.
11		*Revertive*		No	Specifies whether the service shall revert to its initial working path after protection switching and fault conditions have cleared.
12		*Wait to restore*		No	Time delay for switching to backup path
13		*Holdoff time*		No	Time delay for reverting to initial working path
14		*Pre-calculated backup path number*		No	Provides the target number of backup paths conforming with specific engineering rules
15		*Coupled Service*	*Service index*	Yes	Service number of the service that is disjointly routed from the failed service. Service index is the key to a list of coupled services that may be used for restoration.
16			*Service name*	No	Name of the service that is disjointly routed from the failed service
17			*Common ID*	No	Common ID of the service that is disjointly routed from the failed service
18			*Version number*	No	Service version number of the service that is disjointly routed from the failed service
19	Routing Metric	*Wdm hop count*		No	The number of hops in the wdm layer will be used as a metric.
20		*otn-hop-count*		No	Number of hops in the otn layer will be used as a metric.
21		*wdm-load*		No	The load of the wdm layer will be used as a metric, to avoid using heavy loaded links.

	Parameter		***Manda-tory***	***Description***
22		*otn-load*	No	The load of the otn layer will be used as a metric, to avoid using heavy loaded links.
23		*latency*	No	Total path latency is used as a metric.
24		*distance*	No	Total path distance is the metric.
25		*wdm-TE-metric*	No	Used when routing shall be performed according to specific pre-defined TE metric. Total path metric can be calculated from OMS TE-metric attribute defined in org-openroadm-link module.
26		*otn-TE-metric*	No	Used when routing shall be performed according to specific pre-defined metric associated with OTN (OTU/ODU level).
27		*adaptation-number*	No	Adaptation between layers is the metric. Total path metric can be calculated from the total number of transitions between layers.
28	Connection Type		Yes	4 types: Service, Enum=1; Infrastructure, Enum=2; ROADM line, Enum=3; optical-tunnel, Enum=4;
29	Lifecycle State		No	Service lifecycle state, 10 types (string) Deployed, Enum=1; Planned, Enum=2; Maintenance, Enum=3; Deploying, Enum=4; Undeploying, Enum=5; Undeployed, Enum=6; Proposed, Enum=7; Draft, Enum=8; Deploy Failed, Enum=9; Un-deploy Failed, Enum=10
30	Resource status		No	(resource-status-type) Enum to indicate which network resources the controller should consider when computing a path. Options are ***deployed*** (default; use network resources with lifecycle-state=deployed and ignore operational-state), ***in-service*** (route around failed services; lifecycle-state=deployed and operational-state=inService), and ***planned*** (lifecycle-state=planned or any deployed/deploying state including deploy-failed, and operational-state is ignored; if ***planned*** if selected, due date must be specified in the service request)
31	Administrative State		No	Intended state of service (string) – enum with values of inService, outOfService, and maintenance.

	Parameter		***Manda-tory***	***Description***
32	Operational State		No	Actual state of service (string) – enum with values of inService, outOfService, and degraded.
33	Condition		No	Service condition. Additional information about the state of the service. Only sent when applicable. 5 types: Restored temporarily, Enum=1; Re-routed temporarily, Enum=2; Activated for service, Enum=3; Activated for further check, Enum=4; Activated for troubleshooting failure, Enum=5
34	Frequency		No	Used when connection-type = optical-tunnel. Represents the wavelength service frequency in THz.
35	Width		No	Used when connection-type = optical-tunnel. Represents och/otsi slot width in GHz.
36	Service A-end	*Service Format*	Yes	7 types: Ethernet, Enum=1; OTU, Enum=2; OC, Enum=3; STM, Enum=4; OMS, Enum=5; ODU, Enum=6; OTM, Enum=7
37		*Service rate*	No	E.g., 10G, 100G etc. rate in integer (uint32) Service rate not applicable when service format is roadm line (OMS) or ODU; valid for OTU since service-rate has already been supported for wdm layer OTU services (100 for OTU4)
38		*Is split lambda*	No	Boolean to indicate whether total service rate is split over multiple wavelengths or not. This attribute can be requested by the user/orchestrator or determined by the controller.
39		*split lambda service rate*	No	Applies when split lambda=true and service format is not OMS or ODU. Indicates the rate of each constituent wavelength, for instance, if the requested service rate is 400G, and the split lambda service rate is 200G, two wavelengths would be used.
40		*Other service format and rate*	No	Used when the service-format is set to other in the bookend xponder use case. The use of this attribute is not standardized in the MSA and allows the controller to support non-Open ROADM service formats. The string should include both service format and rate information.

	Parameter					Manda-tory	Description
41		*OTU service rate*				No	Full rate of transport of OTUn, e.g., OTU2, OTU4, only applicable for OTU services.
42		*ODU service rate*				No	Sub-rate ODU services, e.g., ODU0 in an OTU4 interface, only applicable for ODU services.
43		*Ethernet Encoding*				No	Type of Ethernet encoding when the rate = 10GE. 2 types: "10GBASE-W", Enum=1; and "10GBASE-R", Enum=2
44		*Mapping Mode*				No	Applies only to 10GE. "GFP-F" maps into an OPU2 with PT=5 (ITU-T G.7041 Section 7.1) "GFP-E" maps into an OPU2 with PT=9 (ITU-T G.7041 Section 7.9). Note GFP-E is an Open ROADM term to mean "Extended" OPU2 mapping "PCS-Transparent" maps into an OPU2E with PT=3 (ITU-T G.709 Section 17.2)
45		*Client phy code*				No	Identifies the client phy code to be used at the service end point so that the proper client optic is selected. Valid client phy codes are defined in org-openroadm-common-phy-codes. When requested by the user/orchestrator, the controller should verify that the client port(s) selected support the requested client-phy-code by checking the device compliance-codes.
46		*OTN attributes*	*Parent ODU allocation*	*Trib port number*		Yes	OTN attributes is used when service-format=OTU or ODU Presence container. If present, provides an explicit assignment of the parent ODU trib-slot and trib-port allocation. Trib port number is mandatory if parent-odu-allocation container is present. Identifies assigned tributary port number in parent OPU, uint16, range 1..80.
47				*Trib slots choice*	*Trib slots*	Yes	Choice container to indicate assigned trib slots occupied, either a list of OPU trib slots or a list of OPUCn trib slots. List of trib slots occupied in parent OPU multiplex structure identifier (MSI), contains 1..80 entries.

	Parameter					Manda-tory	Description
48					*Opuc n trib slots*	No	List of OPUCn 5G trib slots in the format TS#A.B. Refer to G.709 Figure 20-9.
49			*FEC*			No	Type of forward error correction; identity ref with values off, scfec, rsfec, ofec, efec, ufec, sdfec, sdfeca1, sdfecb1, baser.
50			*TX SAPI*			No	The 15-character transmit trace SAPI.
51			*TX DAPI*			No	The 15-character transmit trace DAPI.
52			*TX Operator*			No	The 32 character operator specific field.
53			*Expected SAPI*			No	Expected SAPI value, to be compared with accepted Trail Trace ID (TTI).
54			*Expected DAPI*			No	Expected DAPI value, to be compared with accepted TTI.
55			*TIM action enabled*			No	Indicates whether TTI mismatch action is enabled.
56			*TIM detect mode*			No	TTI mismatch detection mode enum: disabled (default; TTI is ignored), SAPI (expected SAPI is compared with accepted TTI), DAPI (expected DAPI is compared with accepted TTI), SAPI-and-DAPI (expected SAPI and expected DAPI are compared with accepted TTI)
57			*Degrade max intervals*			No	Number of bad intervals required to declare a degraded defect (G.806 dDEG) condition.
58			*Degrade threshold percentage*			No	Percentage of errored blocks required to declare an interval bad, in units of .01%
59			*Reserved TCM layer*			No	
60			*TCM*	*Layer*		No	Tandem connection monitoring list, indexed by layer and tcm-direction
61				*Monitoring mode*		No	Monitoring mode of the TCM layer: not-terminated, terminated, or monitored
62				*LTC activation enabled*		No	Enable or disable alarm transfer on loss of tandem connection

	Parameter				Manda-tory	Description
63				*Proactive delay measurement enabled*	No	Enable or disable proactive delay measurement for TCM
64				*TCM direction*	No	TCM direction: up-tcm (TCM termination direction toward switch fabric) or down-tcm (TCM termination direction toward facilty)
65				*TX SAPI*	No	Transmitted source access point identifier (SAPI) value; string
66				*TX DAPI*	No	Transmitted destination access point identifier (DAPI) value; string
67				*TX Operator*	No	Transmitted operator value; sting
68				*Expected SAPI*	No	Expected SAPI value, string
69				*Expected DAPI*	No	Expected DAPI value, string
70				*TIM action enabled*	No	Indicates whether TTI mismatch action is enabled.
71				*TIM detect mode*	No	TTI mismatch detection mode enum: disabled (default; TTI is ignored), SAPI (expected SAPI is compared with accepted TTI), DAPI (expected DAPI is compared with accepted TTI), SAPI-and-DAPI (expected SAPI and expected DAPI are compared with accepted TTI)
72				*Degrade max intervals*	No	Number of bad intervals required to declare a degraded defect (G.806 dDEG) condition.
73				*Degrade threshold percentage*	No	Percentage of errored blocks required to declare an interval bad, in units of .01%
74		*CLLI*			Yes	Office location. Note the CLLI must match the site associated with the device-id of this endpoint (string)
75		*Node ID*			No	Globally unique identifier for a device length "7..63"

	Parameter				Manda-tory	Description
						pattern "([a-zA-Z][a-zA-Z0-9-]{5,18}[a-zA-Z0-9])"[4] A Node ID can contain letters, numbers, and hyphens. The first character must be a letter. The last character must be a letter or number. Reported against the service but may not get reflected in the service in the network.
76		*Tx direction*	*Index*		No	Used to list TX direction attributes for each split-lambda wavelength
77			*Port*		No	Uses service port, service LGX, and service tail. From the device model perspective, the port-device-name plus the port-circuit-pack-name plus the port-name uniquely identifies the port. From the network model perspective, the openroadm-topology-ref plus port-device-name plus port-name uniquely identify the termination point in the network model.
78				*Port device name*	No	Port defined for the end-to-end service (string)
79				*Port circuit pack name*	No	Port circuit pack name for the service (string)
80				*Port circuit pack type*	No	Port circuit pack type as specified in the device model.
81				*Port type*	No	Port type, e.g. “router” or “POI” etc. (string)
82				*Port name*	No	Port index identifier. Unique within the context of a circuit-pack. E.g. Tx, Rx (string)
83				*Port rack*	No	E.g. Bay FIC: Frame Identification Code (string)
84				*Port shelf*	No	E.g. shelf in the bay (string)
85				*Port slot*	No	E.g. slot in the shelf (string)
86				*Port sub-slot*	No	E.g. sub-slot in the shelf or on a card (string)
87			*LGX*	*LGX device name*	No	E.g. name/identifier of the LGX (string)
88				*LGX port name*	No	E.g. port name of the LGX (string)

[4] The pattern for Node ID is incorrect in the Open ROADM YANG model as it doesn’t allow the length to be extended past 20 characters. This will be fixed in a future release of the YANG models.

	Parameter					Manda-tory	Description
89				*LGX port rack*		No	E.g. rack port of the LGX (string)
90				*LGX port shelf*		No	E.g. shelf port of the LGX (string)
91			*Tail*	*Tail ROADM*	*Node ID*	No	Tail ROADM: ROADM on which the Xponder is connected to (TID, IP Address, or FQDN). Node ID: Refer to line 75.
92				*Xponder port*	*Circuit pack name*	No	Tail Xponder circuit pack name/identifier (string)
93					*Port name*	No	Xponder circuit pack port name (string)
94				*Tail ROADM port AID*		No	Provide Xponder's port for intercity ROADM connection (bay, shelf, slot, and port)
95				*Tail ROADM Port Rack Location*		No	Xponder's location, e.g., FIC (Frame Identification Code) of the tail ROADM
96		*Rx direction*	For Rx direction, repeat parameters from line 76 to line 95.				
97		*Optic type*				No	2 types: Gray, Enum=1; DWDM, Enum=2
98		*Router*[5]		*Node ID*		No	Refer to line 75.
99				*IP Address*		No	Router IP address, inet:*ip-address*
100				*URL*		No	URL needed for communication with DWDM pluggable. (string)
101		*User Label*				No	Label for service endpoint, defined by the user (string)
102		*Ethernet attributes*	*FEC*			No	If present must be a valid FEC value (identiy ref).
103			*Subrate ethernet SLA*	*Committed information rate*		No	Subrate ethernet SLA is a presence container. If present, either committed-information-rate or committed-burst-size must be provided.
104				*Committed burst size*		No	Subrate ethernet SLA is a presence container. If present, either committed-information-rate or committed-burst-size must be provided.

[5] Needed for communication with DWDM pluggable.

	Parameter					Manda-tory	Description
105		*Project ID*				No	May be used to provide an identifier for the project that this service is related to.
106		*Project Note*				No	Free form string to provide additional project information.
107		*Optical attributes*	*Operational-mode*			No	The supported optical operational mode, either Open ROADM or vendor-specific.
108			*RX estimated OSNR*			No	Estimated OSNR for the path (dB); applies to xponders for an infrastructure service and to external optical transceivers for an optical-tunnel-service.
109			*RX estimated GSNR*			No	Estimated GSNR for the path (dB); applies to xponders for an infrastructure service and to external optical transceivers for an optical-tunnel-service.
110			*Max output power*			No	Maximum output power setting (dB); applies to xponders for an infrastructure service and to external optical transceivers for an optical-tunnel-service.
111			*Min output power*			No	Minimum output power setting (dB); applies to xponders for an infrastructure service and to external optical transceivers for an optical-tunnel-service.
112	Service Z-end	Repeat parameters from line 36 to line 111 for Service Z-end					
113	Hard Cons-traints[6]	*Customer Code*				No	For selecting tagged equipment on which to route a service. If more than one customer code is provided, they will be treated as an ordered list. (string)
114		*Operational mode*				No	One or more operational mode(s) can be specified to be used as a constraint. Leaflist (string)
115		*Diversity*	*Service identifier list*	*Service identifier*		No	Diverse from existing services identified by service-identifier (string), either service-name or common-id. Service identifier is mandatory if service diversity is required.
116				*Servi ce*	*site*	No	Whether service should be site (as identified by CLLI) diverse (Boolean)

[6] Routing constraints specified in the initial service creation call are hard (or strict) constraints. If no service path available, hard constraints can be relaxed for PCE to find a path. The relaxed constraints are specified as "Soft Constraints" which need to be re-evaluated.

	Parameter						Manda-tory	Description
117				*applicability*	*node*		No	Whether service should be node (as identified by node-id) diverse (Boolean)
118					*srlg*		No	Whether service should be diverse based on Shared Risk Link Group identifiers, (Boolean)
119					*link*		No	Whether service should be link diverse (Boolean)
120					*equipment*	*Roadmsrg*	No	Whether service should be ROADM SRG diverse, (i.e. service must use a different SRG), Boolean
121						*Xponderrsrg*	No	Whether service should be xponder SRG diverse, (i.e. service must use a different SRG), Boolean
122			*Diversity type*				No	Enum to indicate whether bulk services require serial or synchronous routing for diversity. The value is assumed to be the same for all services in the bulk request (either serial or synchronous).
123		*Exclude*	*Fiber bundle*				No	Fiber segment usually defined by SRLG (string), list.
124			*SRLG id*				No	Unique identifier for SRLG (integer).
125			*Site*				No	Site identifies the CLLI (string), list.
126			*Node-id*				No	Refer to line 75, list. Represents the node-id from the device model.
127			*Link identifier*	*Link network id*				List of link-identifier to exclude, indexed by link-network-id and link-id from the network model. Link-network-id is a string equal to network-id from the network model.

	Parameter				Manda-tory	Description
128				*Link-id*		Link-id from the network model (string).
129			*Supporting service name*		No	Supporting service(s) to exclude from this route (string), list. Supporting service is the service name that another service runs over top. For example, if connection-type is service, then this is the related connection-type = infrastructure service.
130		*Include*	*Is explicit routing*		No	Boolean to indicate whether include list provides a complete (explicit) route.
131			*Is include list ordered*		No	Boolean to indicate whether the include list, whether explicit or not, is provided in order of the route resources.
132			*Fiber bundle*		No	Refer to line 123 for include.
133			*SRLG id*		No	Refer to line 124 for include.
134			*Site*		No	Refer to line 125 for include.
135			*Node id*		No	Refer to line 126 for include.
136			*Link Identifier*	*Link network id*	No	Refer to line 127 for include.
137				*Link id*	No	Refer to line 128 for include.
138			*Supporting service name*		No	Supporting service(s) to include in this route (string), list. Supporting service is the service name that another service runs over top. For example, if connection-type is service, then this is the related connection-type = infrastructure service.
139		*Latency*	*Maximum Latency*		No	Maximum *latency* allowed on service (uint32), units in “ms”.
140		*Hop count*	*Maximum WDM hop count*		No	Maximum number of hops allowed at the WDM layer
141			*Maximum OTN hop count*		No	Maximum number of hops allowed at the OTN layer
142		*TE metric*	*Maximum WDM TE metric*		No	Maximum cost allowed based on cost of WDM layer links
143			*Maximum OTN TE metric*		No	Maximum cost allowed based on cost of OTN layer links
144		*Distance*	*Maximum distance*		No	Maximum distance allowed based on length of physical spans

	Parameter			***Manda-tory***	***Description***
145		*Co-routing*	*Repeat parameters from line 115 to line 122.*	No	See description for Diversity constraints in lines 115 to 122, but substitute “co-routed” for “diverse.” For example, one or more services in the service-identifier-list should follow the same route, using the same sites, nodes, srlgs, links and/or equipment as specified.
146	Soft Cons-traints	*Repeat parameters from line 113 to line 145 for soft constraints*		No	Soft constraints are best effort; the requestor prefers that they be satisfied, but the service request will not fail if they cannot be satisfied. In the service list, the requested soft constraints should be maintained regardless of whether they have been satisfied. This allows the service to be re-routed with the same set of constraints as in the original service request.
147	Due date			No	Date and time service to be turned up. If time is not specified for a given date, default to midnight. Service will be turned up immediately if no *due date* is specified. Type: yang:*date*-and-time
148	End Date			No	Date and time service to be removed. Type: yang:*date*-and-time
149	Event Horizon Start			No	Start time to ensure that the service is routable and viable. Required resources shall be considered reserved from this time. If not provided, defaults to due date. Type: yang:*date*-and-time
150	Event Horizon End			No	End time to ensure that the service is routable and viable. Required resources shall be considered reserved until this time. If not provided, defaults to end-date. Type: yang:*date*-and-time
151	NC code			No	Network Channel code applied to wavelength service only. This is reported against the service but may not get reflected in the service in the network (string).
152	NCI code			No	Network Channel Interface code applied to wavelength service only. This is reported against the service but may not get reflected in the service in the network (string).
153	Secondary NCI code			No	Secondary NCI code applied to wavelength service only. This is reported against the service but may not get reflected in the service in the network (string).
154	Customer			No	To be included in ticket information. This is reported against the service but may not get reflected in the service in the network (string).

	Parameter	*Manda-tory*	*Description*
155	Customer contact	No	Customer contact information to be included in ticket information. This is reported against the service but may not get reflected in the service in the network (string).
156	Operator contact	No	Operator contact information to be included in ticket information. This is reported against the service but may not get reflected in the service in the network (string).
157	Service layer	No	Layer associated with service. 2 types: WDM, Enum=1; OTN, Enum=2
158	CLLI network ref	No	Network-id of the clli-network layer from the network model (string)
159	Openroadm network ref	No	Network-id of the openroadm-network layer from the network model (string)
160	Openroadm topology ref	No	Network-id of the openroadm-topology layer from the network model (string)
161	Sla id	No	SLA defined for the service (string)
162	Bandwidth calendaring	No	When true, triggers the following structure allowing the description of the bandwidth calendaring options. (boolean)
163	Bandwidth calendaring parameters		Container gathering attributes describing the bandwidth calendaring options for the service.
164	Bandwidth calendaring coupled services	No	List of services that may be associated with the considered service. These services can be in service-list, temp-service-list, and versioned-service-list. The service and its coupled-service(s) may be defined on complementary time periods.
165	Service index	No	Coupled service identifier. (String)
166	Service name	No	Coupled service name for standard-service. (String)
167	Common-id	No	Coupled service identifier for temporary service. (String)
168	Version number	No	Coupled service version number for versioned service. (uint64)
169	Recurrence pattern	No	Defines a service that is active in day-of-the-week with start-time and end-time. (String)
170	recurrence id	No	Identifier of the recurrence scheme. (uint32)

	Parameter				Mandatory	Description
171			day of the week		No	Day of the week the service is active. (enumeration)
172			Start time		No	Start time for service activation. Applies to any days of the recurrence scheme. (String)
173			End time		No	Time at which the service is deactivated. Applies to any days of the recurrence scheme. (String)
174	Latency				No	Service Latency in integer (uint32), units in “ms”
175	Fiber Span SRLGs				Yes	List of shared risk link group data on fiber spans, shared risk link group identifiers (string).
176	Equipment SRGs	*SRG number*			Yes	List of shared risk link group data on equipment (string).
177	Supporting Service Name				Yes	List of supporting services. Supporting service is the service name that another service runs over top. For example, if connection-type is service, then this is the related connection-type = infrastructure service.
178	Current active path ID				No	
179	Topology	aToZ	*ID*		Yes	aToZ list. Unique identifier and used as key for this network-topology component within this service (string) Topology reports the individual hops along the service in the A to Z direction and Z to A directions. This includes both ports internal to a device and those at its edge that are available for externally connections. It includes both physical and logical ports. Physical ports are ordered with the logical ports that run over them as follows: a.\t On ingress to a node/card, physical then logical b.\t On egress to a node/card, logical then physical
180			*computation-results*			The results from path computation is added in Release 10.0
181				*rx-estimated-osnr*	No	Estimated osnr at the receiver on the path. (org-openroadm-common-link-types:ratio-dB)
182				*rx-estimated-gsnr*	No	Estimated osnr at the receiver on the path. (org-openroadm-common-link-types:ratio-dB)

	Parameter						Manda-tory	Description
183				*max-output-power*			No	Maximum output-power. (org-openroadm-common-link-types:ratio-dB)
184				*min-output-power*			No	Minimum output-power. (org-openroadm-common-link-types:ratio-dB)
185			*Subroute ID*				No	Unique identifier for the subroute or segment of the topology
186			*Previous IDs*				No	List; pointer to the previous id or set of ids that allows reconstruction of the end-to-end route from the segments.
187			*Hop Type*				No	2 types: Node external, Enum=1, the given resource is on the edge of the node and used in relationships to resources outside of the node. Node internal, Enum=2, the given resource is internally to the node.
188			*Device*	*Node ID*			No	Refer to line 75.
189			*Resource*				No	This resource identifier is intended to provide a generic identifier for any resource that can be used without specific knowledge of the resource. If selected, only one resource type will be chosen.
190			*Circuit Pack*	*Circuit pack name*			Yes, when case selected	Circuit pack, Enum=8 Circuit pack name is the circuit pack identifier. Unique within the context of a device. Same as leafref value in model, if applicable. (string)
191				*Circuit pack type*			No	Circuit pack type as defined by equipment vendor.
192				*Circuit pack product code*			No	Circuit pack product code as defined by equipment vendor.
193			*Port*	*Port*	*Circuit pack name*		Yes, when case selected	Port, Enum=7 Circuit pack name is the circuit pack identifier. Unique within the context of a device.
194					*Port name*		No	Port, Enum=7 Port name is the port identifier. Unique within the context of a circuit-pack. Same as leafref value in model, if applicable. (string)

	Parameter				Manda-tory	Description
195				*Is physical*	No	Indicates whether the port is physical or logical (Boolean).
196				*PM capable*	No	Indicates whether the port supports PM (Boolean).
197				*Alarm capable*	No	Indicates whether the port supports alarm reporting (Boolean).
198			*Connection*	*Connection name*	Yes, when case selected	Connection, Enum=5 This is used by either ROADM connection or ODU connection since they are mutually exclusive in the model. Connection name is unique within the context of a device. Same as leafref value in model, if applicable. (string)
199			*Physical link*	*Physical link name*	Yes, when case selected	Physical link, Enum=10 Physical link name is the physical link identifier. Unique within the context of a device. Same as leafref value in model, if applicable. (string)
200			*Internal link*	*Internal link name*	Yes, when case selected	Internal link, Enum=9 Internal link name is the internal link identifier. Unique within the context of a device. Same as leafref value in model, if applicable. (string)
201			*Shelf*	*Shelf name*	Yes, when case selected	Shelf, Enum=12 Shelf name is the shelf ID identifier. Unique within the context of a device. Same as leafref value in model, if applicable. (string)
202			*SRG*	*Srg number*	Yes, when case selected	Shared Risk Group, Enum=4 SRG number is the shared risk group identifier. Unique within the context of a device. Same as leafref value in model, if applicable. (uint16)
203			*Degree*	*Degree number*	Yes, when case selected	Degree, Enum=3 Degree number is the degree identifier. Unique within the context of a device. Same as leafref value in model, if applicable. (uint16)
204			*Service*	*Service name*	Yes, when case selected	Service, Enum=13 Service name is the service identifier. Unique within the context of a network. Same as leafref value in model, if applicable. (string)

	Parameter					***Manda-tory***	***Description***
205			*Interface*	*Interface name*		Yes, when case selected	Interface, Enum=11 Interface name is the interface identifier. (string)
206				*type*		Yes	Interface type, identityref from org-openroadm-interfaces, e.g. otnOtu, opticalTransport
207				*ethernet*	*speed*	No	Ethernet speed in mbps
208				*MC TTP*	*Min frequency*	No	Media channel termination point attributes Minimum frequency of media channel in THz, range covers C and L bands.
209					*Max frequency*	No	Maximum frequency of media channel in THz, range covers C and L bands.
210				*NMC CTP*	*frequency*	No	Network media channel termination point attributes. Network media channel center frequency in THz
211					*width*	No	Network media channel width in THz
212				*OCH*	*rate*	No	Optical Channel attributes Rate of the OCH, identity ref
213					*Modulation format*	No	Modulation format of the OCH, enum
214					*frequency*	No	Optical channel center frequency in THz
215					*width*	No	Optical channel width in THz

	Parameter					Manda-tory	Description
216					*Optical operational mode*	No	Optical operational mode supported by the calculated path. May be an Open ROADM operational mode or an equipment supplier-specific operational mode.
217				*ODU*	*rate*	No	ODU is a presence container, applies when the interface is an optical data unit Rate of the ODU, identity-ref (e.g. ODU4, ODU2…)
218					*Oducn n rate*	No	Applies when rate is of type ODUCn, rate of ODU container
219					*Oduflex cbr service*	No	ODUflex for CBR client signals. Refers to the ODUflex CBR service type, e.g. ODUflex-imp, ODUflex, gfp…
220					*Oduflex gfp number ts*	No	For ODUflex GFP-F mapped client signals, this is the number of tributary slots
221					*Oduflex gf pts bandwidth*	No	The tributary slot minimum bit rates in the approximated value [Mbps] given ODUflex (GFP) 100 ppm (G.709 Table 7-8)
222					*Oduflex imp s*	No	For ODUflex IMP (Idle insertion Mapping Procedure) mapped client signals, s = 2, 8, n x 5 with N >= 1 (G.709 12.2.6 & Table 7-3)
223					*Oduflex*	No	For ODUflex for FlexE-aware client signals, n in the range of 1 to 20*254 (G.709 17.12)

	Parameter					***Manda-tory***	***Description***
					flexe n		
224					*Pare nt odu alloc ation* *Repe at lines 46 thro ugh 48*	Yes if odu containe r present	See lines 46 through 48 under OTN attributes.
225				*OTU*	*rate*	No	Rate of the OTU, e.g. OTU4, OTUflex, identity ref
226					*OTU 4 mem ber id*	No	When rate is OTU4, this is the OTSi group member id
227					*Otuc n N rate*	No	When rate is OTUCn, this specifies “N”, e.g. N = 2,3,4 for 200G/300G/400G respectively
228					*Otuc n M subr ate*	No	OTUCn subrate (OTUCn-M), value of M specifies the number of active 5 Gbit/s OPUCn tributary slots (G.709 Annex H)
229				*OTSi*	*Otsi rate*	No	OTSi rate identity ref, e.g. R400G-otsi
230					*Mod ulati on form at*	No	Modulation format, enum
231					*freq uenc y*	No	Frequency value in THz

	Parameter					Manda-tory	Description
232					*width*	No	Frequency value in GHz
233					*Optical operational mode*	No	Optical operational mode supported by the calculated path. May be an Open ROADM operational mode or an equipment supplier-specific operational mode.
234				*OTSi group*	*Group rate*	No	Supported group rate, OTSi rate identity ref, e.g. R400G-otsi
235					*Group id*	No	OTSi group id; Mandatory for FlexO B100G
236			*ODU sncp pg*	*ODU sncp pg name*		Yes, when case selected	odu-sncp-pg, enum=14 Name of the ODU subnetwork connection protection protection group (string)
237			*Client sncp pg*	*Client sncp pg name*		Yes, when case selected	client-sncp-pg, enum=19 Name of the client subnetwork connection protection protection group (string)
238			*Circuit pack pg*	*Circuit pack pg name*		Yes, when case selected	circuit-pack-pg, enum=20 Circuit pack protection group name (string)
239			*other*	*Other resource id*		Yes, when case selected	other, enum=1 Resource of type not found in list Resource ID for other (string)
240			*device*	*Node id*		Yes, when case selected	device, enum=2 ROADM, Xponder, etc., Node ID is a globally unique identifier for a device. Same as leafref value in model, if applicable.
241			*Line amplifier*	*Amp number*		Yes, when	line-amplifier, enum=15

	Parameter				Manda-tory	Description
					case selected	Amp number is the number of the line amplifier. (uint8)
242			*xponder*	*Xpdr number*	Yes, when case selected	xponder, enum=16 Xponder resource, e.g. transponder, muxponder, switchponder, regen Xpdr number is the number of the Xponder. (uint16)
243			*Versioned service*	*Versioned service name*	Yes, when case selected	Versioned service, Enum=17 Versioned service name is the versioned service identifier. Unique within the context of a network. Same as leafref value in model, if applicable. (string)
244				*Versioned service number*	Yes, when case selected	Versioned service, Enum=17 Version number of the service (uint64)
245			*Temp service*	*Common ID*	Yes, when case selected	Temp service, Enum=18 Common ID is the temp service identifier. Unique within the context of a network. Same as leafref value in model, if applicable. (string)
246		*Resource Type*	*Type*		Yes	Resource type, refer to line 188 to line 244 for Enum value.
247			*Extension*		No	Populated when resource type not defined or when Enum value is set to 'other' (string)
248		zToA Repeat parameters from line 179 to line 247 for zToA.				zToA list.
249	Backup topology					Backup topology is introduced in R 3.1.1 of the service model
250		Backup path			No	List of backup paths.
251			backup path id		No	Backup path identifier. (string)
252			failure case id		No	Considering photonic or OTN dynamic restoration where protection resources are shared, resource planning can be done by simulating failures and

	Parameter					*Manda-tory*	*Description*
							dimensioning network trunk links according to the maximum load those may experience across different failure scenarios. One failure scenario is associated to a failure case id (simulation of one equipment, node or link failure). A backup path can (optionally) be associated with a failure-case-id which corresponds to a specific failure. (String)
253			aToZ : Repeat parameters from line 179 to line 247				See description for lines 179 to 247.
254			zToA : Repeat parameters from line 179 to line 247				See description for lines 179 to 247.
255	Net-work topolo-gy	a-to-z	ID			Yes	List of network topology resources, indexed by ID.
256			Subroute ID			No	Unique identifier for the subroute or segment of the topology
257			Previous IDs			No	List; pointer to the previous id or set of ids that allows reconstruction of the end-to-end route from the segments.
258			Network resource	Net work reso urce tp	TP netw ork ID	Yes, when case is selected	Network resource, identified by either network-resource-tp or network-resource-link tp-network-id is the network-id from the network model
259					TP node ID	Yes, when case is selected	tp-node-id is the node-id from the network model
260					TP ID	Yes, when case is selected	tp-id from the network model
261				Net work reso urce link	link netw ork ID	Yes, when case is selected	link-network-id is the network-id from the network model
262					Link ID	Yes, when	link-id from the network model

	Parameter					Manda-tory	Description
						case is selected	
263			Network resource type			Yes	Identity-ref, either network-resource-tp or network-resource-link
264		z-to-a	Repeat lines 255 to 263.				See description for line 255 to 263.
265	Network backup topology		Backup path	Backup path ID		Yes	List of pre-calculated backup paths, indexed by backup-path-id (range 1 – 255).
266				Failure case ID		No	Failure-case-id may be used to associate a backup-path with a specific type of failure, e.g. OMS, SRLG, etc.
267				a-to-z Repeat lines 255 to 263.			See description for line 255 to 263.
268				z-to-a Repeat lines 255 to 263.			See description for line 255 to 263.
269	Is Bandwidth Locked					No	Boolean (true or false), default is "false". Bandwidth lock indicates whether the service is administratively prohibited from taking on more capacity, i.e., whether it can be used as a supporting service in any new service creations. Unlike administrative status, this does not impact any previous planned or deployed services.

2.2 Versioned Service List

Versioned service list contains versioned services, regardless of their lifecycle state. Services in this list can only be created, deleted, modified, etc. using special RPCs. The list can report more than one version of a service when supported by the implementation. It may contain deleted services, multiple versions of the same service, as identified by its name.

Table 2-2 Versioned Service List

	Parameter		Manda-tory	Description
Versioned service list			Yes	Root of the list.
1	Services	*Version number*	Yes	Version number is required in this case. Service-name version-number as key. (uint64)

2	Repeat parameters from line 2 to line 269 in Table 2-1. Service List		

2.3 Temp Service List

Temp service list is a list of temporary services. Services in the temp service list can only be created, deleted or modified using special RPCs.

Table 2-3 Temp Service List

	Parameter				***Manda-tory***	***Description***
Temp service list					Yes	Root of the list.
1	Services	*Service Name*			No	List, service name in this case is optional. Refer to Table 2-1, line 2.
2		*Common ID*			Yes	Common ID is required as key for temp service, see description in Table 2-1, line 3.
3	Repeat parameters from line 4 to line 111 in Table 2-1. Service List					See descriptions in Table 2-1. Service List for common service attributes and service-a-end container.
4			Equipment required		No	Added to service-a-end container. List of equipment required for this temp service, indexed by equipment identifier
5				Equipment type	No	Type of equipment, value is derived from the equipment-type grouping in the common model
6				Equipment identifier	Yes	Unique equipment identifier, string
7				Lifecycle state	No	Lifecycle-state enum from common model (deployed, planned, deploying, undeploying, proposed, etc.)
8				Equipment rack	No	Rack identifier, string
9				Equipment shelf	No	Shelf identifier, string
10				Equipment slot	No	Slot identifier, string
11				Equipment sub-slot	No	Sub-slot identifier, string
12				Is reused	No	Boolean; if true indicates that required equipment is being reused

13				port			List of ports, indexed by circuit-pack-name and port-name	
14					Circuit pack name		Circuit pack identifier, string	
15					Port name		Port identifier, string Port is unique within the context of circuit-pack	
16					Lifecycle state		Lifecycle-state enum from common model (deployed, planned, deploying, undeploying, proposed, etc.)	
17	Repeat parameters from line 36 to line 111 in Table 2-1. Service List						See descriptions in Table 2-1. Service List for service-z-end container.	port
18			Equipment required Repeat parameters from line 4 -16				Added to service-z-end container. List of equipment required for this temp service, indexed by equipment-identifier.	
19	Repeat parameters from line 113 to line 269 in Table 2-1. Service List						See descriptions in Table 2-1. Service List for hard constraints, soft constraints, and topology containers.	
20	Intermediate site						List of intermediate sites, indexed by CLLI (site identifier).	
21		CLLI				Yes	Unique site identifier, string	
22		Node					List of nodes within a site/CLLI, indexed by node-id	
23			Node ID			Yes	Globally unique identifier for a device length "7..63" pattern "([a-zA-Z][a-zA-Z0-9-]{5,18}[a-zA-Z0-9])"[7] A Node ID can contain letters, numbers, and hyphens. The first character must be a letter. The last character must be a letter or number. Reported against the service but may not get reflected in the service in the network.	

[7] The pattern for Node ID is incorrect in the Open ROADM YANG model as it doesn't allow the length to be extended past 20 characters. This will be fixed in a future release of the YANG models.

24			Equipment required			No	List of equipment required, indexed by equipment identifier
25				Equipment type		No	Type of equipment, value is derived from the equipment-type grouping in the common model
26				Equipment identifier		Yes	Unique equipment identifier, string
27				Lifecycle state		No	Lifecycle-state enum from common model (deployed, planned, deploying, undeploying, proposed, etc.)
28				Equipment rack		No	Rack identifier, string
29				Equipment shelf		No	Shelf identifier, string
30				Equipment slot		No	Slot identifier, string
31				Equipment sub-slot		No	Sub-slot identifier, string
32				Is reused		No	Boolean; if true indicates that required equipment is being reused
33				port		No	List of ports, indexed by circuit-pack-name and port-name
34					Circuit pack name	Yes	Circuit pack identifier, string
35					Port name	Yes	Port identifier, string Port is unique within the context of circuit-pack
36					Lifecycle state	No	Lifecycle-state enum from common model (deployed, planned, deploying, undeploying, proposed, etc.)
37	Supporting service hierarchy						Lists all the supporting services, indexed by service-identifer. Allows for a hierarchy of supporting services.
38		Service identifer				Yes	Unique service name
39		Service layer				No	Service-layer-type, wdm or otn Default is wdm
40		Service A end				No	A-end CLLI of the supporting service
41			CLLI			No	Unique site identifier (CLLI)
42		Service Z end				No	Z-end CLLI of the supporting service

43			CLLI		No	Unique site identifier (CLLI)
44		Supporting service				List of supporting services, indexed by ID
45			Service identifier		No	Unique service name
46			ID		Yes	Index for list of supporting services, string
47			Service A end			
48				Service format	Yes	Service-format enum (Ethernet, OTU, OC, STM, OMS, ODU, OTM, other)
49				Service rate	No	Rate of supporting service in GBps, not applicable when service-format is OMS or ODU
50				Other service format and rate	No	Used when service-format is other, allows support of non-Open ROADM service formats such as with bookended xponders
51				OTU service rate	No	Rate applicable to OTU services (identityref)
52				ODU service rate	No	Rate applicable to ODU services (identityref)
53				CLLI	No	CLLI/site identifier of the supporting service
54				Node ID	No	Node-id of supporting service
55			Service Z end Repeat lines 32 - 38			See description for lines 32 - 28
56		Transport assignment				
57			Media Channel TTP		No	Media channel trail termination point attributes
58				Minimum frequency	No	Minimum frequency of media channel in THz, range covers C and L bands.
59				Maximum frequency	No	Maximum frequency of media channel in THz, range covers C and L bands.
60			Network media channel CTP			List of network media channel connection termination points, indexed by ID
61				ID	Yes	Index for list of CTPs

62				Frequency			No	Network media channel center frequency in THz
63				Width			No	Network media channel center frequency in THz
64			ODU allocation					Parent ODU allocation
65				Trib port number			No	Trib port number; identifies assigned tributary port number in parent OPU, uint16, range 1..80.
66				Trib slots choice			No	Choice container to indicate assigned trib slots occupied, either OPU trib slots or OPUCn trib slots.
67					OPU	Min trib slot	No	Minimum assigned trib slots occupied in parent OPU MSI
68						Max trib slot	No	Maximum assigned trib slots occupied in parent OPU MSI
69					OPUCn	OPUCn min trib slot	No	Minimum OPUCn 5G trib slots in the form of TS #A.B (G.709)
70						OPUCn max trib slot	No	Maximum OPUCn 5G trib slots in the form of TS #A.B (G.709)
71		Is reused					No	Boolean to indicate whether the transport assignment resources are being reused.
72	Existing service attributes							Used when the temp service create is performed on an existing service to describe attributes related to the existing service.
73		Is existing					No	Boolean; if true, the temp service create is for an existing service, with or without new constraints.
74		Existing service name					No	When is-existing = true, provides the name of the existing service.
75		Reuse existing resources					No	If true, the temp service should use the existing service resources/equipment wherever possible.
76		Reusable existing resources					No	When reuse-existing-resources is true, this provides a list of existing resource types that may be reused. Defined by typedef existing-resource-reuse-type (values are

				regenerator, wavelength, spectrum-portion, xponder, all)

3 CONTROLLER BEHAVIOR SETTINGS

A controller-behaviour-settings container has been added in the R 7.0 service model to store information related to the controller settings, that operators may will to adjust to their needs.

As per today, considering a limited automation, services are provisioned respecting rules defined in various documents and procedures. These rules can be general guidelines corresponding to specific process defined by the operators, engineering rules / best practices that guaranty system operation from day one (limited number of services, Beginning Of Life physical characteristics) until its End Of Life (full load, degradation of line physical characteristics…). In the context of automation, these rules must be translated into models/parameters allowing to set the global behavior of the Network Controller. The role of the controller-behaviour-settings container is to store all these parameters in the controller Data Store, and notably information about:

- non rpc related notifications toward higher layer network controller,
- the way the spectrum shall be allocated considering specific operator engineering rules or specific requirements
- the generation of warning/alarms according to predefined margins
- main parameters associated with restoration and backup path calculation
- SLA definitions

The operator can use the specific RPC "Controller-parameter-settings" to set associated parameters in the data store at controller initialization.

3.1.1 Non rpc related notifications setting

A list of urls to be used for notification to a hierarchical controller and/or more generically an OSS that might be built from several components can be provided. The key used for this list is based on notification-types (service-state-change, topology-change, exceeded-attenuation-crossing-warning, insufficient-margin-crossing-alarm, autonomous-optical-restoration-triggered). The child leaflists events-triggering/disabling-notifications can be used as filters for the event that will be used by the controller to trigger notifications associated with one notification type, or to disable them (allowing a first level of correlation). The current events are currently considered : roadm-to-roadm-link-creation, roadm-to-roadm-link-deletion , otn-link-creation, otn-link-deletion, xpdr-in-out-link-creation, xpdr-in-out-link-deletion, link-state-change, otn-link-bandwidth-usage-change, node-creation, node-deletion, node-state-change, tp-creation, tp-deletion, tp-state-change, ila-state-change, none, all.

3.1.2 Spectrum allocation

Path computation functions include routing, spectrum and mode assignment (RSMA). Some operators may use different rules for spectrum assignment, dictated by engineering rules in order to improve performances or to facilitate service management. As an example some parts of the spectrum might be reserved for specific services such as infrastructure services or wholesale services. Several rules might be applied to implement wavelength assignment policy defined by the operators. Thus these rules are defined through a list. The priority gives an indication to the controller on the way rules shall be applied.

A RSMA policy can be defined for a specific rule which itself applies to a predefined spectrum portion:

- maximize capacity, can be selected to optimize the spectrum occupation. The expected behavior is that the controller will try to use the modulation format that reduces the channel width, to the cost of potential regenerations,

- maximize reach, can be selected to reduce the number of regeneration. The expected behavior is that the controller will try to reduce the number of regeneration, while keeping the maximum throughput if end Xponders allow using different modulation formats,
- minimize margin is used to optimize performances while lowering margins. This presumes that the network will be handled dynamically with potential rerouting when the network deteriorates (aging hardware, fiber repairs,..),
- maximize-margin is used to maximize the OSNR margins
- customer-spectrum-partitioning is used to reserve a part of the spectrum to a list of specific customers defined in dedicated-customer leaflist and oppositely, to forbid the use of a part of the spectrum to a list of specific customers defined non-authorized-customer leaflist,
- fragmentation-limiting partitioning is used to dedicate part of the spectrum to multiples of predefined spectral width, in order to reduce fragmentation. Dedicated-signal-bandwidth-multiple defines the granularity of the width for the channels to be provisioned in the spectrum portion. As an example a spectrum portion can be dedicated to channels with a 50 GHz multiple spectral occupation, whereas another portion of the spectrum is dedicated to channels with a 75 GHz multiple spectral occupation in dedicated-signal-bandwidth-multiple attribute. As an example a part of the specrtrum could be dedicated to channels that occupy a multiple of 50GHz bandwidth, whereas some others might be dedicated to 87,5 GHz channels.

3.1.3 Margin definition

Two kinds of margins are considered. Margins to be applied on fiber attenuation and margins to be applied on calculated OSNR values during path computation. The deployed optical infrastructure has been designed considering that the fiber attenuation at the time the design is performed, does not correspond to the span-loss fibers will experienced at the End Of Life (EOL) of the WDM network. The minimum-fiber-attenuation-bol-margin attributes, provides to the controller an information on the margin that is considered in the design phase. This information will be used to define whether the measured span-loss are compatible with the engineered-spanloss used at the design time. This provides the ability to the controller to raise warnings or alarms when physical characteristics of the network drift. 2 threshold are defined to raise alarm at the commissioning time or warning during the network life when the measured values are not in line with the engineering of the network:

_The RNC shall raise an ALARM to the northbound Controller through an insufficient-margin-crossing-alarm as soon as

(span-loss-base + minimum-fiber-attenuation-bol-margin) - engineered-spanloss > threshold-observed-vs-design-margin

_The RNC shall raise a WARNING to the northbound Controller through an exceeded-attenuation-crossing-warning as soon as: (engineered-spanloss - spanloss-current) < threshold-observed-vs-design-attenuation.

OSNR margins to be applied may differ according to the modulation and the rate. Thus margins are provided through a list, stating the line-rate and the modulation format they apply to.

3.1.4 Metrics policy

Metrics in the service model are provided with a priority (integer). They can be interpreted in two different ways. The composite-metric-versus-selective attributes allows defining the way the path-computation element shall interpret defined priorities.

When set to false, the metric shall be considered as selective: the metric of highest priority is used for path computation. If several computed paths have the same metrics, the selection of the best path will be made considering the metric of immediate lower priority. This can be applied recursively if the secondary metric does not allow to make a choice.

When set to true, the metric shall be considered as composite: the hybrid metric of the path will be calculated based on a weighted sum of the metrics according to the defined priorities (>0). OpenROADM does not specify the way weights shall be defined from the priorities, meaning some additional coefficients may be applied to define how the composite metric is calculated.

3.1.5 Default behavior

Default behavior provides parameters associated with the restoration process. All the parameters can be used as a reference in case the corresponding optional attributes are not provided in the regular rpcs. If these attributes were provided in a service rpc they would prevail on the default parameters we define here.

3.1.6 Sla-definition

Service level Agreements (SLAs) are introduced in Release 7. SLA list allows operators to define different service profiles which fit with their customer requirements. Preemption defines whether a service can or can not be preempted, in case of contention while rerouting services during failure recovery. Restoration-priority defines the order that the controller shall follow in sequentially rerouting services affected by a failure.

3.1.7 Failure-case-list

Operators generally rely on planning tools to plan their network, when advanced protection/restoration mechanism are used to optimize network dimensioning. A way to evaluate the size of the network trunks required between ROADMs or OTN switches consist in simulating failures and reroute services according to their SLA. Each failure case corresponds to a specific scenario, trunks' sizing, and routing of the different services. Failure can be applied at different levels (links, cllis, nodes, ports …), but not all the failure have the same probability to occur. As an example, an operator could decide to dimension its network so that it can restore only a subset of the services (some SLAs), in case of single failures, that only affects links (the highest probable failure case). In this case, if the number of links is limited, it could make sense to store the backup paths of the different services which have been rerouted and associate them with a failure-case-id.

The failure case list allows to describe, for different failure cases identified through a failure-case-id, the corresponding failure which has been simulated. An operator has the ability to provide for each of the declared failure cases the list of network elements (as they appear in the network-topology) that leads to the failure when going out of service; though the controller-parameter-settings rpc. It is the controller responsibility, when it calculates several backup-paths for a service during path computation, to rely on that list and provide a failure-case-id in the backup-path of the backup-topology/ network-backup-topology associated with a service. This failure-case-ids stored with the backup paths in the service-lists can then be used by the controller at a later step, when a failure occurs for restoration purposes. The controller will be able to make the connection between a failure, the failure-case-id and the potential impacted services.

The following table shows the tree-view model of the controller-behaviour-setting container.

Table 3-1 Controller Behaviour Setting Structure

	Parameter	***Manda-tory***	***Descriptions***
non-rpc-related-notification-settings			Defines url, notification-type and type of events that shall / shall not trigger notifications.
1	non-rpc-related-notification-url-list		List used to record url to be used when forwarding non rpc related notifications.
2	non-rpc-related-notification-type	Yes	The type of notification will trigger specific leaves. (non-rpc-related-notification-type)

	Parameter	*Manda-tory*	*Descriptions*
3	notification-url	No	Defines the url the notification is sent to. (String)
4	events-disabling-notification	No	For notification associated with topological changes, list of events for which notifications to SDN-Controller are not desirable. Leaflist (notification-events)
5	events-triggering-notification	No	List of events for which notifications to SDN-Controller are desirable. Has the priority with regards to events-disabling-notification. Leaflist (notification-events)
spectrum-filling			Spectrum may be filled according to specific rules to limit partitioning or to dedicate part of the spectrum to some specific clients (customer-code)
6	spectrum-filling-rules		List defining a set of rules used to fill the spectrum
7	rule-id		Identifier of the rule to be applied (Uint16)
8	priority		The rules shall be applied sequentially according to their defined priority (highest:1, lowest:255). Rules with highest priority will be applied first (Uint8)
9	RMSA-policy		Defines the Routing, Spectrum & Mode assignment policy to be applied. (rmsa-policy)
10	spectrum-range-of-appliance		Triggered when RMSA-policy is set to customer-spectrum-partitioning or fragmentation-limiting-partitioning
11	spectrum-portion-id	No	Identifier of a spectrum portion. (Uint8)
12	start-edge-frequency	No	lowest frequency to start filling the spectrum, or a spectrum portion if stop-bandwidth is also defined (org-openroadm-common-optical-channel-types:frequency-THz)
13	stop-edge-frequency	No	Associated with a start frequency, states the upper frequency boundary to fill the spectrum portion. In the absence of a start frequency, states the highest frequency to start filling the spectrum (org-openroadm-common-optical-channel-types:frequency-THz)

	Parameter	*Manda-tory*	*Descriptions*
14	dedicated-customer	No	Applies only to spectrum-portion (both start and stop edge frequencies defined), when the spectrum-portion is dedicated to some customers. This list includes customer-codes identifying specific customers. No customer out of this list can share the bandwidth of this spectrum portion. Leaflist (String)
15	non-authorized-customer	No	Applies only to spectrum-portion. Includes customer-codes identifying specific customers that are not allowed to share the bandwidth of this spectrum portion. Leaflist (String)
16	dedicated-signal-bandwidth-multiple	No	Used for spectrum partitioning to reduce fragmentation (RSMA policy set to fragmentation-limiting-partitioning). Defines the spectral width of the service to be provisioned in the spectrum portion. As an example a spectrum portion can be dedicated to signals with a 50 GHz multiple spectral occupation, whereas another is dedicated to signal with a 75 GHz multiple spectral occupation. (Uint8)
margins			Defines all margins operator may want to specify
17	minimum-fiber-attenuation-bol-margin	No	Attenuation margin to be considered for path calculation at beginning of life (org-openroadm-common-link-types:ratio-dB)
18	threshold-observed-vs-design-attenuation	No	Defines the threshold used to raise an alarm when fiber initial attenuation is too close to the engineered-spanloss, meaning the value accounted for the design has been underestimated and a new design shall triggered. (org-openroadm-common-link-types:ratio-dB)
19	rw threshold-observed-vs-design-margin	No	Defines the threshold used to raise a warning when fiber measured attenuation comes too close to the engineered-spanloss, so that remaining margin is considered as too limited. (org-openroadm-common-link-types:ratio-dB)
20	minimum-osnr-margins		List of osnr margins to be considered (according to the rate and the modulation format)
21	margin-id	Yes	Identifier of the defined margin. (string)

	Parameter	*Manda-tory*	*Descriptions*
22	minimum-osnr-margin-value	No	Minimum value. (org-openroadm-common-link-types:ratio-dB)
23	line-rates	No	Line-rates to which the osnr-margin applies. Leaflist (uint64)
24	modulation-formats	No	Modulation-formats to which the osnr-margin applies. Leaflist(org-openroadm-common-optical-channel-types:modulation-format)
metrics-policy			Defines how the RNC shall interpret the routing-metrics
25	composite-metric-versus-selective	No	True corresponds to a weighted composite metric, False corresponds to a selective priority based metric. (Boolean)
regeneration-policy			Defines rules to place regenerators when a service-feasibility-check RPC is invoked, or when a service-create RPC is invoked if spare regenerators are already in place.
26	global-placement	No	How regenerators position is handled. (enumeration : regenerator-banks, distributed)
27	on-path-positioning	No	Rule used to position regenerators. (enumeration: maximize-rate, latest-convenient-hop)
28	path-symmetry	No	True corresponds to same location on A to Z and Z to A paths. False allows positioning regenerators in different nodes for A to Z and Z to A paths. (Boolean)
29	preferred-sites	No	CLLI's of the sites to be privileged when positioning regenerators in specific locations. Leaflist (String)
global-restriction			Defines specific restrictions that may apply.
30	site-restriction		Set of the restrictions applying to sites (CLLI).
31	site		List of sites to be excluded from paths when routing
32	site-id	Yes	Site identifier. (String)
33	restriction-type	No	(restriction-type: add-drop, pass-through, both)

	Parameter	*Manda-tory*	*Descriptions*
34	restriction-scopes	No	Defines the RPCs in the scope of the restriction. Default scope is set to all RPCs. Leaflist (restriction-scope)
35	node-restriction		Set of the restrictions applying to nodes.
36	node		List of equipment nodes to be excluded from paths when routing.
37	node-id	Yes	Node identifier. (String)
38	restriction-type	No	(restriction-type: add-drop, pass-through, both)
39	restriction-scopes	No	Defines the RPCs in the scope of the restriction. Default scope is set to all RPCs. Leaflist (restriction-scope)
40	regeneration-site-restriction		Set of the restrictions applying to sites, for what concerns regeneration.
41	forbidden-site		List of forbidden sites where regenerators shall not be placed.
42	site-id	Yes	Site identifier. (String)
43	regeneration-restriction-type	No	(regeneration-restriction-type: do-not-propose, do-not-use-existing, both)
44	restriction-scopes	No	Defines the RPCs in the scope of the restriction. Default scope is set to all RPCs. Leaflist (restriction-scope)
45	regeneration-node-restriction		Set of the restrictions applying to nodes, for what concerns regeneration.
46	forbidden-node		List of forbidden nodes where regenerators shall not be placed.
47	node-id	Yes	(String)
48	regeneration-restriction-type	No	(regeneration-restriction-type: do-not-propose, do-not-use-existing, both)

	Parameter	*Manda-tory*	*Descriptions*
49	restriction-scopes	No	Defines the RPCs in the scope of the restriction. Default scope is set to all RPCs. Leaflist (restriction-scope)
50	link-restriction		Set of the restrictions applying to links.
51	link		List of links to be restricted. Leaflist(string)
52	link-id	Yes	Link identifier
53	link-restriction-type	No	link-restriction-type: fully-restricted or pass-through-only-allowed
54	restriction-scopes	No	Defines the RPCs in the scope of the restriction. Default scope is set to all RPCs. Leaflist (restriction-scope)
55	supporting-service-restriction		Set of the restrictions applying to links.
56	supporting-service-list		List of supporting-services to be excluded from paths when routing. Leaflist(string)
57	supporting-service	Yes	Service identifier; index for supporting-service-list
58	supporting-service-restriction-type	No	Uses link-restriction-type: fully-restricted or pass-through-only-allowed
59	restriction-scopes	No	Defines the RPCs in the scope of the restriction. Default scope is set to all RPCs. Leaflist (restriction-scope)
default-behaviour			Parameters in this container are used to define default behavior in case optional parameters in RPC have not been defined
60	default-backup-path-number	No	0 means on the fly path calculation. Higher number corresponds to backup path pre-calculation, and states the number of paths to be calculated (uint16)
61	reversion	No	Concerns reversion for service that have a resiliency defined as restorable (restoration handled autonomously by the controller at the WDM/OTN layer). (Boolean)

	Parameter	*Manda-tory*	*Descriptions*
62	wait-to-restore	No	Time delay to revert to initial path after conditions for reversion are satisfied in ms. (uint64)
63	holdoff-time	No	Time delay to initiate a protection or restoration event in ms. (uint64)
sla-definition			Definition of Service Level Agreements parameters
64	sla-parameters		List of SLA profiles and associated expected behavior.
65	sla-id	Yes	Sla-identifier as defined by the operator. (string)
66	preemption	No	False: service shall never be preempted. True: preemption of the service is allowed. (Boolean)
67	restoration-priority?	No	The service shall be restored according to the defined priority (First:1, latest:255). Services with highest priority will be restored first. (uint8)
failure-case-list			Use to provide information on failure cases associated with backup-path pre-calculation: if PCE supports the calculation of multiple backup-paths, these might be identified through a failure case-id corresponding to a node, physical-link or logical-link failure
68	failure-case		List of failure cases. Each of them can be associated with one or several backup-paths in the context of backup-path pre-calculation.
69	failure-case-id	Yes	Failure case identifier. (uint32)
70	failure-type	No	(enumeration : node-failure, logical-link-failure, physical-link-failure)
71	nodes	No	List of nodes impacted by the failure as they appear in the openroadm-topology or otn-topology layer (not only single failures may be envisaged). (string)
72	logical-links	No	List of logical links impacted by the failure as they appear in the otn-topology layer (not only single failures may be envisaged). Leaflist (string)

	Parameter		Manda-tory	Descriptions
73		physical-links	No	List of physical links impacted by the failure as they appear in the openroadm-topology layer (not only single failures may be envisaged). Leaflist (string)

4 OPERATIONAL MODE CATALOG

Operational mode was introduced in R 5.0 for bookended Xponders in order to configure end terminals that may not be 100% compliant with OpenROADM optical specifications. In this use case, we consider having Xponders for which the specifications exceed OpenROADM optical specification requirements. Rather than using explicit configuration for parameters such as the FEC or the modulation format, bookended Xponders can be configured through the operational mode letting the responsibility of detailed configuration to the device.

In R10.0 we proposed to extend the concept of operational modes to provide to the Path Computation Element all physical parameters that are required for impairment aware path calculation. The operational mode Catalog model allows keeping in the controller data store all parameters associated with performances, avoiding hardcoding the validation of an optical path using specific constants for any potential use case. This further simplify the evolution of the PCE code and allows addressing the complexity of the specifications for Beyond 100G applications.

The operational mode Catalog is built from two main parts. The first part of the catalog is dedicated to the translation of the OpenROADM optical specifications into a data model. Noise Mask conversion into polynomial fit simplifies the translation of abacuses into data models. Operational modes and their associated physical parameters are provided for the Xponders (W specs), for the ROADMs Degrees (MW-MW specs) and SRGs (MW-WR specs), as well as for the amplifiers. To each specification is associated an openroadm-operational-mode-id.

The second part of the catalog is dedicated to the description of specific-operational-modes. It concerns only Xponders and pluggables. The model used (included parameters) is almost the same as for the openroadm-operational-modes. Specific operational modes are used to address not only the bookended transponders use case, but also the Alien Wavelength use case where Xponders or Pluggable may or may not comply with OpenROADM API specification. If they do not, these Xponders/pluggables may be configured by an external controller, such as an IP-SDNC, but the path computation is still required by the RNC to validate the feasibility of an "optical tunnel" from an SRG PP to another SRG PP, which guaranties that the performance level is adequate and comply with the end-terminals specifications. If they do comply with OpenROADM APIs, the Xponder might be configured by the RNC, which still need to be aware of the performances of end-terminals.

4.1 Operational Modes naming convention

Operational follow a naming convention that differs from Open ROADM operational mode to specific operational mode.

For OpenROADM operational mode the mode id is included in each of the spreadsheet of the optical specifications subject to a transcription in the operational-mode catalog (starting from R 6.0 of the optical specifcations).

For specific operational modes that are defined in the scope of bookended use cases, the naming convention is as follows:

ORBKD-int_type-rate-modeID

- ORBKD: identifies a Bookended operational mode
- int-type: interface type → as an example : "W" as used in the optical specification to define Xponder specifications
- rate: interface rate → as an example : "400G"

- modeID: mode identifier assigned by the MSA to ensure the mode-id uniqueness

4.2 Operational-mode-catalog structure

Table 4-1 Operational Mode Catalog Structure

	Parameter	*Manda-tory*	*Descriptions*
openroadm-operational-modes			
1	grid-parameters		List of general parameters describing spectrum use applicable to all openroadm-operational-modes
2	min-central-frequency	Yes	Lower spectrum boundary (org-openroadm-common-optical-channel-types:frequency-THz)
3	max-central-frequency	Yes	Higher spectrum boundary (org-openroadm-common-optical-channel-types:frequency-THz)
4	central-frequency-granularity	Yes	Spectrum management granularity (org-openroadm-common-optical-channel-types:frequency-GHz)
5	min-spacing	Yes	Minimum spacing between two adjacent channels (org-openroadm-common-optical-channel-types:frequency-GHz)
6	xponders-pluggables		Catalog section describing physical parameters for Xponders and pluggables compliant with OpenROADM optical specifications
7	xponder-pluggable-openroadm-operational-mode		List of openroadm-operational-modes corresponding to W specifications
8	openroadm-operational-mode-id	Yes	openroadm-mode identifier associated with a W specifications (string)
9	baud-rate	No	baud-rate expressed in Gbauds (decimal-64)
10	line-rate	Yes	baud-rate expressed in Gbps (decimal-64)

	Parameter	*Manda-tory*	*Descriptions*
11	modulation-format	Yes	modulation-format (org-openroadm-common-optical-channel-types:modulation-format)
12	channel-width	No	-20 dB channel width, required for target power calculation (org-openroadm-common-optical-channel-types:frequency-GHz)
13	min-TX-osnr	Yes	Minimum transmitter OSNR defined in 0.1nm bandwidth @ 193.6 THz, using the approach defined in ITU-T G.698.2 (org-openroadm-common-link-types:ratio-dB)
14	TX-OOB-osnr		List describing noise contribution associated with a specific transponder according to the multiplexing architecture of the SRG it is connected to.
15	WR-openroadm-operational-mode-id	Yes	Specification followed by the SRG the transponder is connected to
16	min-TX-OOB-osnr-single-channel-value	Yes	Minimum transmitter OSNR measured outside +/-150GHz BW, excluding SMSR, single channel (org-openroadm-common-link-types:ratio-dB)
17	min-TX-OOB-osnr-multi-channel-value	Yes	Minimum transmitter OSNR measured outside +/-150GHz BW, including SMSR, and 15 channels worst case contribution, corresponding to a specific multiplexing architecture defined by the WR-operational-mode-id (org-openroadm-common-link-types:ratio-dB)
18	output-power-range		List of output power ranges provided according to the multiplexing architecture the Xponder is connected to
19	WR-openroadm-operational-mode-id	Yes	Specification followed by the SRG the transponder is connected to
20	min-output-power	Yes	Minimum transmitter output power (org-openroadm-common-link-types:ratio-dB)
21	max-output-power	Yes	Maximum transmitter output power (org-openroadm-common-link-types:ratio-dB)
22	min-RX-osnr-tolerance	Yes	Minimum RX osnr required defined in 0.1 nm @ 193.6 Thz bandwidth (org-openroadm-common-link-types:ratio-dB)

	Parameter	*Manda-tory*	*Descriptions*
23	min-input-power-at-RX-osnr	Yes	Minimum receiver input power for which the RX-osnr-tolerance is provided (org-openroadm-common-link-types:ratio-dB)
24	max-input-power	Yes	Maximum receiver input power (org-openroadm-common-link-types:ratio-dB)
25	fec-type	No	Fec-type (identityref, base org-openroadm-common-types:fec-identity)
26	min-roll-off	No	minimum roll-off factor expressed in dB/decade (dec64)
27	max-roll-off	No	maximum roll-off factor expressed in dB/decade (dec64)
28	penalties		List of penalties defined for specific impairments, indexed by parameter-and-unit and up-to-boundary
29	parameters-and-unit	Yes	The parameter for which the penalty is defined and the unit used (typedef impairment-type)
30	up-to-boundary	Yes	Defines the upper (for positive values) and lower (for negative values) limit for which the penalty value is valid (uint32)
31	penalty-value	No	Defined penalty value (org-openroadm-common-link-types:ratio-dB)
32	roadms		
33	Express		
34	openroadm-operational-mode		List of operational modes defined for Express path
35	openroadm-operational-mode-id	Yes	The operational mode identifier corresponding to a specific OpenROADM MW-MW specification (string)
36	per-channel-Pin-min	No	Minimum per channel input power (org-openroadm-common-link-types:ratio-dB)
37	per-channel-Pin-max	No	Maximum per channel input power (org-openroadm-common-link-types:ratio-dB)

	Parameter	*Manda-tory*	*Descriptions*
38	max-introduced-pdl	No	Maximum Polarization dependent Loss introduced by the module (org-openroadm-common-link-types:ratio-dB)
39	max-introduced-dgd	No	Maximum Differential Group Delay introduced by the module (decimal-64)
40	max-introduced-cd	No	Maximum Chromatic Dispersion introduced by the module (decimal-64)
41	osnr-polynomial-fit		List of parameters describing the Noise Mask polynomial fit OSNR (dB/0.1nm) = A*Pin^3+B*Pin^2+C*Pin +D; Pin (dB)
42	A	No	Multiplier for Pin^3
43	B	No	Multiplier for Pin^2
44	C	No	Multiplier for Pin
45	D	No	Constant
46	mask-power-vs-pin		List describing how target-output-power shall be calculated according to the exit span loss. The curve has different profile for each loss range defined by upper and lower boundary. for each range we define Pout (dBm) = C*span-loss (dB) +D
47	lower-boundary	Yes	Span-loss lower boundary (uint32)
48	upper-boundary	Yes	Span-loss upper boundary (uint32)
49	C	No	Multiplier for Span-loss
50	D	No	Constant
51	fiber-type	No	The mask is given for a specific fiber-type (enumeration)
52	Add		
53	openroadm-operational-mode		List of operational modes defined for Add path

	Parameter	*Manda-tory*	*Descriptions*
54	openroadm-operational-mode-id	Yes	The operational mode identifier corresponding to a specific OpenROADM MW-WR specification (string)
55	per-channel-Pin-min	No	Minimum per channel input power (org-openroadm-common-link-types:ratio-dB)
56	per-channel-Pin-max	No	Maximum per channel input power (org-openroadm-common-link-types:ratio-dB)
57	max-introduced-pdl	No	Maximum Polarization dependent Loss introduced by the module (org-openroadm-common-link-types:ratio-dB)
58	max-introduced-dgd	No	Maximum Differential Group Delay introduced by the module (decimal-64)
59	max-introduced-cd	No	Maximum Chromatic Dispersion introduced by the module (decimal-64)
60	incremental-osnr	No	Incremental OSNR considering noiseless input at 0dBm (org-openroadm-common-link-types:ratio-dB)
61	mask-power-vs-pin		List describing how power shall be calculated according to the exit span loss. The curve has different profile for each loss range defined by upper and lower boundary. for each range we define Pout (dBm) = C*span-loss (dB) +D
62	lower-boundary	Yes	Span-loss lower boundary (uint32)
63	upper-boundary	Yes	Span-loss upper boundary (uint32)
64	C	No	Multiplier for Span-loss
65	D	No	Constant
66	fiber-type	No	The mask is given for a specific fiber-type (enumeration)
67	Drop		
68	openroadm-operational-mode		List of operational modes defined for Drop path

	Parameter	*Manda-tory*	*Descriptions*
69	openroadm-operational-mode-id	Yes	The operational mode identifier corresponding to a specific OpenROADM specification (string)
70	per-channel-Pin-min	No	Minimum per channel input power (org-openroadm-common-link-types:ratio-dB)
71	per-channel-Pin-max	No	Maximum per channel input power (org-openroadm-common-link-types:ratio-dB)
72	max-introduced-pdl	No	Maximum Polarization dependent Loss introduced by the module (org-openroadm-common-link-types:ratio-dB)
73	max-introduced-dgd	No	Maximum Differential Group Delay introduced by the module (decimal-64)
74	max-introduced-cd	No	Maximum Chromatic Dispersion introduced by the module (decimal-64)
75	osnr-polynomial-fit	No	List of parameters describing the Noise Mask polynomial fit OSNR (dB/0.1nm) = A*Pin^3+B*Pin^2+C*Pin +D; Pin (dB)
76	A	No	Multiplier for Pin^3
77	B	No	Multiplier for Pin^2
78	C	No	Multiplier for Pin
79	D	No	Constant
80	per-channel-Pout-min	No	Minimum per channel output power (org-openroadm-common-link-types:ratio-dB)
81	per-channel-Pout-max	No	Maximum per channel output power (org-openroadm-common-link-types:ratio-dB)
82	amplifiers		
83	Amplifier		
84	openroadm-operational-mode		List of operational modes defined for amplifiers

	Parameter	*Manda-tory*	*Descriptions*
85	openroadm-operational-mode-id	Yes	The operational mode identifier corresponding to a specific OpenROADM specification (string)
86	per-channel-Pin-min	No	Minimum per channel input power (org-openroadm-common-link-types:ratio-dB)
87	per-channel-Pin-max	No	Maximum per channel input power (org-openroadm-common-link-types:ratio-dB)
88	max-introduced-pdl	No	Maximum Polarization dependent Loss introduced by the module (org-openroadm-common-link-types:ratio-dB)
89	max-introduced-dgd	No	Maximum Differential Group Delay introduced by the module (decimal-64)
90	max-introduced-cd	No	Maximum Chromatic Dispersion introduced by the module (decimal-64)
91	osnr-polynomial-fit		List of parameters describing the Noise Mask polynomial fit OSNR (dB/0.1nm) = A*Pin^3+B*Pin^2+C*Pin +D; Pin (dB)
92	A	No	Multiplier for Pin^3
93	B	No	Multiplier for Pin^2
94	C	No	Multiplier for Pin
95	D	No	Constant
96	mask-power-vs-pin		For power range provides C and D parameter : Pout[50GHz BW] (dBm) = C*span-loss (dBm)+ D". List, indexed by lower-boundary and upper-boundary
97	lower-boundary		defines the lower Power boundary for which C & D parameters apply
98	upper-boundary		defines the upper Power boundary for which C & D parameters apply
99	C		C*span loss
100	D		Constant

	Parameter	*Manda-tory*	*Descriptions*
101	min-gain	No	Minimum amplifier gain (org-openroadm-common-link-types:ratio-dB)
102	max-gain	No	Maximum amplifier gain respecting guaranteed tilt (org-openroadm-common-link-types:ratio-dB)
103	max-extended-gain	No	Maximum amplifier extended gain, for which tilt is no more guaranteed (org-openroadm-common-link-types:ratio-dB)
104	mask-gain-ripple-vs-tilt	No	Provides the mask used to define ripple's limits according to the target tilt
105	lower-boundary	Yes	Defines the lower boundary of the target tilt on the abacus for the corresponding part of the curve (int32)
106	upper-boundary	Yes	Defines the lower boundary of the target tilt on the abacus for the corresponding part of the curve (int32)
107	C	No	Defines the C parameter in: max-gain-ripple= C*Target-tilt +D (decimal64)
108	D	No	Defines the D parameter in: max-gain-ripple= C*Target-tilt +D (decimal64)
specific-operational-modes			
109	specific-operational-mode		List of specific-operational-modes and their associated parameters
110	operational-mode-id		Specific operational mode id (string)
111	originator	No	System vendor which originated associated specification (string)
112	sponsor	No	Service provider which reviewed and sponsored associated specification (string)
113	min-central-frequency	Yes	Lower spectrum boundary (org-openroadm-common-optical-channel-types:frequency-THz)
114	max-central-frequency	Yes	Higher spectrum boundary (org-openroadm-common-optical-channel-types:frequency-THz)

	Parameter	Manda-tory	Descriptions
115	central-frequency-granularity	Yes	Spectrum management granularity (org-openroadm-common-optical-channel-types:frequency-GHz)
116	min-spacing	Yes	Minimum spacing between two adjacent channels (org-openroadm-common-optical-channel-types:frequency-GHz)
117	baud-rate	No	baud-rate expressed in Gbauds (decimal-64)
118	line-rate	Yes	baud-rate expressed in Gbps (decimal-64)
119	modulation-format	Yes	modulation-format (org-openroadm-common-optical-channel-types:modulation-format)
120	channel-width	No	-20 dB channel width, required for target power calculation (org-openroadm-common-optical-channel-types:frequency-GHz)
121	min-TX-osnr	Yes	Minimum transmitter OSNR defined in 0.1nm bandwidth @ 193.6 THz, using the approach defined in ITU-T G.698.2 (org-openroadm-common-link-types:ratio-dB)
122	TX-OOB-osnr		List describing noise contribution associated with a specific transponder according to the multiplexing architecture of the SRG it is connected to.
123	WR-openroadm-operational-mode-id	Yes	Specification followed by the SRG the transponder is connected to
124	min-TX-OOB-osnr-single-channel-value	Yes	Minimum transmitter OSNR measured outside +/-150GHz BW, excluding SMSR, single channel (org-openroadm-common-link-types:ratio-dB)
125	min-TX-OOB-osnr-multi-channel-value	Yes	Minimum transmitter OSNR measured outside +/-150GHz BW, including SMSR, and 15 channels worst case contribution, corresponding to a specific multiplexing architecture defined by the WR-operational-mode-id (org-openroadm-common-link-types:ratio-dB)
126	output-power-range		List of output power ranges provided according to the multiplexing architecture the Xponder is connected to

	Parameter	*Manda-tory*	*Descriptions*
127	WR-openroadm-operational-mode-id	Yes	Specification followed by the SRG the transponder is connected to
128	min-output-power	Yes	Minimum transmitter output power (org-openroadm-common-link-types:ratio-dB)
129	max-output-power	[22]	Maximum transmitter output power (org-openroadm-common-link-types:ratio-dB)
130	min-RX-osnr-tolerance	Yes	Minimum RX osnr required defined in 0.1 nm @ 193.6 Thz bandwidth (org-openroadm-common-link-types:ratio-dB)
131	min-input-power-at-RX-osnr	Yes	Minimum receiver input power for which the RX-osnr-tolerance is provided (org-openroadm-common-link-types:ratio-dB)
132	max-input-power	Yes	Maximum receiver input power (org-openroadm-common-link-types:ratio-dB)
133	fec-type	No	Fec-type (identityref, base org-openroadm-common-types:fec-identity)
134	min-roll-off	No	minimum roll-off factor expressed in dB/decade (dec64)
135	max-roll-off	No	maximum roll-off factor expressed in dB/decade (dec64)
136	penalties		List of penalties defined for specific impairments
137	parameters-and-unit	Yes	The parameter for which the penalty is defined and the unit used (typedef impairment-type)
138	up-to-boundary	Yes	Defines the upper (for positive values) and lower (for negative values) limit for which the penalty value is valid (uint32)
139	penalty-value	No	Defined penalty value (org-openroadm-common-link-types:ratio-dB)
140	Configurable-output-power	Yes	Boolean, defines whether the output power can be set or not

5 Remote Procedure Calls (RPCs)

The ROADM Service Model specifies Remote Procedure Calls (RPCs). The service providers' SDN Controllers can make requests to the ROADM Network Controller or Open ROADM Controller using RPCs to create or delete services, perform changes in the ROADM network.

There are 15 RPCs defined in the Open ROADM Service Model version 4.1.0 by typedef rpc-actions:

RPC Name	Enum Value
Service create	1
Service feasibility check	2
Service delete	3
Equipment notification	4
Temp service create	5
Temp service delete	6
Service roll	7
Service reconfigure	8
Service restoration	9
Service reversion	10
Service reroute	11
Service reroute confirm	12
Network reoptimization	13
Service feasibility check bulk	14
BER test	15

Release 7.0 introduces 1 and Release 10.0, 3 additional RPCs.

RPC Name	Enum Value
Controller parameters setting	16
Optical tunnel create	17
Optical tunnel request cancel	18
Add-openroadm-operational-modes-to-catalog	19
Add-specific-operational-modes-to-catalog	20

Release 11.1 introduces 4 additional RPCs.

RPC Name	Enum Value
End terminal performance info request	21

End terminal activation request	22
End terminal deactivation request	23
End terminal power control	24

5.1 Service Create RPC

This RPC is for the service providers' SDN Controller to request the RNC or Open ROADM Controller to create a new service either immediately or in the future. If this request passed the initial validation and was accepted for processing, a service RPC result notification shall be sent once the request completes processing. Table 5-1Table 5-1 Service Create RPC and Input Prameters lists the input parameters included in the service create RPC and their descriptions. The synchronous response to the service create RPC is listed in Table 5-2

Table 5-1 Service Create RPC and Input Prameters

	Input Parameter		***Manda-tory***	***Descriptions***
1	Services Name		Yes	Identifier for the service to be created in the ROADM network, e.g., CLFI, CLCI, etc. This is reported against the service but may not get reflected in the service in the network. (string)
2	Common ID		No	Service order #, or identifier to be used by the ROADM controller to identify routing constraints received from planning applications. (string). Also used to correlate to an existing temp service when converting the temp service into a normal service.
3	Order ID		No	Service order identifier. May reflect the end customer service order (string).
4	Order note		No	Additional details associated with the service order identified by order-id (string).
5	SDNC Request Header	*Request ID*	No	From original system requesting for the service. Uniquely generated by calling system. (string)
6		*RPC Action*	No	Service create, Enum=1
7		*Notification url*	No	URL for asynchronous response (string)
8		*Request System ID*	No	Identifier of application initiates the request (string)

	Input Parameter		***Manda-tory***	***Descriptions***
9	Routing Metric	*Wdm hop count*	No	The number of hops in the wdm layer will be used as a metric.
10		*otn-hop-count*	No	Number of hops in the otn layer will be used as a metric.
11		*wdm-load*	No	The load of the wdm layer will be used as a metric, to avoid using heavy loaded links.
12		*otn-load*	No	The load of the otn layer will be used as a metric, to avoid using heavy loaded links.
13		*latency*	No	Total path latency is used as a metric.
14		*distance*	No	Total path distance is the metric.
15		*wdm-TE-metric*	No	Used when routing shall be performed according to specific pre-defined TE metric. Total path metric can be calculated from OMS TE-metric attribute defined in org-openroadm-link module.
16		*otn-TE-metric*	No	Used when routing shall be performed according to specific pre-defined metric associated with OTN (OTU/ODU level).
17		*adaptation-number*	No	Adaptation between layers is the metric. Total path metric can be calculated from the total number of transitions between layers.
18	Service Resiliency	*Resiliency*	No	Identity ref with the following types: unprotected, unprotected-diversely-routed, protected, restorable, and external-trigger-restorable
19		*Revertive*	No	Specifies whether the service shall revert to its initial working path after protection switching and fault conditions have cleared.
20		*Wait to restore*	No	Time delay for switching to backup path
21		*Holdoff time*	No	Time delay for reverting to initial working path
22		*Pre-calculated backup path number*	No	Provides the target number of backup paths conforming with specific engineering rules

	Input Parameter				***Manda-tory***	***Descriptions***
23			*Coupled Service*	*Service index*	Yes	Service number of the service that is disjointly routed from the failed service. Service index is the key to a list of coupled services that may be used for restoration.
24				*Service name*	No	Name of the service that is disjointly routed from the failed service
25				*Common ID*	No	Common ID of the service that is disjointly routed from the failed service
26				*Version number*	No	Service version number of the service that is disjointly routed from the failed service
27	Connection Type				Yes	4 types: Service, Enum=1; Infrastructure, Enum=2; ROADM line, Enum=3; optical-tunnel, Enum=4;
28	Resource status				No	(resource-status-type) Enum to indicate which network resources the controller should consider when computing a path. Options are ***deployed*** (default; use network resources with lifecycle-state=deployed and ignore operational-state), ***in-service*** (route around failed services; lifecycle-state=deployed and operational-state=inService), and ***planned*** (lifecycle-state=planned or any deployed/deploying state including deploy-failed, and operational-state is ignored; if ***planned*** if selected, due date must be specified in the service request)
29	Service A-end	*See input parameters for Service A-end in* Table 2-1. Service List*, (lines 36-106)*			Yes	Refer to descriptions in Table 2-1, lines 36 - 106
30	Service Z-end	*See input parameters for Service A-end in* Table 2-1. Service List *(lines 36 - 106)*			Yes	Refer to descriptions in Table 2-1, lines 36 - 106
31	Hard Cons-traints	*See input parameters for Hard Constraints in* Table 2-1. Service List *(lines 113 – 145)*			No	Refer to descriptions in Table 2-1, lines 113-145
32	Soft Cons-traints	*See input parameters for Hard Constraints in* Table 2-1. Service List *(lines 113 – 145)*			No	Refer to descriptions in Table 2-1, lines 113-145

	Input Parameter	*Manda-tory*	*Descriptions*
33	Due date	No	Date and time service to be turn up. If time is not specified for a given date, default to midnight. Service turned up immediately if no *due date* is specified. Type: yang:*date*-and-time
34	End Date	No	Date and time service to be removed. Type: yang:*date*-and-time
35	Event Horizon Start	No	Start time to ensure that the service is routable and viable. Required resources shall be considered reserved from this time. If not provided, defaults to due-date. Type: yang:*date*-and-time
36	Event Horizon End	No	End time to ensure that the service is routable and viable. Required resources shall be considered reserved until this time. If not provided, defaults to end-date. Type: yang:*date*-and-time
37	NC code	No	Network Channel code applied to wavelength service only. This is reported against the service, but may not get reflected in the service in the network (string).
38	NCI code	No	Network Channel Interface code applied to wavelength service only. This is reported against the service, but may not get reflected in the service in the network (string).
39	Secondary NCI code	No	Secondary NCI code applied to wavelength service only. This is reported against the service, but may not get reflected in the service in the network (string).
40	Customer	No	To be included in ticket information. This is reported against the service, but may not get reflected in the service in the network (string).
41	Customer contact	No	Customer contact information to be included in ticket information. This is reported against the service, but may not get reflected in the service in the network (string).
42	Operator contact	No	Operator contact information to be included in ticket information. This is reported against the service, but may not get reflected in the service in the network (string).
43	Service layer	No	Layer associated with service. 2 types: WDM, Enum=1; OTN, Enum=2

	Input Parameter	*Manda-tory*	*Descriptions*
44	CLLI network ref	No	Network-id of the clli-network layer from the network model (string)
45	Openroadm network ref	No	Network-id of the openroadm-network layer from the network model (string)
46	Openroadm topology ref	No	Network-id of the openroadm-topology layer from the network model (string)
47	Sla id	No	SLA defined for the service (string)
48	Bandwidth calendaring	No	When true, triggers the following structure allowing the description of the bandwidth calendaring options. (boolean)
49	Bandwidth calendaring parameters		Container gathering attributes describing the bandwidth calendaring options for the service.
50	BW calendaring coupled services	No	List of services that may be associated with the considered service. These services can be in service-list, temp-service-list, and versioned-service-list. The service and its coupled-service(s) may be defined on complementary time periods.
51	Service index	No	Coupled service identifier. (String)
52	Service name	No	Coupled service name for standard-service. (String)
53	Common-id	No	Coupled service identifier for temporary service. (String)
54	Version number	No	Coupled service version number for versioned service. (uint64)
55	Recurrence pattern	No	Defines a service that is active in day-of-the-week with start-time and end-time. (String)
56	recurrence id	No	Identifier of the recurrence scheme. (uint32)
57	day of the week	No	Day of the week the service is active. (enumeration)
58	Start time	No	Start time for service activation. Applies to any days of the recurrence scheme. (String)
59	End time	No	Time at which the service is deactivated. Applies to any days of the recurrence scheme. (String)

The Open ROADM Service Model defines the synchronous response to the service create RPC. Table 5-2 lists the output parameters.

Table 5-2 Synchronous Response to Service Create RPC

Output	***Field Name***						***Manda-tory***	***Note***
Configuration Response Common[8]	*Request ID*						Yes	The request ID from the request message for which this is the response (string)
	Response Code						Yes	One of the codes defined for success or error (string)
	Response Message						No	Message included for error code (string)
	Ack-final-indicator						Yes	Indicates if this is the last response that the client should expect (string).
Response Parameters[9]	*Hard Constraints*	*Customer Code*					No	For selecting tagged equipment on which to route a service. If more than one customer code is provided, they will be treated as an ordered list. (string)
		Operational mode					No	List of operational modes supported by the service. Leaflist (string)
		Diversity						List of services from which this service is diverse, identified by service-identifier.
			Service identifier list				Yes	Mandatory if service diversity is supported.
				Service identifier			Yes	Unique service-identifier, may be service-name or common-id
				Service applicability				Identifies the scope of service diversity
					Site		No	Whether service is site diverse (Boolean)

[8] Is a container, must be part of the table.
[9] E.g., violated soft constraints, etc.

					Node			No		Whether service is node diverse (Boolean)
					SRLG			No		Whether service is SRLG diverse (Boolean)
					link			No		Whether service is link diverse (Boolean)
					equipment					Whether service is equipment diverse (Boolean)
						Roadm srg		No		If equipment diverse, does service use a different ROADM SRG
						Xponder srg		No		If equipment diverse, does service use a different xponder SRG
			Diversity type							Enum to indicate whether bulk services are serially or synchronously routed for diversity. The value is assumed to be the same for all services in the bulk request (either serial or synchronous).
		Exclude								
			Fiber bundle					No		List of fiber bundles excluded.
			SRLG id					No		List of SRLGs excluded
			site					No		List of sites excluded, site is identified by CLLI.
			Node id					No		List of nodes excluded, as identified by node-id. Globally unique identifier for a device length "7..63" pattern "([a-zA-Z][a-zA-Z0-9-]{5,18}[a-zA-Z0-9])"[10] A Node ID can contain letters, numbers, and hyphens. The first character must be a letter. The last character must be a letter or number. Reported against the service but may not get reflected in the service in the network.

[10] The pattern for Node ID is incorrect in the Open ROADM YANG model as it doesn't allow the length to be extended past 20 characters. This will be fixed in a future release of the YANG models.

			Link identifier		List of link-identifier excluded, indexed by link-network-id and link-id from the network model.
			Link-network-id		Link-network-id is a string equal to network-id from the network model.
			Link-id		Link-id from the network model (string).
			Supporting service name		Supporting service(s) excluded from this route (string), list. Supporting service is the service name that another service runs over top.
		Include			
			Is explicit routing		Boolean; indicates whether list of included resources provides a full explicit service path.
			Is include list ordered		Boolean; indicates whether list of included resources is provided in order.
			Fiber bundle		List of fiber bundles included, ordered if is-include-list-ordered is true.
			SRLG id		List of SRLGs included, ordered if is-include-list-ordered is true.
			site		List of sites included, site is identified by CLLI; ordered if is-include-list-ordered is true.
			Node id		List of node-id included; ordered if is-include-list-ordered is true. Globally unique identifier for a device length "7..63" pattern "([a-zA-Z][a-zA-Z0-9-]{5,18}[a-zA-Z0-9])"[11] A Node ID can contain letters, numbers, and hyphens. The first character must be a letter. The last character must be a letter or

[11] The pattern for Node ID is incorrect in the Open ROADM YANG model as it doesn't allow the length to be extended past 20 characters. This will be fixed in a future release of the YANG models.

								number. Reported against the service but may not get reflected in the service in the network.
			Link identifier					List of link-identifier included, indexed by link-network-id and link-id from the network model. Ordered if is-include-list-ordered is true.
				Link network id				Link-network-id is a string equal to network-id from the network model.
				Link id				Link-id from the network model (string).
		Latency						
			Max latency					Maximum latency allowed on service (uint32), units in “ms”.
		Hop count						
			Max WDM hop count					Maximum number of hops allowed at the WDM layer
			Max OTN hop count					Maximum number of hops allowed at the OTN layer
		TE Metric						
			Max WDM TE metric					Maximum cost allowed based on cost of WDM layer links
			Max OTN TE metric					Maximum cost allowed based on cost of OTN layer links
		Distance						
			Max distance					Maximum distance allowed based on length of physical spans
		Co-routing						
			Service identifier list					List of existing services that co-routed with this service, indexed by service-identifier.
				Service identifier				Unique service-identifier, may be service-name or common-id
				Service applicability				Identifies the scope of service co-routing
					Site			Whether service is co-routed based on site (Boolean)

					Node			Whether service is co-routed based on node (Boolean)
					SRLG			Whether service is co-routed based on SRLGs (Boolean)
					equipment			Whether service is co-routed based on equipment (Boolean)
						ROADM SRG		If equipment co-routed, does service use the same ROADM SRG
						Xponder SRG		If equipment co-routed, does service use the same xponder SRG
	Soft Cons-traints	*Repeat parameters in the Hard Constraints above. Indicate soft constraints that have been satisfied by the service create request.*					No	See descriptions for hard constraints.

5.2 Service Feasibility Check RPC

The service feasibility check RPC is a call to check whether a service can be provisioned in the existing network, i.e., requesting the RNC or Open ROADM Controller to check connectivity, equipment availability, and reachability. It is expected that the response from the RNC will confirm existing equipment is available for a new service or propose additional equipment to be ordered for the new service. Options are made available to choose from one of the following for routing:

- Using only deployed and planned equipment
- Using existing equipment first, then proposing new equipment as needed
- Using proposed equipment

No resources will be reserved, provisioned or planned because of this RPC. Table 5-3 lists the input parameters and their descriptions in the service feasibility check RPC. Note that service name is not present in this RPC. If this request passed the initial validation and was accepted for processing, a service RPC result notification shall be sent once the request completes processing.

Table 5-3 Service Feasibility Check RPC and Input Parameters

	Input Parameter		***Manda-tory***	***Descriptions***
1	Common ID		No	Service order #, or identifier to be used by the ROADM controller to identify routing constraints received from planning applications. (string).
2	SDNC Request Header	*Request ID*	No	From original system requesting for the service. Uniquely generated by calling system. (string)
3		*RPC Action*	No	Service feasibility check, Enum=2

	Input Parameter		***Manda-tory***	***Descriptions***
4		*Notification url*	No	URL for asynchronous response (string)
5		*Request System ID*	No	Identifier of application initiates the request (string)
6	Connection Type		No	4 types: Service, Enum=1; Infrastructure, Enum=2; ROADM line, Enum=3; optical-tunnel, Enum=4;
7	Resource status		No	(resource-status-type) Enum to indicate which network resources the controller should consider when computing a path. Options are ***deployed*** (default; use network resources with lifecycle-state=deployed and ignore operational-state), ***in-service*** (route around failed services; lifecycle-state=deployed and operational-state=inService), and ***planned*** (lifecycle-state=planned or any deployed/deploying state including deploy-failed, and operational-state is ignored; if ***planned*** if selected, due date must be specified in the service request)
8	Routing Metric	*Wdm hop count*	No	The number of hops in the wdm layer will be used as a metric.
9		*otn-hop-count*	No	Number of hops in the otn layer will be used as a metric.
10		*wdm-load*	No	The load of the wdm layer will be used as a metric, to avoid using heavy loaded links.
11		*otn-load*	No	The load of the otn layer will be used as a metric, to avoid using heavy loaded links.
12		*latency*	No	Total path latency is used as a metric.
13		*distance*	No	Total path distance is the metric.
14		*wdm-TE-metric*	No	Used when routing shall be performed according to specific pre-defined TE metric. Total path metric can be calculated from OMS TE-metric attribute defined in org-openroadm-link module.

	Input Parameter			***Manda-tory***	***Descriptions***
15		*otn-TE-metric*		No	Used when routing shall be performed according to specific pre-defined metric associated with OTN (OTU/ODU level).
16		*adaptation-number*		No	Adaptation between layers is the metric. Total path metric can be calculated from the total number of transitions between layers.
17	Service Resiliency	*Resiliency*		No	Identity ref with the following types: unprotected, unprotected-diversely-routed, protected, restorable, and external-trigger-restorable.
18		*Revertive*		No	Specifies whether the service shall revert to its initial working path after protection switching and fault conditions have cleared.
19		*Wait to restore*		No	Time delay for switching to backup path
20		*Holdoff time*		No	Time delay for reverting to initial working path
21		*Pre-calculated backup path number*		No	Provides the target number of backup paths conforming with specific engineering rules
22		*Coupled Service*	*Service index*	Yes	Service number of the service that is disjointly routed from the failed service. Service index is the key to a list of coupled services that may be used for restoration.
23			*Service name*	No	Name of the service that is disjointly routed from the failed service
24			*Common ID*	No	Common ID of the service that is disjointly routed from the failed service
25			*Version number*	No	Service version number of the service that is disjointly routed from the failed service
26	*Propose equipment*			No	Whether or not this request can propose new equipment that could be used to fulfill this request. If never, the request will just use existing deployed and planned equipment. If ifNeeded, routes using existing equipment will be preferred. If always, a route with proposed equipment shall be returned, if possible.

	Input Parameter				***Manda-tory***	***Descriptions***
						3 types: “Never”, Enum=1; “ifNeeded”, Enum=2, “Always”, Enum=3. Default is “ifNeeded”.
27	Service A-end	*See input parameters for Service A-end in* Table 2-1. Service List*, (lines 36-106)*			Yes	See description in Table 2-1, lines 36-106
28		*Requesting interface properties*			Yes	Triggered when connection-type = optical-tunnel. Provides external pluggable/Xponder characteristics
29			*supported-operational-modes*		Yes	List of supported operational modes
30				*preference*	Yes	Preference/priority associated with an operational mode. 1 is highest priority. (int16)
31				*operational-mode-id*	No	Operational mode identifier. (string)
32			*min-frequency*		No	Minimum frequency to be set. (org-openroadm-common-optical-channel-types:frequency-THz)
33			*max-frequency*		No	Maximum frequency to be set. (org-openroadm-common-optical-channel-types:frequency-THz)
34			*min-granularity*		No	Minimum NMC width. (org-openroadm-common-optical-channel-types:frequency-GHz)
35	Service Z-end	Repeat parameters from line 27 to 34 for Service Z-end				
36	Hard Cons-traints	*See input parameters for Hard Constraints in* Table 2-1. Service List *(lines 113 – 145)*			No	Refer to descriptions in Table 2-1, lines 113-145 Hard constraints must be met by the service feasibility check.
37	Soft Cons-traints	*See input parameters for Hard Constraints in* Table 2-1. Service List *(lines 113 – 145)*			No	Refer to descriptions in Table 2-1, lines 113-145 Soft constraints are preferred but if not met, the feasibility check will not fail.
38	Due date				No	Date and time service to be turned up. If time is not specified for a given date, default to midnight. Service will be turned up immediately if no *due date* is specified. Type: yang:*date*-and-time
39	End Date				No	Date and time service to be removed. Type: yang:*date*-and-time
40	Event Horizon Start				No	Start time to ensure that the service is routable and viable. Required resources shall be

	Input Parameter	*Manda-tory*	*Descriptions*
			considered reserved from this time. If not provided, defaults to due-date. Type: yang:*date*-and-time
41	Event Horizon End	No	End time to ensure that the service is routable and viable. Required resources shall be considered reserved until this time. If not provided, defaults to end-date. Type: yang:*date*-and-time
42	NC code	No	Network Channel code applied to wavelength service only. This is reported against the service but may not get reflected in the service in the network (string).
43	NCI code	No	Network Channel Interface code applied to wavelength service only. This is reported against the service but may not get reflected in the service in the network (string).
44	Secondary NCI code	No	Secondary NCI code applied to wavelength service only. This is reported against the service but may not get reflected in the service in the network (string).
45	Customer	No	To be included in ticket information. This is reported against the service but may not get reflected in the service in the network (string).
46	Customer contact	No	Customer contact information to be included in ticket information. This is reported against the service but may not get reflected in the service in the network (string).
47	Operator contact	No	Operator contact information to be included in ticket information. This is reported against the service but may not get reflected in the service in the network (string).
48	Service layer	No	Layer of the service. 2 types: WDM, Enum=1; OTN, Enum=2
49	CLLI network reference	No	Network-id of the clli-network layer from the network model (string)
50	Open Roadm network reference	No	Network-id of the openroadm-network layer from the network model (string)
51	Open Roadm topology reference	No	Network-id of the openroadm-topology layer from the network model (string)

	Input Parameter			*Manda-tory*	*Descriptions*
52	SLA id			No	SLA defined for the service (string)
53	Bandwidth calendaring			No	When true, triggers the following structure allowing the description of the bandwidth calendaring options. (boolean)
54	Bandwidth calendaring parameters				Use when bandwidth calendaring is true. Container gathering attributes describing the bandwidth calendaring options for the service.
55		BW calendaring coupled services		No	List of services that may be associated with the considered service. These services can be in service-list, temp-service-list, and versioned-service-list. The service and its coupled-service(s) may be defined on complementary time periods.
56			Service index	No	Coupled service identifier. (String)
57			Service name	No	Coupled service name for standard-service. (String)
58			Common ID	No	Coupled service identifier for temporary service. (String)
59			Version number	No	Coupled service version number for versioned service. (uint64)
60		Recurrence pattern		No	Defines a service that is active in day-of-the-week with start-time and end-time. (String)
61			Recurrent ID	No	Identifier of the recurrence scheme. (uint32)
62			Day of the week	No	Day of the week the service is active. (enumeration)
63			Start time	No	Start time for service activation. Applies to any days of the recurrence scheme. (String)
64			End time	No	Time at which the service is deactivated. Applies to any days of the recurrence scheme. (String)
65		Maximum regeneration options		No	Maximum number of regeneration elements that may be returned. The RNC may return fewer regen options than the max. Default value is 1.
66		Existing service attributes			Used when the feasibility check is performed on an existing service to describe attributes related to the existing service.

	Input Parameter			*Manda-tory*	*Descriptions*
67			Is existing	No	Boolean; if true, the feasibility check is for an existing service, with or without new constraints.
68			Existing service name	No	When is-existing = true, provides the name of the existing service.
69			Reuse existing resources	No	If true, the feasibility check should use the existing service resources/equipment wherever possible.
70			Reusable existing resources	No	When reuse-existing-resources is true, this provides a list of existing resource types that may be reused. Define by typedef existing-resource-reuse-type (values are regenerator, wavelength, spectrum-portion, xponder, all)

The Open ROADM Service Model defines the synchronous response to the service feasibility check RPC. Table 5-4Table 5-4 Synchronous Response to Service Feasibility Check RPC lists the output parameters.

Table 5-4 Synchronous Response to Service Feasibility Check RPC

	Output Field Name			*Manda-tory*	*Note*
1	*Common ID*			Yes	Service order #, or identifier to be used by the ROADM controller to identify routing constraints received from planning applications. (string).
2	*Configuration Response Common*	*Request ID*		Yes	The request ID from the request message for which this is the response (string)
3		*Response Code*		Yes	One of the codes defined for success or error (string)
4		*Response Message*		No	Message included for error code (string)
5		*Ack-final-indicator*		Yes	Indicates if this is the last response that the client should expect (string).
6	*Response parameters*				E.g., violated soft constraints, etc.
7		*Hard constraints*	*See input parameters for Hard Constraints in*	No	Refer to descriptions in Table 2-1, lines 113-145. Hard constraints must be met otherwise the service feasibility check fails.

			Table 2-1. Service List *(lines 113 – 145)*		
8		*Soft constraints*	*See input parameters for Hard Constraints in* Table 2-1. Service List *(lines 113 – 145)*	No	Refer to descriptions in Table 2-1, lines 113-145. Soft constraints may be met. Indicate the soft constraints that have been met in this container.
18	Connection Type			Yes	4 types: Service, Enum=1; Infrastructure, Enum=2; ROADM line, Enum=3; optical-tunnel, Enum=4;
19	Resource status			No	(resource-status-type) Enum to indicate which network resources the controller should consider when computing a path. Options are ***deployed*** (default; use network resources with lifecycle-state=deployed and ignore operational-state), ***in-service*** (route around failed services; lifecycle-state=deployed and operational-state=inService), and ***planned*** (lifecycle-state=planned or any deployed/deploying state including deploy-failed, and operational-state is ignored; if ***planned*** if selected, due date must be specified in the service request)
20	Routing Metric	*Wdm hop count*		No	The number of hops in the wdm layer will be used as a metric.
21		*otn-hop-count*		No	Number of hops in the otn layer will be used as a metric.
22		*wdm-load*		No	The load of the wdm layer will be used as a metric, to avoid using heavy loaded links.
23		*otn-load*		No	The load of the otn layer will be used as a metric, to avoid using heavy loaded links.
24		*latency*		No	Total path latency is used as a metric.
25		*distance*		No	Total path distance is the metric.
26		*wdm-TE-metric*		No	Used when routing shall be performed according to specific pre-defined TE metric. Total path metric can be calculated from OMS TE-metric attribute defined in org-openroadm-link module.

27		*otn-TE-metric*		No	Used when routing shall be performed according to specific pre-defined metric associated with OTN (OTU/ODU level).
28		*adaptation-number*		No	Adaptation between layers is the metric. Total path metric can be calculated from the total number of transitions between layers.
29	Service Resiliency	*Resiliency*		No	Identity ref with the following types: unprotected, unprotected-diversely-routed, protected, restorable, and external-trigger-restorable.
30		*Revertive*		No	Specifies whether the service shall revert to its initial working path after protection switching and fault conditions have cleared.
31		*Wait to restore*		No	Time delay for switching to backup path
32		*Holdoff time*		No	Time delay for reverting to initial working path
33		*Pre-calculated backup path number*		No	Provides the target number of backup paths conforming with specific engineering rules
34		*Coupled Service*	*Service index*	Yes	Service number of the service that is disjointly routed from the failed service. Service index is the key to a list of coupled services that may be used for restoration.
35			*Service name*	No	Name of the service that is disjointly routed from the failed service
36			*Common ID*	No	Common ID of the service that is disjointly routed from the failed service
37			*Version number*	No	Service version number of the service that is disjointly routed from the failed service
38	Propose equipment			No	Whether or not this request can propose new equipment that could be used to fulfill this request. If never, the request will just use existing deployed and planned equipment. If ifNeeded, routes using existing equipment will be preferred. If always, a route with proposed equipment shall be returned, if possible. 3 types: “Never”, Enum=1; “ifNeeded”, Enum=2, “Always”, Enum=3. Default is “ifNeeded”.
39	Due date			No	Date and time service to be turned up. If time is not specified for a given date, default to midnight.

			Service will be turned up immediately if no *due date* is specified. Type: yang:*date*-and-time
40	End Date	No	Date and time service to be removed. Type: yang:*date*-and-time
41	Event Horizon Start	No	Start time to ensure that the service is routable and viable. Required resources shall be considered reserved from this time. If not provided, defaults to due-date. Type: yang:*date*-and-time
42	Event Horizon End	No	End time to ensure that the service is routable and viable. Required resources shall be considered reserved until this time. If not provided, defaults to end-date. Type: yang:*date*-and-time
43	NC code	No	Network Channel code applied to wavelength service only. This is reported against the service but may not get reflected in the service in the network (string).
44	NCI code	No	Network Channel Interface code applied to wavelength service only. This is reported against the service but may not get reflected in the service in the network (string).
45	Secondary NCI code	No	Secondary NCI code applied to wavelength service only. This is reported against the service but may not get reflected in the service in the network (string).
46	Customer	No	To be included in ticket information. This is reported against the service but may not get reflected in the service in the network (string).
47	Customer contact	No	Customer contact information to be included in ticket information. This is reported against the service but may not get reflected in the service in the network (string).
48	Operator contact	No	Operator contact information to be included in ticket information. This is reported against the service but may not get reflected in the service in the network (string).
49	CLLI network ref	No	Network-id of the clli-network layer from the network model (string)
50	Openroadm network ref	No	Network-id of the openroadm-network layer from the network model (string)
51	Openroadm topology ref	No	Network-id of the openroadm-topology layer from the network model (string)

52	SLA id			No	SLA defined for the service (string)
53	Bandwidth calendaring			No	When true, triggers the following structure allowing the description of the bandwidth calendaring options. (boolean)
54	Bandwidth calendaring parameters				Container gathering attributes describing the bandwidth calendaring options for the service.
55		BW calendaring coupled services		No	List of services that may be associated with the considered service. These services can be in service-list, temp-service-list, and versioned-service-list. The service and its coupled-service(s) may be defined on complementary time periods.
56			Service index	No	Coupled service identifier. (String)
57			Service name	No	Coupled service name for standard-service. (String)
58			Common-id	No	Coupled service identifier for temporary service. (String)
59			Version number	No	Coupled service version number for versioned service. (uint64)
60		Recurrence pattern		No	Defines a service that is active in day-of-the-week with start-time and end-time. (String)
61			recurrence id	No	Identifier of the recurrence scheme. (uint32)
62			day of the week	No	Day of the week the service is active. (enumeration)
63			Start time	No	Start time for service activation. Applies to any days of the recurrence scheme. (String)
64			End time	No	Time at which the service is deactivated. Applies to any days of the recurrence scheme. (String)
9	Service A-end	*Repeat A-end parameters from* Table 2-1. Service List*, line 36-106*		Yes	See descriptions in Table 2-1. Service List, lines 36-106.
10		*Equipment required*			Added to service-a-end container. List of equipment required for this temp service, indexed by equipment identifier
11			*Equipment type*	No	Type of equipment, value is derived from the equipment-type grouping in the common model
12			*Equipment identifier*	Yes	Unique equipment identifier, string

13			*Lifecycle state*		No	Lifecycle-state enum from common model (deployed, planned, deploying, undeploying, proposed, etc.)
14			*Equipment rack*		No	Rack identifier, string
15			*Equipment shelf*		No	Shelf identifier, string
16			*Equipment slot*		No	Slot identifier, string
17			*Equipment sub-slot*		No	Sub-slot identifier, string
18			*Is reused*		No	Boolean; if true indicates that required equipment is being reused
19			*port*			List of ports, indexed by circuit-pack-name and port-name
20				*Circuit pack name*	Yes	Circuit pack identifier, string
21				*Port name*	Yes	Port identifier, string Port is unique within the context of circuit-pack
22				*Lifecycle state*	No	Lifecycle-state enum from common model (deployed, planned, deploying, undeploying, proposed, etc.)
23		*Expected Settings and Performances*				Added to service-a-end container. Applies when connection type is optical-tunnel or infrastructure; provides performance and characteristics of wavelength services.
24			*frequency*		No	Optical channel center frequency in THz
25			*width*		No	Optical channel width in GHz
26			*Optical operational mode*		No	Optical operational mode used in path computation. May be a standard Open ROADM operational mode or a supplier specific operational mode.
27			*RX estimated OSNR*		No	Estimated OSNR for the path
28			*RX estimated GSNR*		No	Estimated GSNR (including non linear impairments) for the path
29			*Maximum output power*		No	Maximum output power in dB

30			*Minimum output power*		No	Minimum output power in dB
31	Service Z-end	*Repeat A-end parameters from* Table 2-1. Service List*, line 36-106*			Yes	See descriptions in Table 2-1. Service List, lines 36-106.
32		*Equipment required* *Repeat lines 10 - 22*				Added to service-z-end container. List of equipment required for this temp service, indexed by equipment identifier. See descriptions in lines 10 – 22.
33		*Expected Settings and Performances* *Repeat lines 23 - 30*				Added to service-z-end container. Applies when connection type is optical-tunnel or infrastructure; provides performance and characteristics of wavelength services. See descriptions in lines 23 - 30
34	Intermediate Site[12]	*CLLI*			Yes	List of intermediate sites with CLLI (string, unique site identifier) as key.
35		*Node*			Yes	List of nodes at intermediate site, indexed by node-id
36			*Node ID*		Yes	Globally unique identifier for a device length "7..63" pattern "([a-zA-Z][a-zA-Z0-9-]{5,18}[a-zA-Z0-9])"[13] A Node ID can contain letters, numbers, and hyphens. The first character must be a letter. The last character must be a letter or number. Reported against the service but may not get reflected in the service in the network.
37			*Equipment required*			List of equipment required at intermediate site, indexed by equipment-identifier
38				*Equipment type*	No	Type of equipment, value is derived from the equipment-type grouping in the common model
39				*Equipment identifier*	Yes	Unique equipment identifier, string
40				*Lifecycle state*	No	Lifecycle-state enum from common model (deployed, planned, deploying, undeploying, proposed, etc.)

[12] List of required equipment, including equipment type, state and quantity over entire route of the service.
[13] The pattern for Node ID is incorrect in the Open ROADM YANG model as it doesn't allow the length to be extended past 20 characters. This will be fixed in a future release of the YANG models.

41				*Equipment rack*		No	Rack identifier, string
42				*Equipment shelf*		No	Shelf identifier, string
43				*Equipment slot*		No	Slot identifier, string
44				*Equipment sub-slot*		No	Sub-slot identifier, string
45				*Is reused*		No	Boolean; if true indicates that required equipment is being reused
46				*Port*			List of ports, indexed by circuit-pack-name and port-name
47					*Circuit pack name*	Yes	Circuit pack identifier, string
48					*Port name*	Yes	Port identifier, string Port is unique within the context of circuit-pack
49					*Lifecycle state*	No	Lifecycle-state enum from common model (deployed, planned, deploying, undeploying, proposed, etc.)
50	Requested Service Topology	*Topology* *See* Table 2-1. Service List *lines 179 to 247*					*See* Table 2-1. Service List *lines 179 to 247*
51		*Backup topology* *See* Table 2-1. Service List *lines 249 - 254*					*See descriptions in* Table 2-1. Service List *lines 249 - 254*
52		*Network topology* *See* Table 2-1. Service List *lines 255 - 264*					*See descriptions in* Table 2-1. Service List *lines 255 - 264*
53		*Network Backup Topology* *See* Table 2-1. Service List *lines 265 - 268*					*See descriptions in* Table 2-1. Service List *lines 265 - 268*
54	Supporting service hierarchy	*See* Table 2-3 Temp Service List *lines 37 - 71*					*See descriptions in* Table 2-3 Temp Service List *lines 37 - 71*
55		*Service metrics*				No	Service metrics associated with the primary service path

56	Primary path metrics		*latency*		No	Provides one-way end-to-end latency (in milliseconds) of a service-path; may be measured or estimated
57			*distance*		No	Provides end-to-end distance of a service-path in km.
58			*Hop count*			Number of path hops
59				*WDM hop count*	No	Number of hops at the WDM layer
60				*OTN hop count*	No	Number of hops at the OTN layer
61			*TE Metric*			Cost of end-to-end path
62				*WDM TE metric*	No	Cost at the WDM layer
63				*OTN TE metric*	No	Cost at the OTN layer
64	Backup path metrics	*Backup path ID*			Yes	
65		*Service metrics* *See lines 55 - 63*				See descriptions in lines 55 - 63
66	*Maximum regeneration options*				No	Maximum number of regeneration elements that may be returned. The RNC may return fewer regen options than the max. Default value is 1.
67	*Regeneration option list*					List of regeneration options, indexed by regeneration-option-rank.
68		*Regeneration option rank*			Yes	Integer to indicate rank order of regeneration options. Rank is derived based on optical performance or other metrics.
69		*Regeneration CLLI list*				List of CLLIs (sites) for regeneration, ordered by sequence-id
70			*Sequence ID*		Yes	Sequence id defines the order of regeneration locations in the A to Z direction
71			*Regeneration CLLI*		No	CLLI (site identifier) for planned or deployed regenerator
72			*Lifecycle state*		No	Lifecycle state of regenerator (typedef lifecycle-state), typically planned or deployed

73	*Existing service attributes*			Used when the feasibility check is performed on an existing service to describe attributes related to the existing service.
74		*Is existing*	No	Boolean; if true, the temp service create is for an existing service, with or without new constraints.
75		*Existing service name*	No	When is-existing = true, provides the name of the existing service.
76		*Reuse existing resources*	No	If true, the temp service should use the existing service resources/equipment wherever possible.
77		*Resusable existing resources*	No	When reuse-existing-resources is true, this provides a list of existing resource types that may be reused. Defined by typedef existing-resource-reuse-type (values are regenerator, wavelength, spectrum-portion, xponder, all)

5.3 Service Feasibility Check Bulk RPC

This RPC checks feasibility for multiple services. It takes a list of potential services and requests the RNC or Open ROADM Controller to analyze feasibility collectively and return results on connectivity, equipment availability and reachability. It ensures that a given resource is not used more than once. No resources are reserved, provisioned or planned because of this operation. If this request passed the initial validation and was accepted for processing, a service RPC result notification shall be sent once the request completes processing. Table 5-5 lists the service feasibility check bulk RPC and its associated input parameters.

Table 5-5 Service Feasibility Check Bulk PRC and Input Parameters

	Input Parameter		***Manda-tory***	***Descriptions***
1	SDNC Request Header	*Request ID*	No	From original system requesting for the service. Uniquely generated by calling system. (string)
2		*RPC Action*	No	Service feasibility check bulk, Enum=14
3		*Notification url*	No	URL for asynchronous response (string)
4		*Request System ID*	No	Identifier of application initiates the request (string)
5	Service Request List		Yes	List of the potential services for bulk feasibility check, indexed by common-id

	Input Parameter		*Manda-tory*	*Descriptions*
6		Common ID	Yes	Service order #, or identifier to be used by the ROADM controller to identify routing constraints received from planning applications. (string)
7		For each service in line 5, repeat parameters in Table 5-3Table 5-3 Service Feasibility Check RPC and Input Parameters lines 6 - 70		See descriptions in Table 5-3 lines 6 - 70

The Open ROADM Service Model defines the synchronous response to the service feasibility check bulk RPC. Table 5-6 lists the output parameters in version 2.2.

Table 5-6 Synchronous Response to Service Feasibility Check Bulk RPC

Output	*Field Name*		*Manda-tory*	*Note*
Configuration Response Common	*Request ID*		Yes	The request ID from the request message for which this is the response (string)
	Response Code		Yes	One of the codes defined for success or error (string)
	Response Message		No	Message included for error code (string)
	Ack-final-indicator		Yes	Indicates if this is the last response that the client should expect (string).
Service Response List	*Common ID*		Yes	
	Response Parameters	For each service in the feasibility check bulk, repeat parameters in Table 5-4 lines 6 - 77.		

5.4 Service Delete RPC

This RPC is for the SDN Controller to request the RNC or Open ROADM Controller to remove an existing service either immediately or in future. If this request passed initial validation and was accepted for processing, once the processing completes, a service RPC result notification shall be sent. Once the service has been deleted, it will no longer appear in the service list.

The parameters included in the service delete RPC are described in Table 5-7 below.

Table 5-7 Service Delete RPC and Input Parameters

	Input Parameter		***Manda-tory***	***Descriptions***
1	SDNC Request Header	*Request ID*	No	From original system requesting for the service. Uniquely generated by calling system. (string)
2		*RPC Action*	No	Service delete, Enum=3
3		*Notification URL*	No	URL for asynchronous response (string)
4		*Request System ID*	No	Identifier of application initiates the request (string)
5	Service Delete Request Info	*Service name*	Yes	Identifier for the service to be deleted in the ROADM network, e.g., CLFI, CLCI, etc. (string)
6		*Due date*	No	Date and time service to be turned down. If time is not specified for a given date, default to midnight. Service will be turned down immediately if no *due date* is specified. Type: yang:*date*-and-time
7		*Tail retention*	Yes	"Yes", Enum=1, tails are left intact. "No", Enum=2, tails are deleted.

The Open ROADM Service Model version 2.2 defines the synchronous response to the service delete RPC. The synchronous response only contains the configuration response common body, see Table 5-8Table 5-8 below.

Table 5-8 Synchronous Response to Service Delete RPC

Output	***Field Name***	***Manda-tory***	***Note***
Configuration Response Common	*Request ID*	Yes	The request ID from the request message for which this is the response (string)
	Response Code	Yes	One of the codes defined for success or error (string)
	Response Message	No	Message included for error code (string)
	Ack-final-indicator	Yes	Indicates if this is the last response that the client should expect (string).

5.5 Equipment Notification RPC

This RPC is for the RNC or Open ROADM Controller to notify the SDN controller that new equipment, e.g., a new ROADM node, was self-discovered in the network. The parameters included in the equipment notification RPC are described in Table 5-9.

Table 5-9 Equipment Notification RPC and Input Parameters

	Parameter		***Manda-tory***	***Note***
1	*SDNC Request Header*	*Request ID*	No	From original system requesting for the service. Uniquely generated by calling system. (string)
2		*RPC Action*	No	Equipment notification, Enum=4
3		*Notification url*	No	URL for asynchronous response (string)
4		*Request System ID*	No	Identifier of application initiates the request (string)
5	*Equipment ID*		Yes	Identifier of the equipment (e.g. ROADM node). This is also the primary key for updates. (string)
6	*Equipment Name*		No	Equipment name and description (string)
7	*Equipment Type*		Yes	ROADM, Xponder, etc. (string) The set of valid values is derived from the equipment-type grouping used in the device model.
8	*Equipment Vendor*		Yes	Name of the vendor for the equipment. (string)
9	*Equipment customer*		No	Name of customer to which this equipment belongs. (string)
10	*Equipment CLLI*		Yes	Expected 11 char CLLI but minimally 8 character CLLI of the equipment being added/updated. Note that the same equipment-clli cannot be allowed to map to more than one controller-id. (string)
11	*Equipment IP*		No	Format is IP address. (string)

	Parameter	*Manda-tory*	*Note*
12	*Controller ID*	Yes	Identifier of the RNC which controls the equipment. (string)

The Open ROADM Service Model version 2.2 defines the synchronous response to the equipment notification RPC. The synchronous response only contains the "Configuration Response Common" body, refer to Table 5-8.

5.6 Temp Service Create RPC

This RPC is for requesting the Open ROADM Controller or RNC to compute a service path and reserve the wavelengths assigned to the service. The temporary services will be converted to the normal services upon creation of a service request from the SDN controller using the matching Common ID.

A temp service can be converted to a normal service using the service-create RPC. Once converted to a normal service, that service will no longer show in the temp service list.

The parameters included in the RPC are described in Table 5-10.

Table 5-10 Temp Service Create RPC and Input Parameters

	Input Parameter			*Manda-tory*	*Descriptions*
1	Common ID			Yes	Service order #, or identifier to be used by ROADM controller and planning applications for routing constraints etc. (string)
2	SDNC Request Header	*Request ID*		No	From original system requesting for the service. Uniquely generated by calling system. (string)
3		*RPC Action*		No	Temp service create, Enum=5
4		*Notification url*		No	URL for asynchronous response (string)
5		Request System ID		No	Identifier of application initiates the request (string)
6	Repeat Table 5-1 Service Create RPC and Input Pramemters from line 7 to line 39.				
7	Service A end	requesting-interface-properties			Triggered when connection-type = optical-tunnel. Provides external pluggable/Xponder characteristics
8			*supported-operational-modes*		List of supported operational modes

	Input Parameter				Manda-tory	Descriptions
9				*preference*	Yes	Preference/priority associated with an operational mode. 1 is highest priority. (int16)
10				*operational-mode-id*	No	Operational mode identifier. (string)
11			*min-frequency*		No	Minimum frequency to be set. (org-openroadm-common-optical-channel-types:frequency-THz)
12			*max-frequency*		No	Maximum frequency to be set. (org-openroadm-common-optical-channel-types:frequency-THz)
13			*min-granularity*		No	Minimum NMC width. (org-openroadm-common-optical-channel-types:frequency-GHz)
38	Service Z-end	Repeat parameters from line 6 to line 13 for Service Z-end				
14	Repeat Table 5-1 Service Create RPC and Input Pramemters from line 41 to line 67.					

The synchronous response to the Temp Service Create RPC has the same output parameters in the Service Create RPC section. Refer to Table 5-2 Synchronous Response to Service Create RPC.

5.7 Temp Service Delete RPC

This RPC is to request the RNC or Open ROADM Controller to remove wavelengths that were reserved via a temporary service create RPC.

This command is typically used to cancel a temp service if it is not to be converted to a normal service.

The parameters included in the Temp Service Delete RPC are described in Table 5-11 below.

Table 5-11 Temp Service Delete RPC and Input Parameter

	Input Parameter	Manda-tory	Descriptions
1	Common ID	Yes	The Common ID in the Temp Service Create request before.

The synchronous response to the Temp Service Delete RPC only contains the “Configuration Response Common” body. Refer to Table 5-8.

5.8 Service Roll RPC

This RPC is to request the RNC or Open ROADM Controller to change the path of a service while keeping the same A and Z end points. The new path must comply with the routing constraints that were imposed on the service initially. This capability

is mostly exercised by the SDN Controller following a network re-optimization request (Section 5.14) through which the RNC identified more optimal paths for some embedded services.

If this request passed the initial validation and was accepted for processing, a service RPC result notification shall be sent once the request completes processing

The parameters included in the Service Roll RPC are described in Table 5-12.

Table 5-12 Service Roll RPC and Input Parameters

	Input Parameter	***Manda-tory***	***Descriptions***
1	Service Name	Yes	Identifier for the service to be rolled in the ROADM network, e.g., CLFI, CLCI, etc. (string)
2	Due Date	No	Date and time service to be rolled. If time is not specified for a given date, default to midnight. Service will be rolled immediately if no *due date* is specified. Type: yang:*date*-and-time

The synchronous response to the Service Roll RPC is listed in Table 5-13.

Table 5-13 Synchronous Response to Service Roll RPC and Output Parameters

	Output Parameter	***Manda-tory***	***Descriptions***
1	Status	Yes	2 types: “Successful”, Enum=1; “Failed”, Enum=2
2	Status message	No	Details about the status (string)

5.9 Service Reconfigure RPC

This RPC provides the capability to request the RNC or Open ROADM Controller to change the service to different terminating equipment, i.e., re-home the service, to change the service path, and to route the service with different routing constraints etc. If this request passed the initial validation and was accepted for processing, a service RPC result notification shall be sent once the request completes processing. Table 5-14 lists the Service Reconfigure RPC and input parameters.

Table 5-14 Service Reconfigure RPC and Input Parameters

	Input Parameter	*Manda-tory*	*Descriptions*
1	Service Name	Yes	Existing identifier for the service to be reconfigured in the ROADM network, e.g., CLFI, CLCI, etc.
2	New Service Name	No	New identifier for the service to be reconfigured in the ROADM network, e.g., CLFI, CLCI, etc.
3	Common ID	No	Service order #, or identifier to be used by ROADM controller and planning applications for routing constraints etc. (string)
4	Connection Type	No	4 types: Service, Enum=1; Infrastructure, Enum=2; ROADM line, Enum=3; optical-tunnel, Enum=4;
5	Repeat parameters in Table 5-1 Service Create RPC and Input Prameters from line 6 to line 14.		Parameters blow line 4 in this table are the same as in a Service Create RPC line 8 to line 67 in Table 5-1 Service Create RPC and Input Prameters.

The synchronous response to the Service Reconfigure RPC is the same as listed in Table 5-13.

5.10 Service Restoration RPC

This RPC is to restore the service disrupted by regen failures. The SDN Controller receives notification from the RNC or Open ROADM Controller whether the service can be restored either permanently or temporarily by a spare regen. The SDN Controller then instructs the RNC to restore the service using spare regen(s). Service restoration is to be carried out immediately. If this request passed the initial validation and was accepted for processing, a service RPC result notification shall be sent once the request completes processing. Table 5-15 lists the Service Restoration RPC and input parameters.

Table 5-15 Service Restoration RPC and Input Parameters

	Input Parameter	*Manda-tory*	*Descriptions*
1	Service Name	Yes	Identifier for the service to be restored in the ROADM network, e.g., CLFI, CLCI, etc.
2	Option	Yes	2 types: “Permanent” Enum=1; “Temporary”, Enum=2. When “Permanent” is selected, a spare regen can be used to restore the service permanently without reverting back to the

	Input Parameter	*Manda-tory*	*Descriptions*
			original regen. When “Temporary” is selected, a spare regen can be used to restore the service temporarily. The service then needs to be reverted back to the original regen transponder.

The synchronous response to the Service Restoration RPC is the same as listed in Table 5-13.

5.11 Service Reversion RPC

This RPC is to revert the service that was restored or rerouted temporarily to the original equipment or path. Service reversion is expected to be performed in a maintenance window with a due date. If this request passed the initial validation and was accepted for processing, a service RPC result notification shall be sent once the request completes processing. The Service Reversion RPC and input parameters are listed in Table 5-16.

Table 5-16 Service Reversion RPC and Input Parameters

	Input Parameter	*Manda-tory*	*Descriptions*
1	Service Name	Yes	Existing identifier for the service to be reverted in the ROADM network, e.g., CLFI, CLCI, etc.
2	Due Date	No	Date and time service to be reverted. If time is not specified for a given date, default to midnight. Service turned up immediately if no *due date* is specified. Type: yang:*date*-and-time

The synchronous response to the Service Reversion RPC is the same as listed in Table 5-13.

5.12 Service Reroute RPC

This RPC can be used by the SDN Controller to restore a service that is affected by ROADM line failures such as fiber cut, optical amplifier failure, etc. Service reroute is to be carried out immediately without consideration of any routing constraints.

Note:

Since service re-route is always on a temporary basis, the RNC must mark the equipment and wavelengths in the original path as “Out of Service Maintenance” so that the rerouted service can be reverted back through “Service Reversion”.

If this request passed the initial validation and was accepted for processing, a service RPC result notification shall be sent once the request completes processing. The Service Reroute RPC and input parameter are listed in

Table 5-17.

Table 5-17 Service Reroute RPC and Input Parameter

	Input Parameter		***Manda-tory***	***Descriptions***
1	Service Name		Yes	Existing identifier for the service to be rerouted in the ROADM network, e.g., CLFI, CLCI, etc.
2	SDNC Request Header	*Request ID*	No	From original system requesting for the service. Uniquely generated by calling system. (string)
3		*RPC Action*	No	Service create, Enum=1
4		*Notification url*	No	URL for asynchronous response (string)
5		Request System ID	No	Identifier of application initiates the request (string)
6	Routing Metric	*Wdm hop count*	No	The number of hops in the wdm layer will be used as a metric.
7		*otn-hop-count*	No	Number of hops in the otn layer will be used as a metric.
8		*wdm-load*	No	The load of the wdm layer will be used as a metric, to avoid using heavy loaded links.
9		*otn-load*	No	The load of the otn layer will be used as a metric, to avoid using heavy loaded links.
10		*latency*	No	Total path latency is used as a metric.
11		*distance*	No	Total path distance is the metric.
12		*wdm-TE-metric*	No	Used when routing shall be performed according to specific pre-defined TE metric. Total path metric can be calculated from OMS TE-metric attribute defined in org-openroadm-link module.
13		*otn-TE-metric*	No	Used when routing shall be performed according to specific pre-defined metric associated with OTN (OTU/ODU level).

	Input Parameter			***Manda-tory***	***Descriptions***
14		*adaptation-number*		No	Adaptation between layers is the metric. Total path metric can be calculated from the total number of transitions between layers.
15	Service Resiliency	*Resiliency*		No	Identity ref with the following types: unprotected, unprotected-diversely-routed, protected, restorable, and external-trigger-restorable.
16		*Revertive*		No	Specifies whether the service shall revert to its initial working path after protection switching and fault conditions have cleared.
17		*Wait to restore*		No	Time delay for switching to backup path
18		*Holdoff time*		No	Time delay for reverting to initial working path
19		*Pre-calculated backup path number*		No	Provides the target number of backup paths conforming with specific engineering rules
20		*Coupled Service*	*Service index*	Yes	Service number of the service that is disjointly routed from the failed service. Service index is the key to a list of coupled services that may be used for restoration.
21			*Service name*	No	Name of the service that is disjointly routed from the failed service
22			*Common ID*	No	Common ID of the service that is disjointly routed from the failed service
23			*Version number*	No	Service version number of the service that is disjointly routed from the failed service
24	Connection Type			Yes	4 types: Service, Enum=1; Infrastructure, Enum=2; ROADM line, Enum=3; optical-tunnel, Enum=4;

The synchronous response to the Service Reroute RPC is listed in Table 5-18.

Table 5-18 Synchronous Response to Service Reroute RPC and Output Parameters

Output Parameter	***Manda-tory***	***Descriptions***

1	Status					Yes	2 types: “Successful”, Enum=1; “Failed”, Enum=2
2	Status message					No	Details about the status (string)
3	Hard Constraints	*Customer Code*				No	For selecting tagged equipment on which to route a service. If more than one customer code is provided, they will be treated as an ordered list. (string)
4		*Operational mode*				No	An operational mode can be specified to be used as a constraint. Leaflist (string)
5		*General*	*Diversity*	*Existing service*		No	Diverse from existing services identified by facility CLFI, list. (string) Constraints are either general or co-routing. Under general constraints, there are diversity, exclude, include and latency constraints.
6				*Existing service applicability*	*Site*	No	Site identifies the CLLI (Boolean)
7					*Node*	No	Refer to Table 5-1 Service Create RPC and Input Prameters line 15 (Boolean)
8					*SRLG*	No	Shared Risk Link Group data, (Boolean)
9			*Exclude*	*Fiber bundle*		No	Fiber segment usually defined by SRLG (string), list.
10				*Site*		No	Site identifies the CLLI, list.
11				*Node*		No	Refer to Table 5-1 Service Create RPC and Input Prameters line 15, list.
12				*Supporting service name*		No	Refer to Table 5-1 Service Create RPC and Input Prameters line 49.
13			*Include*	*Fiber bundle*		No	Refer to line 8.
14				*Site*		No	Refer to line 9.
15				*Node*		No	Refer to Table 5-1 Service Create RPC and Input Prameters line 15, list.
16				*Supporting service name*		No	Refer to Table 5-1 Service Create RPC and Input Prameters line 49.

17			*Latency*	*Maximum latency*		No	Refer to Table 5-1 Service Create RPC and Input Prameмters line 54.
18		*Co-routing*	*Existing Service*			No	The existing service that is to be co-routed, list.
19	Soft Con-straints	*Repeat line 3 to line 17 for soft constraints.*				No	

5.13 Service Reroute Confirm RPC

This RPC is to confirm the service reroute. The input parameters are described in Table 5-19. If this request passed the initial validation and was accepted for processing, a service RPC result notification shall be sent once the request completes processing.

Table 5-19 Service Reroute Confirm RPC and Input Parameters

1	***Input Parameter***					***Manda-tory***	***Descriptions***
2	Service Name					Yes	Identifier for the service to be rerouted in the ROADM network, e.g., CLFI, CLCI, etc. (string)
3	Hard Constraints	*Customer Code*				No	For selecting tagged equipment on which to route a service. If more than one customer code is provided, they will be treated as an ordered list. (string)
4		*General*	*Diversity*	*Existing service*		No	Diverse from existing services identified by facility CLFI, list. (string) Constraints are either general or co-routing. Under general constraints, there are diversity, exclude, include and latency constraints.
5				*Existing service applicability*	*Site*	No	Site identifies the CLLI (Boolean)
6					*Node*	No	Refer to Table 5-1 Service Create RPC and Input Prameмters line 15, list.
7					*SRLG*	No	Shared Risk Link Group data, (Boolean)
8			*Exclude*	*Fiber bundle*		No	Fiber segment usually defined by SRLG (string), list.

9				*Site*	No	Site identifies the CLLI, list.
10				*Node*	No	Refer to Table 5-1 Service Create RPC and Input Prameмters line 15, list.
11				*Supporting service name*	No	Refer to Table 5-1 Service Create RPC and Input Prameмters line 49.
12			*Include*	*Fiber bundle*	No	Refer to line 8.
13				*Site*	No	Refer to line 9.
14				*Node*	No	Refer to Table 5-1 Service Create RPC and Input Prameмters line 15, list.
15				*Supporting service name*	No	Refer to Table 5-1 Service Create RPC and Input Prameмters line 49.
16			*Latenc y*	*Maximum latency*	No	Refer to Table 5-1 Service Create RPC and Input Prameмters line 54.
17		*Co-routing*	*Existing Service*		No	The existing service that is to be co-routed, list.
18	Soft Constrai nts	*Repeat line 3 to line 17 for soft constraints.*			No	

The synchronous response to the Service Reroute Confirm RPC is the same as listed in Table 5-13.

5.14 Network Re-optimization RPC

As the network topology changes over time, the SDN Controller can periodically request the RNC to check whether any embedded services can be routed more efficiently without violating any routing constraints imposed on the services. The parameters included in the network re-optimization RPC are described in Table 5-20.

Table 5-20 Network Re-optimization RPC and Input Parameters

	Input Parameter	***Manda-tory***	***Descriptions***
1	Service Name	No	Identifier for the service to be checked by the RNC for re-optimization in the ROADM network, e.g., CLFI, CLCI, etc.

	Input Parameter	*Manda-tory*	*Descriptions*
2	A-end	No	Services whose A-ends are terminated at the specified office location are to be checked by the RNC for re-optimization.
3	Z-end	No	Services whose Z-ends are terminated at the specified office location are to be checked by the RNC for re-optimization.
4	Pass-through	No	Services that are pass-through (either via regen or express) at the specified office location are to be checked by the RNC for re-optimization.
5	Customer Code	No	Services that belong to the specified customer are to be checked by the RNC for re-optimization.

The synchronous response to the Network Re-optimization RPC is listed in Table 5-21.

Table 5-21 Synchronous Response to Network Re-optimization RPC and Output Parameters

	Output Parameter	*Manda-tory*	*Descriptions*
1	Status	Yes	2 types: “Successful”, Enum=1; “Failed”, Enum=2
2	Status message	No	Details about the status (string)
3	Optimization Candidate	No	Specify each of the services that can be optimized (string)

5.15 BER Test RPC

The ber-test rpc is used to perform BER tests for a service in the service-list. The input of this rpc includes the sdnc-request-header container, service-name (mandatory), and a ber-options container which defines target-prefec-ber, duration, pm-polling-timeout and retry-attempts. The output of this rpc consists of the configuration-response-common container which lists request-id, response code, response message, and ack-final-indicator. Below is the tree view data model of the ber-test rpc:

```
  +---x ber-test
      |  +---w input
      |  |  +---w sdnc-request-header
      |  |  |  +---w request-id?                    string
      |  |  |  +---w rpc-action?                    rpc-actions
```

```
|  |  |  +---w notification-url?                string
|  |  |  +---w request-system-id?               string
|  |  +---w service-name                   string
|  |  +---w ber-options
|  |     +---w target-prefec-ber?               decimal64
|  |     +---w duration?                        uint16
|  |     +---w pm-polling-timeout?              uint16
|  |     +---w retry-attempts?                  uint16
|  +--ro output
|     +--ro configuration-response-common
|        +--ro request-id                   string
|        +--ro response-code                string
|        +--ro response-message?            string
|        +--ro ack-final-indicator          string
```

5.16 Service RPC BER Test Async Callback RPC

The service-rpc-ber-test-async-callback is the notification rpc that the ROADM network controller invokes on the carrier system, i.e., the service provider's SDN controller to report BER test results. It returns service-identifiers (service-name, common-id, version-number), ber-options parameters, timestamps of both the initial rpc and its callback, service line rate, and details of the BER test results. The callback provides measured-prefec-ber at service a-end and z-end, and a pass or fail status of the BER tests. The output conforms to the configuration-response-common container. Below is the tree-view data model of the service-rpc-ber-test-async-callback:

```
+---x service-rpc-ber-test-async-callback
       +---w input
       |  +---w configuration-response-common
       |  |  +---w request-id                       string
       |  |  +---w response-code                    string
       |  |  +---w response-message?                string
       |  |  +---w ack-final-indicator              string
       |  +---w service-identifiers
       |  |  +---w service-name?                    string
       |  |  +---w common-id?                       string
       |  |  +---w version-number?                  uint64
       |  +---w rpc-timestamp?                      yang:date-and-time[14]
       |  +---w timestamp?                          yang:date-and-time[15]
       |  +---w ber-results
       |     +---w ber-options
       |     |  +---w target-prefec-ber?            decimal64
       |     |  +---w duration?                     uint16
       |     |  +---w pm-polling-timeout?           uint16
       |     |  +---w retry-attempts?               uint16
       |     +---w line-rate?                       uint64
       |     +---w a-end-ber-measurement
       |     |  +---w ber-passed?                   boolean
       |     |  +---w target-prefec-ber?            decimal64
```

[14] Updated timestamp in v4.1.0 from unit16 to yang:date-and-time.
[15] Updated timestamp in v4.1.0 from unit16 to yang:date-and-time.

```
|      |  +---w measured-prefec-ber?          decimal64
|      +---w z-end-ber-measurement
|         +---w ber-passed?                    boolean
|         +---w target-prefec-ber?             decimal64
|         +---w measured-prefec-ber?           decimal64
+--ro output
```

There are notifications associated with the service-rpc-ber-test-async-callback. Please refer to section xx for details.

5.17 Service Create result Notifications request (Callback) RPC

The service-create-result-notification-request rpc is the callback notification from the ROADM network controller to the carrier system, i.e., the service provider's SDN controller to report the service-create results. This callback input consists of the configuration-response-common container and the service-identifiers container to specify service-name, common-id and version-number. The output follows the configuration-response-common container. Below is the tree-view data model of the service-create-result-notification-request rpc:

```
+---x service-create-result-notification-request
     |  +---w input
     |  |  +---w configuration-response-common
     |  |  |  +---w request-id                 string
     |  |  |  +---w response-code              string
     |  |  |  +---w response-message?          string
     |  |  |  +---w ack-final-indicator        string
     |  |  +---w service-identifiers
     |  |     +---w service-name?              string
     |  |     +---w common-id?                 string
     |  |     +---w version-number?            uint64
     |  +--ro output
     |     +--ro configuration-response-common
     |        +--ro request-id                 string
     |        +--ro response-code              string
     |        +--ro response-message?          String
     |        +--ro ack-final-indicator        string
```

5.18 Service Delete Result Notification Request (Callback) RPC

The service-delete-result-notification-request rpc is the callback notification from the ROADM network controller to the carrier system, i.e., the service provider's SDN controller to report the service-delete results. The input of this callback consists of the configuration-response-common container and the service-identifiers container to specify service-name, common-id and version-number. The output follows the configuration-response-common container. Below is the tree-view data model of the service-delete-result-notification-request rpc:

```
+---x service-delete-result-notification-request
     |  +---w input
     |  |  +---w configuration-response-common
     |  |  |  +---w request-id                      string
     |  |  |  +---w response-code                   string
     |  |  |  +---w response-message?               string
     |  |  |  +---w ack-final-indicator             string
     |  |  +---w service-identifiers
```

```
|  |     +---w service-name?                      string
|  |     +---w common-id?                         string
|  |     +---w version-number?                    uint64
|  +--ro output
|     +--ro configuration-response-common
|        +--ro request-id                         string
|        +--ro response-code                      string
|        +--ro response-message?                  string
|        +--ro ack-final-indicator                string
```

5.19 Controller parameter settings RPC

The Controller-parameter-settings RPC, introduced in Release 7.0 allows to set parameters associated with the controller behavior. The operator can use it to customize the controller operation, including the way the path computation element will calculate paths according to the metrics, the regeneration policy, the spectrum assignment, and the way non rpc related notification shall be triggered. The RPC is also used to define different Service Level Agreements (SLAs).

The RPC structure is the same as the one of the controller-behavior-settings container, used to store the settings in the service Data Store, thus allowing to set all its parameters. The purpose of the corresponding attributes is detailed in section Controller behavior settings3. Following is the tree-view data model of the controller-parameter-settings RPC:

```
+---x controller-parameters-setting
    +---w input
    |  +---w sdnc-request-header
    |  |  +---w request-id?          string
    |  |  +---w rpc-action?          rpc-actions
    |  |  +---w notification-url?    string
    |  |  +---w request-system-id?   string
    |  +---w non-rpc-related-notification-settings
    |  |  +---w non-rpc-related-notification-url-list*
                  [non-rpc-related-notification-type]
    |  |     +---w non-rpc-related-notification-type
                                 non-rpc-related-notification-type
    |  |     +---w notification-url?                  string
    |  |     +---w events-disabling-notification*     notification-events
    |  |     +---w events-triggering-notification*    notification-events
    |  +---w spectrum-filling
    |  |  +---w spectrum-filling-rules* [rule-id]
    |  |     +---w rule-id                       uint16
    |  |     +---w priority?                     uint8
    |  |     +---w RMSA-policy?                  rmsa-policy
    |  |     +---w spectrum-range-of-appliance
    |  |        +---w spectrum-portion-id?                uint8
    |  |        +---w start-edge-frequency?
                     org-openroadm-common-optical-channel-types:frequency-THz
    |  |        +---w stop-edge-frequency?
                     org-openroadm-common-optical-channel-types:frequency-THz
    |  |        +---w dedicated-customer*                 string
    |  |        +---w non-authorized-customer*            string
```

```
|  |        +---w dedicated-signal-bandwidth-multiple?   uint8
|  +---w margins
|  |  +---w minimum-fiber-attenuation-bol-margin?
                                  org-openroadm-common-link-types:ratio-dB
|  |  +---w threshold-observed-vs-design-attenuation?
                                  org-openroadm-common-link-types:ratio-dB
|  |  +---w threshold-observed-vs-design-margin?
                                  org-openroadm-common-link-types:ratio-dB
|  |  +---w minimum-osnr-margins* [margin-id]
|  |     +---w margin-id                    string
|  |     +---w minimum-osnr-margin-value?
                                  org-openroadm-common-link-types:ratio-dB
|  |     +---w line-rates*                  uint64
|  |     +---w modulation-formats*
                org-openroadm-common-optical-channel-types:modulation-format
|  +---w metrics-policy
|  |  +---w composite-metric-versus-selective?   boolean
|  +---w regeneration-policy
|  |  +---w global-placement?      enumeration
|  |  +---w on-path-positioning?   enumeration
|  |  +---w path-symmetry?         boolean
|  |  +---w preferred-sites*       string
|  |  +---w forbidden-sites*       string
|  +---w default-behaviour
|  |  +---w default-backup-path-number?   uint16
|  |  +---w reversion?                    boolean
|  |  +---w wait-to-restore?              uint64
|  |  +---w holdoff-time?                 uint64
|  +---w sla-definition
|  |  +---w sla-parameters* [sla-id]
|  |     +---w sla-id                  string
|  |     +---w preemption?             boolean
|  |     +---w restoration-priority?   uint8
|  +---w failure-case-list
|     +---w failure-case* [failure-case-id]
|        +---w failure-case-id    uint32
|        +---w failure-type?      enumeration
|        +---w nodes*             string
|        +---w logical-links*     string
|        +---w physical-links*    string
+--ro output
   +--ro configuration-response-common
   |  +--ro request-id             string
   |  +--ro response-code          string
   |  +--ro response-message?      string
   |  +--ro ack-final-indicator    string
   +--ro unsupported-customization-options*   string
```

5.20 Optical tunnel create RPC

Optical-tunnel-create RPC is used in the second step of an optical tunnel service creation, addressing alien wavelength use case. After a temp-service-create request has been exercised by the external controller, if a path has been found to create an optical tunnel between two SRG's PPs, the RNC notifies the external controller that a path has been found, provides the path computation results which includes estimated performances, as well as the settings to apply on the external transponder or pluggable. Indeed, the optical tunnel to be created also implies that the pluggables (not handled by the RNC) are correctly set, and that an optical power is launched at the correct wavelength, so that control loops in the ROADMs can operates. After it has configured the pluggable to do so, the external controller confirms to the RNC that this path fits with initial constraints, and that the optical tunnel can be created using this RPC.

Following is the tree-view data model of the optical-tunnel-create RPC:

```
+---x optical-tunnel-create
     |  +---w input
     |  |  +---w service-name    string
     |  |  +---w common-id       string
     |  |  +---w set-frequency
                               org-openroadm-common-optical-channel-types:frequency-THz
     |  |  +---w full-bandwidth-at-3dB
                               org-openroadm-common-optical-channel-types:frequency-GHz
     |  |  +---w full-bandwidth-at-10dB
                               org-openroadm-common-optical-channel-types:frequency-GHz
     |  |  +---w sdnc-request-header
     |  |     +---w request-id?          string
     |  |     +---w rpc-action?          rpc-actions
     |  |     +---w notification-url?    string
     |  |     +---w request-system-id?   string
     |  +--ro output
     |     +--ro configuration-response-common
     |        +--ro request-id             string
     |        +--ro response-code          string
     |        +--ro response-message?      string
     |        +--ro ack-final-indicator    string
```

5.21 Optical tunnel request cancel RPC

Optical-tunnel-request-cancel RPC is used in the second step of an optical tunnel service creation, when the settings provided by the RNC for the pluggable can not be applied by the external controller to pluggables or if the expected performances do not match assumptions made by the external controller.

The process for optical tunnel creation is interrupted by the external controller through this RPC. This allows the RNC releasing the resources that were reserved during the first step (temp-service-create request).

Following is the tree-view data model of the optical-tunnel-request-cancel RPC:

```
+---x controller-parameters-setting
     |  +---w input
     |  |  +---w service-name?          string
     |  |  +---w common-id?             string
     |  |  +---w sdnc-request-header
     |  |     +---w request-id?          string
```

```
|  |     +---w rpc-action?          rpc-actions
|  |     +---w notification-url?    string
|  |     +---w request-system-id?   string
|  +--ro output
|     +--ro configuration-response-common
|        +--ro request-id             string
|        +--ro response-code          string
|        +--ro response-message?      string
|        +--ro ack-final-indicator    string
```

5.22 Add openroadm operational modes to catalog RPC

The add-openroadm-operational-modes-to-catalog RPC is used to fill the first part of the catalog dedicated to the description of the OpenROADM optical specifications. The catalog is maintained in the RNC data store. At controller initialization only the model is there and the catalog is empty. This dedicated rpc is used to fill the catalog one shot, with all modes corresponding to current state of the art of the OpenROADM specification. The Body of the rpc containing the translation of all defined specifications is publicly released with the models in the form of an xml or json file to be imported as a collection of a RESTCONF client application.

```
+---x add-openroadm-operational-modes-to-catalog
     |  +---w input
     |  |  +---w sdnc-request-header
     |  |  |  +---w request-id?          string
     |  |  |  +---w rpc-action?          rpc-actions
     |  |  |  +---w notification-url?    string
     |  |  |  +---w request-system-id?   string
     |  |  +---w operational-mode-info
     |  |     +---w grid-parameters
                    … grid parameters as they appear in OM-catalog (line 2-5)
     |  |     +---w xponders-pluggables
     |  |     |  +---w xponder-pluggable-openroadm-operational-mode*
                                                  [openroadm-operational-mode-id]
     |  |     |     … xponder parameters as they appear in OM-catalog (line 8-31)
     |  |     +---w roadms
     |  |     |  +---w Express
     |  |     |  |  +---w openroadm-operational-mode* [openroadm-operational-mode
                                                  [openroadm-operational-mode-id]
                    … roadm parameters as they appear in OM-catalog (line 35-51)
     |  |     |  +---w Add
     |  |     |  |  +---w add-openroadm-operational-mode*
                                                  [openroadm-operational-mode-id]
                    … roadm parameters as they appear in OM-catalog (line 54-66)
     |  |     |  +---w Drop
     |  |     |     +---w openroadm-operational-mode*
                                                  [openroadm-operational-mode-id]
                    … roadm parameters as they appear in OM-catalog (line 69-81)
     |  |     +---w amplifiers
     |  |        +---w Amplifier
     |  |           +---w openroadm-operational-mode*
                                                  [openroadm-operational-mode-id]
                    … amplifier parameters as they appear in OM-catalog (line 85-97)
     |  +--ro output
     |     +--ro configuration-response-common
```

```
|        +--ro request-id              string
|        +--ro response-code           string
|        +--ro response-message?       string
|        +--ro ack-final-indicator     string
```

5.23 add-specific-operational-modes-to-catalog RPC

The add-specific-operational-modes-to-catalog RPC is used to fill the second part of the catalog dedicated to the description of the modes associated with Bookended and Alien-Wavelength use cases. This dedicated rpc is used to fill the catalog each time a new transponder specific-operational-mode needs to be declared. The rules defined for parameters communication by an OEM to an operator is out of the scope of the OpenROADM MSA. The Body of the rpc providing information on a specific-operational-mode can NOT be publicly released with the models. It is the operator responsibility to define its own xml or json file to be imported as a collection of a RESTCONF client application.

```
+---x add-specific-operational-modes-to-catalog
       +---w input
       |  +---w sdnc-request-header
       |  |  +---w request-id?       string
       |  |  +---w rpc-action?       rpc-actions
                                     (fill-catalog-with-specific-operational-modes)
       |  |  +---w notification-url?    string
       |  |  +---w request-system-id?   string
       |  +---w operational-mode-info
       |     +---w specific-operational-modes
       |        +---w specific-operational-mode* [operational-mode-id]
       |              …
       | … grid & xponder parameters as they appear in OM-catalog (line 99-128)
       |              …
       |
       +--ro output
          +--ro configuration-response-common
             +--ro request-id              string
             +--ro response-code           string
             +--ro response-message?       string
             +--ro ack-final-indicator     string
```

5.24 end-terminal-performance-info-request RPC

The end-terminal-performance-info-request RPC is used to retrieve information on the alien end-terminal performances. This RPC is exercised by a Higher Layer Controller (HLC) or an OSS towards the End-Terminal Controller (ETC). The output of the RPC includes a list of supported operational modes, the min and max frequencies supported by the end terminal and the minimum granularity that can be applied to set the channel central frequency. The information will be used at a later step by the HLC/OSS to provide end-terminal characteristics to the RNC in the temp-service-create RPC.

```
+---x end-terminal-performance-info-request
|  +---w input
|  |  +---w sdnc-request-header
|  |  |  +---w request-id?          string
|  |  |  +---w rpc-action?          rpc-actions
|  |  |  +---w notification-url?    string
|  |  |  +---w request-system-id?   string
|  |  +---w service-a-end
|  |  |  +---w service-endpoint-details
|  |  |     +---w clli            string
```

```
|  |  |     +---w node-id?        org-openroadm-common-node-types:node-id-type
|  |  |     +---w tx-direction
|  |  |     |  +---w port
|  |  |     |     +---w port-device-name?         string
|  |  |     |     +---w port-circuit-pack-name?   string
|  |  |     |     +---w port-circuit-pack-type?   string
|  |  |     |     +---w port-type?                string
|  |  |     |     +---w port-name?                string
|  |  |     |     +---w port-rack?                string
|  |  |     |     +---w port-shelf?               string
|  |  |     |     +---w port-slot?                string
|  |  |     |     +---w port-sub-slot?            string
|  |  |     +---w rx-direction
|  |  |        +---w port
|  |  |       …  same parameters as the ones used for tx-direction
|  |  +---w service-z-end
|  |      …  same parameters as the ones describing service-a-end
|  +--ro output
|     +--ro configuration-response-common
|     |  +--ro request-id             string
|     |  +--ro response-code          string
|     |  +--ro response-message?      string
|     |  +--ro ack-final-indicator    string
|     +--ro a-z-end-common-interface-characteristics
|        +--ro supported-operational-modes* [preference]
|        |  +--ro preference             int16
|        |  +--ro operational-mode-id?   string
|        +--ro min-frequency?   org-openroadm-common-optical-channel-
                                types:frequency-THz
|        +--ro max-frequency?   org-openroadm-common-optical-channel-
                                types:frequency-THz
|        +--ro min-granularity? org-openroadm-common-optical-channel-
                                types:frequency-GHz
```

5.25 end-terminal-activation-request RPC

The end-terminal-activation-request RPC is used to activate alien end-terminals on both A and Z end nodes. This RPC is exercised by a HLC or an OSS towards the End-Terminal Controller (ETC). Before the RNC can configure connections in ROADMs, the end-terminal needs to transmit power at a compatible level, a relevant wavelength and according to a specific operational mode. This is needed by the RNC to configure connections on ROADMs where control loops rely on the presence of an input power. All these settings have been previously calculated by the RNC and provided to the HLC/OSS through a service-rpc-result notification.

The output power is adjusted through a loop where this RPC is used 2 times in combination with the end-terminal-power-control RPC and the end-terminal activation-status notification. This process allows compensating loss that could deviate from initial assumption on the fiber connecting the TX of the end terminal to the PP port of an SRG. However, the loss on the fiber shall not be excessive, otherwise the process will terminate, the HLC/OSS exercising an end-terminal-deactivation-request.

```
+---x end-terminal-activation-request
|  +---w input
|  |  +---w end-terminal-controller-service-name?   string
|  |  +---w service-name?                           string
|  |  +---w common-id?                              string
```

```
|  |  +---w sdnc-request-header
|  |  |  +---w request-id?          string
|  |  |  +---w rpc-action?          rpc-actions
|  |  |  +---w notification-url?    string
|  |  |  +---w request-system-id?   string
|  |  +---w service-a-end
|  |  |  +---w service-endpoint-details
|  |  |  |  +---w clli            string
|  |  |  |  +---w node-id?        org-openroadm-common-node-types:node-id-type
|  |  |  |  +---w tx-direction
|  |  |  |  |  +---w port
|  |  |  |  |     +---w port-device-name?         string
|  |  |  |  |     +---w port-circuit-pack-name?   string
|  |  |  |  |     +---w port-circuit-pack-type?   string
|  |  |  |  |     +---w port-type?                string
|  |  |  |  |     +---w port-name?                string
|  |  |  |  |     +---w port-rack?                string
|  |  |  |  |     +---w port-shelf?               string
|  |  |  |  |     +---w port-slot?                string
|  |  |  |  |     +---w port-sub-slot?            string
|  |  |  |  +---w rx-direction
|  |  |  |     +---w port
|  |  |  |        …  same parameters as the ones used for tx-direction
|  |  |  +---w min-output-power?    org-openroadm-common-link-types:power-dBm
|  |  |  +---w max-output-power?    org-openroadm-common-link-types:power-dBm
|  |  +---w service-z-end
|  |  |  +---w service-endpoint-details
|  |  |  |     …  same parameters as the ones used for tx-direction
|  |  |  +---w min-output-power?    org-openroadm-common-link-types:power-dBm
|  |  |  +---w max-output-power?    org-openroadm-common-link-types:power-dBm
|  |  +---w frequency? org-openroadm-common-optical-channel-types:frequency-THz
|  |  +---w width?     org-openroadm-common-optical-channel-types:frequency-GHz
|  |  +---w optical-operational-mode?              string
|  +--ro output
|     +--ro configuration-response-common
|        +--ro request-id             string
|        +--ro response-code          string
|        +--ro response-message?      string
|        +--ro ack-final-indicator    string
```

5.26 end-terminal-deactivation-request RPC

The end-terminal-deactivation-request RPC is used to deactivate alien end-terminals on both A and Z end nodes. This RPC is exercised by a HLC or an OSS towards the End-Terminal Controller (ETC) when :

- The end-terminal power can not be adjusted to a reasonable level because the fiber connecting the TX of the end terminal to the PP port of an SRG presents an excessive loss
- An alien wavelength service is to be deleted; prior the optical tunnel is deleted by the RNC.

In both cases, it will be followed by an end-terminal-activation-status notification that confirms whether the end-terminal has been correctly deactivated or not.

```
+---x end-terminal-deactivation-request
|  +---w input
|  |  +---w end-terminal-controller-service-name?   string
```

```
|  |  +---w due-date?                yang:date-and-time
|  |  +---w sdnc-request-header
|  |     +---w request-id?           string
|  |     +---w rpc-action?           rpc-actions
|  |     +---w notification-url?     string
|  |     +---w request-system-id?    string
|  +--ro output
|     +--ro configuration-response-common
|        +--ro request-id              string
|        +--ro response-code           string
|        +--ro response-message?       string
|        +--ro ack-final-indicator     string
```

5.27 end-terminal-power-control RPC

The end-terminal-power-control RPC is used in the process of adjusting end-terminals' power on both A and Z end nodes. This RPC is exercised by a HLC or an OSS towards the RNC.

The output power is adjusted through a loop where this RPC is used after the end-terminal activation-status notification provides the OSS/HLC information about the power-setpoints on end-terminals. The OSS/HLC reuses these last to provide it to the RNC feeding the input of the end-terminal-power-control RPC.

The output of the RPC allows the RNC to return the value of the correction that shall be applied to the TX of the end-terminals if a correction is needed (output-power-adjustment-needed = TRUE). If no adjustment is needed, output-power-adjustment-needed will be set to FALSE by the RNC. If the adjustment needed to compensate for excessive loss on the fiber connecting the end-terminal TX to the SRG PP is out of a range that could comply with end-terminal specification, srg-input-power-out-of-range is set to TRUE by the RNC.

```
+---x end-terminal-power-control
|  +---w input
|  |  +---w end-terminal-controller-service-name?   string
|  |  +---w service-name?                           string
|  |  +---w common-id?                              string
|  |  +---w sdnc-request-header
|  |  |  +---w request-id?          string
|  |  |  +---w rpc-action?          rpc-actions
|  |  |  +---w notification-url?    string
|  |  |  +---w request-system-id?   string
|  |  +---w frequency? org-openroadm-common-optical-channel-types:frequency-THz
|  |  +---w width?     org-openroadm-common-optical-channel-types:frequency-GHz
|  |  +---w optical-operational-mode?               string
|  |  +---w output-power-setpoint-a-end? org-openroadm-common-link-types:power-
                                                                        dBm
|  |  +---w output-power-setpoint-z-end? org-openroadm-common-link-types:power-
                                                                        dBm
|  +--ro output
|     +--ro output-power-adjustment-needed?   boolean
|     +--ro corrected-output-power-a-end?     org-openroadm-common-link-
                                                  types:power-dBm
|     +--ro corrected-output-power-z-end?     org-openroadm-common-link-
                                                  types:power-dBm
|     +--ro srg-input-power-out-of-range?     boolean
|     +--ro configuration-response-common
|        +--ro request-id              string
|        +--ro response-code           string
|        +--ro response-message?       string
|        +--ro ack-final-indicator     string
```

6 NOTIFICATIONS

The Service Model specifies notifications. The service providers' SDN Controllers can be notified by the ROADM Network Controller of the results of a RPCs action , as well as the results of some changes in the service Data Store that may not result directly from a request exercised by service providers' SDN Controllers.

There are 6 notifications defined in the Open ROADM Service Model:

Notification name	Release of introduction	Notification type
Service rpc result	2.2	org-openroadm-common-service-types/service-notification-types
	2.2	1 service-creation-result
	2.2	2 service-reconfigure-result
	2.2	3 service-delete-result
	2.2	4 service-roll-result
	2.2	5 service-revert-result
	2.2	6 service-reroute-result
	2.2	7 service-restoration-result
	7.0	8 successful-path-computation-resource-reserved
	7.0	9 path-computation-failed
	11.1	10 end-terminal-activation-status
Service traffic flow	2.2	No type included in the notification body
Service notification	2.2	org-openroadm-resource-types/resource-notification-types
	2.2	1 resourceCreation
	2.2	2 resourceModification
	2.2	3 resourceDeletion
Non rpc related notification	7.0	org-openroadm-controller-customization/non-rpc-related-notification/type
	7.0	1 service-state-change
	7.0	2 topology-change
	7.0	3 exceeded-attenuation-crossing-warning
	7.0	4 insufficient-margin-crossing-alarm
	7.0	5 autonomous-optical-restoration-triggered
Optical-tunnel-created	10.0	No type included in the notification body
End terminal activation status	10.0	No type included in the notification body

6.1 service rpc result notification

The service rpc result notification indicates the result of a service RPC exercised by the Operator SDN controller.

Table 6-1 Service RPC result notification in the Open ROADM Service Model

	Notification Parameter		*Manda-tory*	*Description*
Service RPC Result			Yes	
1	Notification Type		No	There are 9 types of Service notification. – Service create result, Enum=1 – Service reconfigure result, Enum=2 – Service delete result, Enum=3 – Service roll result, Enum=4 – Service revert result, Enum=5 – Service reroute result, Enum=6 – Service restoration result, Enum=7 – successful-path-computation-resource-reserved, Enum=8 – path-computation-failed, Enum=9
2	Status		Yes	2 types: “Successful”, Enum=1; “Failed”, Enum=2
3	Status Message		No	Details about the status (string)
4	Service Name		No	Identifier for the service being reported on, e.g., CLFI etc. (string)
5	version-number		No	Versioned service number (uint64)
6	Common id		No	Common id used to identify temporary services. (string)
7	Actual Date		No	Actual date and time (if successful) yang:*date-*and-time
8	Path computation result	*A to Z*		Results of the path computation based on settings that shall be used by the pluggable/xponders to achieve estimated performances.
9		*frequency*	No	Frequency that the external pluggable/ xponder shall be tuned to. (org-openroadm-common-optical-channel-types:frequency-THz)

Notification Parameter				*Manda-tory*	*Description*
10			*width*	No	Width that shall not be exceeded. (org-openroadm-common-optical-channel-types:frequency-GHz)
11			*optical-operational-mode*	No	Operational mode that the external pluggable/xponder shall be tuned to (string)
12			*rx-estimated-osnr*	No	Estimated osnr at the receiver side (org-openroadm-common-link-types:ratio-dB)
13			*rx-estimated-gsnr*	No	Estimated gsnr at the receiver side (org-openroadm-common-link-types:ratio-dB)
14			*max-output-power*	No	Maximum output power (org-openroadm - common-link-types:ratio-dB)
15			*min-output-power*	No	Minimum output power (org-openroadm - common-link-types:ratio-dB)
16		*Z to A*			For Z to A direction, repeat parameters from line 8 to line 16.

6.2 service traffic flow notification

The service traffic flow notification indicates that traffic is flowing again on the service after an administrative action has been completed.

Table 6-2 Service Traffic Flow Notification in the Open ROADM Service Model

Notification Parameter		*Manda-tory*	*Description*
Service Traffic Flow			
1	Services Name	Yes	– Service identifier. Unique within the context of a network, e.g., CLFI etc. (string)
2	Actual Date	No	*Actual date* and time traffic started flowing, yang:*date*-and-time

6.3 service notification

The service notification indicates that a service has been added, modified, or removed. A resource creation notification shall contain the created service in its entirety. A resource modified notification shall contain just the modified field, plus the service identifier. A resource deleted notification shall just contain the service identifier.

Table 6-3 Service Notification in the Open ROADM Service Model

	Notification Parameter		*Manda-tory*	*Description*
Service Notification				
1	Notification Type		No	Refer to line 1 of Service RPC Result
2	Service Name		Yes	Service identifier. Unique within the context of a network, e.g., CLFI etc. (string)
3	Version-number		No	Service version (uint64)
4	Common ID		No	Service order #, or identifier to be used by the ROADM controller to identify routing constraints received from planning applications. (string)
5	SDNC Request Header	*Request ID*	No	From original system requesting for the service. Uniquely generated by calling system. (string)
6		*RPC Action*	No	20 types[16]: – Service create, Enum=1 – Service feasibility check, Enum=2 – Service delete, Enum=3 – Equipment notification, Enum=4 – Temp service create, Enum=5 – Temp service delete, Enum=6 – Service roll, Enum=7 – Service reconfigure, Enum=8 – Service restoration, Enum=9 – Service reversion, Enum=10 – Service reroute, Enum=11 – Service reroute confirm, Enum=12 – Network re-optimization, Enum=13 – Service feasibility check bulk, Enum=14 – Ber test, Enum=15 – Controller parameterg setting, Enum=16 – Optical tunnel create, Enum=17 – Optical tunnel request cancel, Enum=18

[16] Not all RPCs generate service notifications.

Notification Parameter			*Manda-tory*	*Description*
				– fill catalog with or operational modes, Enum=19 – fill catalog with specific operational modes, Enum=20
7		*Notification url*	No	URL for asynchronous response (string)
8		Request System ID	No	Identifier of application initiates the request (string)
9	Connection Type		Yes	4 types: Service, Enum=1; Infrastructure, Enum=2; ROADM line, Enum=3; optical-tunnel, Enum=4;
10	Lifecycle State		No	Service lifecycle state, 8 types (string) Deployed, Enum=1; Planned, Enum=2; Maintenance, Enum=3; Deploying, Enum=4; Undeploying, Enum=5; Undeployed, Enum=6; Proposed, Enum=7; Draft, Enum=8
11	Administrative State		No	Intended state of service (string)
12	Operational State		No	Actual state of service (string)
13	Condition		No	Service condition. Additional information about the state of the service. Only sent when applicable. 5 types: Restored temporarily, Enum=1; Re-routed temporarily, Enum=2; Activated for service, Enum=3; Activated for further check, Enum=4; Activated for troubleshooting failure, Enum=5
14	Service A-end	*Service Format*	Yes	7 types: Ethernet, Enum=1; OTU, Enum=2; OC, Enum=3; STM, Enum=4; OMS, Enum=5; ODU, Enum=6; OTM, Enum=7;
15		*Service rate*	No	E.g., 10G, 100G etc. rate in integer (uint32)
16		*OTU service rate*	No	Full rate of transport of OTUn, e.g., OTU2, OTU4
17		*ODU service rate*	No	Sub-rate ODU services, e.g., ODU0 in an OTU4 interface.

Notification Parameter					Manda-tory	Description
18		*Ethernet Encoding*			No	Type of Ethernet encoding when the rate = 10GE. 2 types: "10GBASE-W", Enum=1; and "10GBASE-R", Enum=2
19		*Mapping Mode*			No	Applies only to 10GE. "GFP-F" maps into an OPU2 with PT=5 (ITU-T G.7041 Section 7.1) "GFP-E" maps into an OPU2 with PT=9 (ITU-T G.7041 Section 7.9). Note GFP-E is an Open ROADM term to mean "Extended" OPU2 mapping "PCS-Transparent" maps into an OPU2E with PT=3 (ITU-T G.709 Section 17.2)
20		*CLLI*			Yes	Office location, Note the CLLI must match the site associated with the device-id of this endpoint (string)
21		*Node ID*			No	Globally unique identifier for a device length "7..63" pattern "([a-zA-Z][a-zA-Z0-9-]{5,18}[a-zA-Z0-9])" A Node ID can contain letters, numbers, and hyphens. The first character must be a letter. The last character must be a letter or number.
22		*Tx direction*	*Port*			Uses service port, service LGX, and service tail. From the device model perspective the port-device-name plus the port-circuit-pack-name plus the port-name uniquely identifies the port. From the network model perspective the openroadm-topology-ref plus port-device-name plus port-name uniquely identify the termination point in the network model.
23				*Port device name*	No	Port defined for the end-to-end service (string)
24				*Port circuit pack name*	No	Port circuit pack name for the service (string)
25				*Port type*	No	Port type, e.g. "router" or "POI" etc. (string)
26				*Port name*	No	E.g. Tx, Rx (string)

Notification Parameter						***Manda-tory***	***Description***
27				*Port rack*		No	E.g. Bay FIC: Frame Identification Code (string)
28				*Port shelf*		No	E.g. shelf in the bay (string)
29				*Port slot*		No	E.g. slot in the shelf (string)
30				*Port sub-slot*		No	E.g. sub-slot in the shelf or on a card (string)
31			*LGX*	*LGX device name*		No	E.g. name/identifier of the LGX (string)
32				*LGX port name*		No	E.g. port name of the LGX (string)
33				*LGX port rack*		No	E.g. rack port of the LGX (string)
34				*LGX port shelf*		No	E.g. shelf port of the LGX (string)
35			*Tail*	*Tail ROADM*	*Node ID*	No	Tail ROADM: ROADM on which the Xponder is connected to (TID, IP Address, or FQDN). Node ID: Refer to line 21.
36				*Xponder Port*	*Circuit pack name*	No	Tail Xponder circuit pack name/identifier (string)
37					*Port name*	No	Xponder circuit pack port name (string)
38				*Tail ROADM AID*		No	Provide Xponder's port for intercity ROADM connection (bay, shelf, slot, and port)
39				*Tail ROADM Port Rack Location*		No	Xponder's location, e.g., FIC (Frame Identification Code) of the tail ROADM
40		*Rx direction*	For Rx direction, repeat parameters from line 22 to line 40.				
41		*Optics type*				No	2 types: Gray, Enum=1; DWDM, Enum=2
42		*Router*		*Node ID*		No	Refer to line 21.
43				*IP Address*		No	Router IP address, inet:*ip-address*
44				*URL*		No	Router URL (string)
45		*User Label*				No	Label for service endpoint, defined by the user (string)

Notification Parameter						***Manda-tory***	***Description***
46	Service Z-end	Repeat parameters from line 20 to line 51 for Service Z-end					
47	Hard Cons-traints	*Customer Code*				No	For selecting tagged equipment on which to route a service. If more than one customer code is provided, they will be treated as an ordered list. (string)
48		*General*	*Diversity*	*Existing service*		No	Diverse from existing services identified by facility CLFI, list. (string) Constraints are either general or co-routing. Under general constraints, there are diversity, exclude, include and latency constraints.
49				*Existing service applicability*	*Site*	No	Site identifies the CLLI (Boolean)
50					*Node*	No	Refer to line 21 (Boolean)
51					*SRLG*	No	Shared Risk Link Group data, (Boolean)
52			*Exclude*	*Fiber bundle*		No	Fiber segment usually defined by SRLG (string), list.
53				*Site*		No	Site identifies the CLLI (Boolean), list.
54				*Node*		No	Refer to line 21, list.
55				*Supporting service name*		No	Supporting service(s) to exclude from this route (string), list. Supporting service is the service name that another service runs over top. For example, if connection-type is service, then this is the related connection-type = infrastructure service.
56			*Include*	*Fiber bundle*		No	Refer to line 52 for include.
57				*Site*		No	Refer to line 53 for include.
58				*Node*		No	Refer to line 54 for include.
59				*Supporting service name*		No	Refer to line 55 for include.
60			*Latency*	*Maximum Latency*		No	Maximum *latency* allowed (uint32), units in “ms”.

Notification Parameter				***Manda-tory***	***Description***
61		*Co-routing*	*Existing service*	No	The existing service that is to be co-routed, list.
62	Soft Cons-traints	*Repeat parameters from line 53 to line 67 for soft constraints*		No	
63	Due date			No	Date and time service to be turn up. If time is not specified for a given date, default to midnight. Service turned up immediately if no *due date* is specified. Type: yang:*date*-and-time
64	End Date			No	Date and time service to be removed. Type: yang:*date*-and-time
65	Event Horizon Start			No	Start time to ensure that the service is routable and viable. Required resources shall be considered reserved from this time. If not provided, defaults to due date. Type: yang:*date*-and-time
66	Event Horizon End			No	End time to ensure that the service is routable and viable. Required resources shall be considered reserved until this time. If not provided, defaults to end-date. Type: yang:*date*-and-time
67	NC code			No	Network Channel code applied to wavelength service only. This is reported against the service but may not get reflected in the service in the network (string).
68	NCI code			No	Network Channel Interface code applied to wavelength service only. This is reported against the service but may not get reflected in the service in the network (string).
69	Secondary NCI code			No	Secondary NCI code applied to wavelength service only. This is reported against the service but may not get reflected in the service in the network (string).
70	Customer			No	To be included in ticket information. This is reported against the service but may not get reflected in the service in the network (string).
71	Customer contact			No	Customer contact information to be included in ticket information. This is reported against the

Notification Parameter				*Manda-tory*	*Description*
					service but may not get reflected in the service in the network (string).
72	Operator contact			No	Operator contact information to be included in ticket information. This is reported against the service but may not get reflected in the service in the network (string).
73	Service layer			No	Layer of the service. 2 types: WDM, Enum=1; OTN, Enum=2
74	Latency			No	Service Latency in integer (uint32), units in "ms"
75	Fiber Span SRLGs			Yes	List of shared risk link group data on fiber spans, shared risk link group identifiers (string).
76	Equip-ment SRGs	*SRG number*		Yes	List of shared risk link group data on equipment (string).
77	Supporting Service Name			Yes	Supporting service is the service name that another service runs over top. For example, if connection-type is service, then this is the related connection-type = infrastructure service, list.
78	Topo-logy	aToZ	*ID*	Yes	aToZ list. Unique identifier and used as key for this network-topology component within this service (string) Topology reports the individual hops along the service in the A to Z direction and Z to A directions. This includes both ports internal to a device and those at its edge that are available for externally connections. It includes both physical and logical ports. Physical ports are ordered with the logical ports that run over them as follows: a.\t On ingress to a node/card, physical then logical b.\t On egress to a node/card, logical then physical
79			*Hop Type*	No	2 types: Node external, Enum=1, the given resource is on the edge of the node and used in relationships to resources outside of the node. Node internal, Enum=2, the given resource is internally to the node.

Notification Parameter					Manda-tory	Description
80			*Device*	*Node ID*	No	Refer to line 21.
81			*Resource*		No	This resource identifier is intended to provide a generic identifier for any resource that can be used without specific knowledge of the resource. If selected, only one of the parameters in line 88 to line 106 will be chosen.
82			*Circuit Pack*	*Circuit Pack Name*	Yes, in case selected	Circuit pack, Enum=8 Circuit pack name is the circuit pack identifier. Unique within the context of a device. Same as leafref value in model, if applicable. (string)
83			*Port*	*Circuit Pack Name*	Yes, in case selected	Port, Enum=7 Circuit pack name, see line above.
84				*Port Name*	No	Port, Enum=7 Port name is the port identifier. Unique within the context of a device. Same as leafref value in model, if applicable. (string)
85			*Connection*	*Connection Name*	Yes, in case selected	Connection, Enum=5 This is used by either ROADM connection or ODU connection since they are mutually exclusive in the model. Connection name is unique within the context of a device. Same as leafref value in model, if applicable. (string)
86			*Physical Link*	*Physical Link Name*	Yes, in case selected	Physical link, Enum=10 Physical link name is the physical link identifier. Unique within the context of a device. Same as leafref value in model, if applicable. (string)
87			*Internal Link*	*Internal Link Name*	Yes, in case selected	Internal link, Enum=9 Internal link name is the internal link identifier. Unique within the context of a device. Same as leafref value in model, if applicable. (string)
88			*Shelf*	*Shelf Name*	Yes, in case selected	Shelf, Enum=12 Shelf name is the shelf ID identifier. Unique within the context of a device. Same as leafref value in model, if applicable. (string)
89			*SRG*	*SRG Number*	Yes, in case selected	Shared Risk Group, Enum=4

Notification Parameter					*Manda-tory*	*Description*
						SRG number is the shared risk group identifier. Unique within the context of a device. Same as leafref value in model, if applicable. (uint16)
90			*Degree*	*Degree Number*	Yes, in case selected	Degree, Enum=3 Degree number is the degree identifier. Unique within the context of a device. Same as leafref value in model, if applicable. (uint16)
91			*Service*	*Service Name*	Yes, in case selected	Service, Enum=13 Service name is the service identifier. Unique within the context of a network. Same as leafref value in model, if applicable. (string)
92			*Interface*	*Interface Name*	Yes, in case selected	Interface, Enum=11 Interface name is the interface identifier. (string)
93			*ODU sncp pg*	*ODU sncp pg name*	Yes, in case selected	ODU sncp pg, Enum=14 ODU sncp pg name is the name of the ODU sncp pg. (string)
94			*Other*	*other-resource-id string*	Yes, in case selected	Other, Enum=1 Resource of type not found in list Resource ID for other (string)
95			*Device*	*Node ID*	Yes, in case selected	Device, Enum=2 ROADM, Xponder, etc., Node ID is a globally unique identifier for a device. Same as leafref value in model, if applicable.
96			*Line amplifier*	*Amp Number*	Yes, in case selected	Line amplifier, Enum=15 Amp number is the number of the line amplifier. (uint8)
97			*Xponder*	*Xpdr Number*	Yes, in case selected	Xponder, Enum=16 Xpdr number is the number of the Xponder. (uint16)
98			*Versioned Service*	*Versioned Service Name*	Yes, in case selected	Versioned service, Enum=17 Versioned service name is the versioned service identifier. Unique within the context of a network. Same as leafref value in model, if applicable. (string)

Notification Parameter					*Manda-tory*	*Description*
99				*Version Number*	Yes, in case selected	Versioned service, Enum=17 Version number of the service (uint64)
100			*Temp Service*	*Common ID*	Yes, in case selected	Temp service, Enum=18 Common ID is the temp service identifier. Unique within the context of a network. Same as leafref value in model, if applicable. (string)
101		*Resource Type*	*Type*		Yes	Resource type, refer to line 88 to line 106 for Enum value.
102			*Extension*		No	Populated when resource type not defined or when Enum value is set to 'other' (string)
103		zToA Repeat parameters from line 84 to line 108 for zToA.				zToA list.
104	Is Bandwidth Locked				No	Boolean (true or false), default is "false". Bandwidth lock indicates whether the service is administratively prohibited from taking on more capacity, i.e., whether it can be used as a supporting service in any new service creations. Unlike administrative status, this does not impact any previous planned or deployed services.
Optical tunnel created						
105	Actual Date				No	*Actual date* and time traffic started flowing, yang:*date*-and-time
106	Repeat parameters from line 3 to line 105 in Table 2-1. Service List					

6.4 Non rpc related notification

The non-rpc-related notification is added in R7.0. It is used to provide information that may not result directly from a request exercised by a higher-level controller

Several notification types have been defined for this purpose

- service-state-change: when the state of service changed, the service name is also communicated,

- topology-change: when a change is observed in the topology. The notification will then include information about the topology-layer, and the impacted resources (type and id),
- exceeded-attenuation-crossing-warning, when attenuation on a link raised a value that exceeds the predefined threshold, and is out of the range for which the WDM line was engineered,
- insufficient-margin-crossing-alarm: when at initial commissioning, the measured attenuation is not in line with engineered-spanloss,
- autonomous-optical-restoration-triggered: when a service rerouting was triggered autonomously by the RNC.

```
+---n non-rpc-related-notification
   +--ro notificationType?   org-openroadm-controller-customization:non-rpc-
                                related-notification-type
   +--ro notificationId?        string
   +--ro notification-time?     yang:date-and-time
   +--ro service-name?          string
   +--ro impacted-resource-type? string
   +--ro impacted-resource-id?  string
   +--ro topology-layer?        enumeration
```

6.5 Optical tunnel created notification

The optical-tunnel-created notification is sent to the higher layer controller by the RNC, at the end of the process to create an optical tunnel in the scope of alien wavelength service and IPoWDM use cases ; to confirm that the creation of the tunnel was successful.

The structure of the notification is the same as the one used in service-notification presented in the section 6.3 with the addition of the leaf actual-date.

```
+---n optical-tunnel-created
|  +--ro notificationType?      org-openroadm-resource-types:
|                               resource-notification-type
|  +--ro version-number?             uint64
|  +--ro actual-date?                yang:date-and-time
|   …              All Attributes related to the service
          (Replicates the structure used in the service notification)
```

6.6 End terminal activation status notification

The end terminal activation status notification provides the status of the end-terminal after an activation or deactivation is requested. It is sent by the End Terminal Controller (ETC) to the higher layer controller. Activation status can be either activated (1) or deactivated (2). The output power setpoint gives a reference that is indirectly provided to the RNC via the higher layer controller. Thus the RNC can evaluate from the set-point and the received power in SRG-PP the attenuation of the fiber between the end-terminal TX and the SRG-PP-RX.

```
+---n end-terminal-activation-status
|  +--ro end-terminal-controller-service-name?   string
|  +--ro service-name?                           string
|  +--ro common-id?                              string
|  +--ro status                                  rpc-status
|  +--ro status-message?                         string
|  +--ro activation-status?     org-openroadm-common-service-types:end-
                                terminal-activation-status-type
|  +--ro actual-date?           yang:date-and-time
|  +--ro output-power-setpoint-a-end? org-openroadm-common-link-types:power-dBm
```

```
|  +--ro output-power-setpoint-z-end? org-openroadm-common-link-types:power-dBm
```

7 SERVICE MODEL WHITE PAPER HISTORY

7.1 Version 2.2 (August 31, 2018)

The initial version of the Service Model White Paper reflected the MSA v2.2 model and explained the structure of the service model, the service model RPCs, and the service model notifications.

The MSA v2.2 service model data store covered the following:

- ***Service List*** – the list of services requested or created in the ROADM Network Controller (RNC). The Service List contains a single service per service-name.
- ***Versioned Service List*** – the Versioned Service List adds a version number to the Service List, which allows for tracking of multiple versions of a Service. Services in this list are uniquely identified by service-name and version-number.
- ***Temp Service List*** – the Temp Service List contains a list of service reservations that may be provisioned in the future. Once provisioned, a Temp Service becomes part of the Service List and is removed from the Temp Service List. Services in this list are uniquely identified by common-id.

The v2.2 Service Model RPCs are summarized in the table below. A brief description of each RPC is included because the scope of some of the RPCs changed over time.

Table 5-1 Service Model MSA v2.2 Remote Procedure Calls

RPC Name	***Purpose***	***Enum Value***
Service Create	Request the RNC to create a new service	1
Service Feasibility Check	Request the RNC to verify the feasibility of a single service request	2
Service feasibility Check Bulk	Request the RNC to verify the feasibility of a set of service requests	14
Service Delete	Request the RNC to remove an existing service	3
Equipment Notification	Request the RNC to report the discovery of new network equipment to a higher layer system or controller	4
Temp Service Create	Request the RNC to compute a service path and reserve the corresponding wavelength (optical spectrum) for a future service request	5
Temp Service Delete	Request the RNC to release a reserved wavelength (optical spectrum)	6
Service Roll	Request the RNC to change the path of a service while keeping the same A and Z end points	7
Service Reconfigure	Request the RNC to reconfigure a service to use different terminating equipment (re-home) or to re-route the service using different routing constraints	8

Service Restoration	Request the RNC to temporarily or permanently restore a service disrupted by regen failures	9
Service Reversion	Request the RNC to revert a temporarily restored or re-routed service to the original equipment or service path	10
Service Reroute	Request the RNC to determine whether it is possible to temporarily restore a service that has been affected by ROADM network failures	11
Service Reroute Confirm	Request the RNC to temporarily restore a service that has been affected by ROADM network failures	12
Network Re-optimization	Request the RNC to determine whether it is possible to optimize the route of an existing service without violating any of the routing constraints	13

The v2.2 Service Model defined three notifications to provide results of service requests. Note that each of the Service Model RPCs includes an output container intended to provide the RPC result to the RPC requestor (results may be reported asynchronously), so the Service Model notifications are an additional mechanism for providing service updates. The three notifications included in the v2.2 service model are:

- Service RPC Result – a short notification to indicate the result (success/failure) of any service RPC
- Service Traffic Flow – a short notification to provide the date and time traffic began flowing after an administrative action
- Service Notification – a notification to indicate that a service was added, modified, or removed. Depending on the action, this notification may contain all service details.

7.2 Version 3.1.1 and 4.1.0 (September 23, 2019)

An update to the v2.2 baseline service model was developed in late 2019 to cover Service Model updates included in MSA versions 3.1.1 and 4.1.0. These updates were documented as an addendum to the v2.2 Service Model White Paper and although it was reviewed by the forum, the addendum was not officially released.

Key updates in versions 3.1.1 and 4.1.0 included:

- The introduction of new RPCs to better handle service creation, including BER test and callback RPCs which allow the ROADM network controller to report information on the result of the operations associated with service creation/deletion and BER tests to the service provider's SDN controller
- The addition of several PCE routing metrics
- The introduction of resiliency type to model different Service Level Agreements (SLA)
- The introduction of the notion of coupled-services to link services that are bound together and not handled separately
- The addition of a backup topology including a list of backup-paths
- The integration of new functions for existing RPCs
- The ability to report the service topology against Network model entities
- The ability to report the layer-specific link resources assigned to a service in the service topology
- The ability to specify subrate Ethernet services
- The support for bandwidth calendaring
- Updated the service model RP and notifications to be generic to support temp services and versioned services

The following new RPCs were added in MSA v3.1.1:

- BER Test RPC (section 5.15)
- Service RPC BER Test Async Callback RPC (section 5.16)
- Service Create Result Notification Request RPC (section 5.17)
- Service Delete Result Notification Request RPC (section 5.18)

The following modifications were made in the MSA v3.1.1 and 4.1.0 service models and RPCs:

- Routing Metrics
    - Added to service-list, versioned-service-list and temp-service-list data stores
    - Affected RPCs: service-create, service-feasibility-check, service-feasibility-check-bulk, temp-service-create, service-roll, service-reconfigure, service-restoration, service-reroute, and network-re-optimization
- Service Resiliency
    - Added to service-list, versioned-service-list and temp-service-list data stores
    - Affected RPCs: service-create, service-feasibility-check, service-feasibility-check-bulk, temp-service-create, service-reconfigure and service-reroute
- Bandwidth Calendaring
    - Added to service-list, versioned-service-list and temp-service-list data stores
    - Affected RPCs: service-create, service-feasibility-check, service-feasibility-check-bulk, temp-service-create, and service-reconfigure
- Link related parameters in Routing Constraints (added for diversity as well as include/exclude constraints)
    - Added to service-list, versioned-service-list and temp-service-list data stores
    - Affected RPCs: service-create, service-feasibility-check, service-feasibility-check-bulk, temp-service-create, service-reconfigure, service-reroute, and service-reroute-confirm
- Sub-rate Ethernet Service Level Agreement (SLA)
    - Added to service-list, versioned-service-list and temp-service-list data stores
    - Affected RPCs: service-create, service-feasibility-check, service-feasibility-check-bulk, temp-service-create, and service-reconfigure
- Backup Path ID and Failure Case ID
    - Added to service-list, versioned-service-list and temp-service-list data stores
    - Affected RPCs: service-restoration
- Leaf parameters to allow for reference to network model layers (clli-network-ref, openroadm-network-ref, openroadm-topology-ref)
    - Added to service-list, versioned-service-list and temp-service-list data stores
    - Affected RPCs: service-create, service-feasibility-check, service-feasibility-check-bulk, temp-service-create, and service-reconfigure
- Backup Topology to support service-resiliency; this container is similar to (service) Topology but may contain a list of backup topologies indexed by backup-path-id
    - Added to service-list, versioned-service-list and temp-service-list data stores
    - Affected RPCs: none
- Network Topology and Network Backup Topology, which provide the service topology and service backup topology using network model resources that may be used by the RNC path computation element.
    - Added to service-list, versioned-service-list and temp-service-list data stores
    - Affected RPCs: service-create, service-feasibility-check, service-feasibility-check-bulk, temp-service-create, and service-reconfigure

The following RPCs were redefined to improve consistency and to clarify their application:

- Service Restoration – the scope of this RPC was redefined to cover service disruption as a result of both regen and optical line failures. The latter was originally supported by the Service Reroute RPC; with this update, all service affecting failures of the ROADM network may be restored using the Service Restoration RPC.

- Service Reroute – the scope of this RPC was redefined to support temporary or permanent reroute requests

The following updates were made to service notifications in MSA versions 3.1.1 and 4.1.0, primarily to align with other changes to the service model:

- version-number
- service-name
- subrate-eth-sla container in service a-end and z-end
- parameters related to interface in topology and backup topology
- network-topology and network-backup-topology
- service-resiliency
- routing-metrics
- bandwidth-calendaring
- link and link-identifier list

In addition, a notification for ber-test rpc was added in version 3.1.1.

7.3 Version 7.0 (March 27, 2020)

Service Model version 7.0 defines the following service related notifications:

- non-rpc-related-notification
  - Used to provide information that may not result directly from a request exercised by a higher level controller

    Several notification types (org-openroadm-controller-customization:non-rpc-related-notification-type) have been defined for this purpose:

    • service-state-change : when the state of service changed, the service name is also communicated,

    • topology-change : when the change is observed in the topology. The notification will then include information about the topology-layer, and the impacted resources (type and id).

    • exceeded-attenuation-crossing-warning, when attenuation on a link raised a value that exceeds the predefined threshold, and is out of the range for which the WDM line was engineered,

    • insufficient-margin-crossing-alarm, when at initial commissioning, the measured attenuation is not in line with engineered-spanloss,

    • autonomous-optical-restoration-triggered, when a service rerouting was triggered autonomously by the RNC,

7.4 Version 10.0 (September 25, 2021)

Key updates in versions 10.0 included:

- (To be updated)
- The introduction of an operational mode catalog to translate the OpenROADM optical specifications, for W, MW-MW, MW-WR and MWi spreadsheets. This catalog also includes a section which allows translating specifications for transponders that may not fully comply with OpenROADM specifications and are used in the bookended mode.
- The introduction of some parameters in the model and specific RPCs to handle alien wavelength use case, where external transponders or pluggables are connected to the infrastructure handled by the RNC. These external terminals may not comply with OpenROADM APIs, and are controlled through an external controller such as an IP-SDNC. The IP-SDNC shall request the creation of an optical tunnel service using the set of RPCs defined in the service model.

The following new RPCs were added in MSA v10.0:

- Add openroadm operational modes to catalog (section 5.22)
    - Used to fill the catalog part which translates the OpenROADM specifications.
- Add specific operational modes to catalog (section 5.23)
    - Used to fill the catalog part which translates bookended Xponder/pluggable specifications.
- Optical tunnel create (section 5.20)
    - Used in the second step of an optical-tunnel service creation after a temp-service-create (connection-type= optical-tunnel) has been exercised if a path has been successfully calculated and the parameters to set the external pluggable/Xponders accepted by the external controller.
- Optical tunnel request cancel (section 5.21)
    - Used in the second step of an optical-tunnel service creation after a temp-service-create (connection-type= optical-tunnel) has been exercised if a path has not been successfully calculated or if the parameters to set the external pluggable/Xponders are rejected by the external controller. Allows to abort the optical-tunnel service creation

The following new notification was added in MSA v10.0:

- Optical tunnel created (section 6.5)
    - Used to confirm to the higher layer controller that the optical tunnel service has been successfully created

The following modifications were made in the MSA v10.0 service models and RPCs:

- Optical-tunnel added to connection-type enumeration
    - Used in service-list, versioned-service-list and temp-service-list data stores
    - Affected RPCs: temp-service-create, service-feasibility-check, service-feasibility-check-bulk, service-reconfigure
- Operational mode added to the constraints
    - Operational mode can be specified as either a soft or a hard constraints to be apply for path computation. In case end points host flexible Xponders, the choice of the operational mode to be used can this way be forced (hard constraint) or suggested (soft constraint).
- Requesting interface properties container added to service A and Z end container
    - This container is triggered in temp-service-create, service-feasibility-check, service-feasibility-check-bulk, and service-reconfigure, when the connection type is set to optical-tunnel. It contains information on the external pluggable/Xponder (supported-operational-modes, min and max frequency...) so that the controller, and more specifically the Path Computation Engine can get the knowledge of the performances of external devices.
- A path-computation-results grouping is used to keep trace of the results of the path computation (calculated osnr and gsnr) as well as the assumptions used by the PCE to reach calculated performances (TX-power). One shall note that the frequency, width and optical operational mode information is described by otsi/och resources;
    - Both are used in service/topology and backup-topology (container computation-results, and och/otsi resource describing the path)
    - Both are used in service-feasibility-check-outputs (container expected-settings-and-performances)
    - Both are used in service-rpc-result notification where a container path-computation-result is added when the service-notification-type = path-computation-result
- 2 rpc actions are added in org-openroadm-common-service-type
    - optical-tunnel-create
    - optical-tunnel-request-cancel
- 2 service-notification-types are added org-openroadm-common-service-type
    - successful-path-computation-result-resource-reserved
    - path-computation-failed

The following notification and RPCs were updated to align with other changes to the service model:

- Service rpc result : when the notification type is path-computation-result, a path-computation-result container is triggered and provides information on the result of the path computation (osnr, gsnr) as well as the settings (output power, operational mode, frequency…) that shall be applied by an external controller to pluggables to address the alien wavelength use case
- Temp-service-create, service-feasibility-check, service-feasibility-check-bulk, and service-reconfigure includes a container requesting-interface-properties container added to service A and Z end container when the connection-type =optical-tunnel. The operational-mode can also be provided as a constraint.

The following updates were made to service notifications in MSA versions 10.0, primarily to align with other changes to the service model:

- service-rpc-result includes a container path-computation-results when the notification-type = path-computation-results

7.5 Version 10.1 (December 10, 2021)

Key updates in versions 10.1 included:

- Yang model changes that simplify implementation in OpenDaylight framework, but has no other specific impact.

7.6 Version 11.0 (March 25, 2022)

Key updates in versions 11.0 included:

- (To be updated)

7.7 Version 11.1 (June 7, 2021)

Key updates in versions 11.1 included the addition of RPCs and notification to handle the alien wavelength and IPoWDM use cases. The added set of RPCs and notification allows retrieving parameters associated with the end terminals that can be either Xponders or external pluggable, and their activation/deactivation.

The following new RPCs were added in MSA v11.1:

- End terminal performance info request: (section 5.24)
  - used to retrieve from the End-Terminal Controller (ETC) information about end terminals that can be either Xponders or external pluggable. This information includes the supported operational modes and the frequency range of operation.
- End terminal power control: (section 5.27)
  - Used in the control loop dedicated to power adjustment of the end-terminal.
- End terminal activation request: (section 5.25)
  - used to activate the end-terminal so that incoming channel power allows triggering connection control in ROADMs.
- End terminal deactivation request: (section 5.26)
  - used to deactivate the end-terminal if the optical tunnel creation by the RNC failed, or when an alien wavelength service is terminated.

The following notification was added in MSA v11.1:

- End terminal activation status : (section 6.6)
  - used to provide a status on the activation/deactivation process of the end terminals, as well as information about their output-power set points (in the case of an activation).

7.8 Version 13.1 (March 19, 2024)

Updates to this version included an update to the service-list structure in Table 2-1, Table 2-2, and Table 2-3 to include the OTN structures and the refactored constraints sections. Similar updates were made to the service-create RPC and synchronous response sections (Table 5-1, Table 5-2) and service-feasibility-check request/response RPCs (Tables 5-3 through 5-6).

8 ALIEN WAVELENGTH USE CASE DESCRIPTION

External pluggable were introduced in the release 2 of the device model. However, the model suffered from limitations that did not allow a real implementation. In release 10.0, we complemented both the service and the device model to support external pluggable according to several options.

In the first option, we complement the device model to make it usable, considering OpenROADM compliant external pluggable that follows the device API. The pluggable is handled in this case as regular devices by a controller which can be the RNC or another controller, provided that it supports a Netconf/OpenROADM south bound interface (SBI). The proposed models have no impact on the service model and are described in the device white paper. Controlling the pluggable through the RNC, would lead to have the device that hosts the pluggable handled through 2 different SDN controller (SDNC) : the ROADM Network Controller (RNC) which controls the optical line side (Network port of the pluggable), and an another controller which controls the device that hosts the pluggable, and the pluggable outside any parameters that are directly associated with the network port (handled through the OpenROADM model). If the pluggable is hosted in a router, this would be the IP-SDNC. This alternative is associated to a number of limitations and leads to a complex implementation. This is the reason why a second option has been considered.

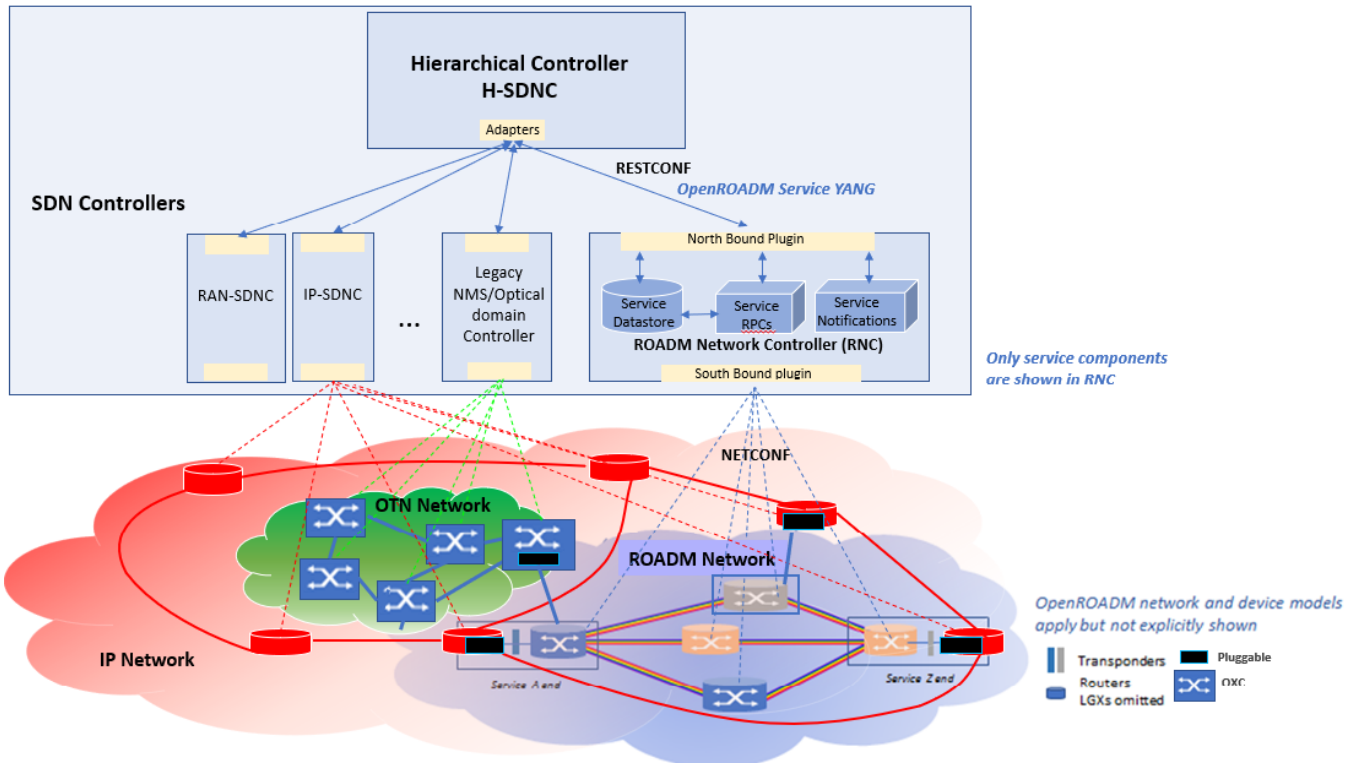
In the second option, we propose to handle external pluggable as an external device that does not follow the OpenROADM device model, as an alien transceiver. The alien wavelength use case has been considered in an agnostic way: it allows handling not only external pluggable but also any kind of external devices that may be connected to the ROADM infrastructure, considered as an Open Line System (OLS). The RNC is used to configure an optical tunnel between the PPs of 2 different SRGs. The implementation of the alien wavelength use case implies some communications between the controller that controls not only the pluggable but also the device that hosts it, designated as the “end-terminal controller” (ETC) afterwards, and the RNC that configures the optical tunnel. Thus the physical characteristics of the pluggable are known by the RNC, and the performance and characteristics of the OLS path are known by the ETC. We assumed that there is no direct communication between the RNC and the ETC, and that the exchanges of needed information between the RNC and the ETC are handled through a hierarchical controller, which has the global knowledge of the multilayer topology. The corresponding control architecture is detailed in the following section. The RPCs and the asynchronous notifications used between the hierarchical controller and the RNC/ETC are detailed in the previous section of this document.

In this first step of implementation, alien wavelength service creation and deletion are supported, as well as service-feasibility-check and service reconfigure. Proposed RPCs and notifications only allow the implementation of unprotected services. Thus in case of a failure, the H-SDNC will be notified, and it is supposed to trigger relevant actions (service deletion, and re-creation after a new path has been found, possibly implying to retune the wavelength of the pluggable). Autonomous restauration performed by the RNC is not supported, since the network port of the pluggable needs to be disactivated by the ETC.

8.1 Control architecture

An example of the control architecture that could be deployed to handle the alien wavelength use case is shown on the following diagram. The hierarchical controller is supposed to have a global abstracted view of the multilayer topology, so

that it can handle the creation of services that span across several (either horizontal or vertical) domains. It is responsible for converting Customer Facing Services (CFS) into Resource Facing Services (RFS). Thus, it will orchestrate the service creation, and handle request-to / notification-from the different SDN controllers that are dedicated to a specific domain. The RNC is used in that use case to act as the OLS controller. The ETC used to manage devices that host pluggable could be an IP-SDNC, if the pluggable is hosted in a router, or any other controller depending on the type of equipment. Considering brownfield deployment, it could be the controller (or a NMS with needed API) of a pre-deployed OTN-XC that supports WDM interfaces.



High level Architecture considered for the alien wavelength use case

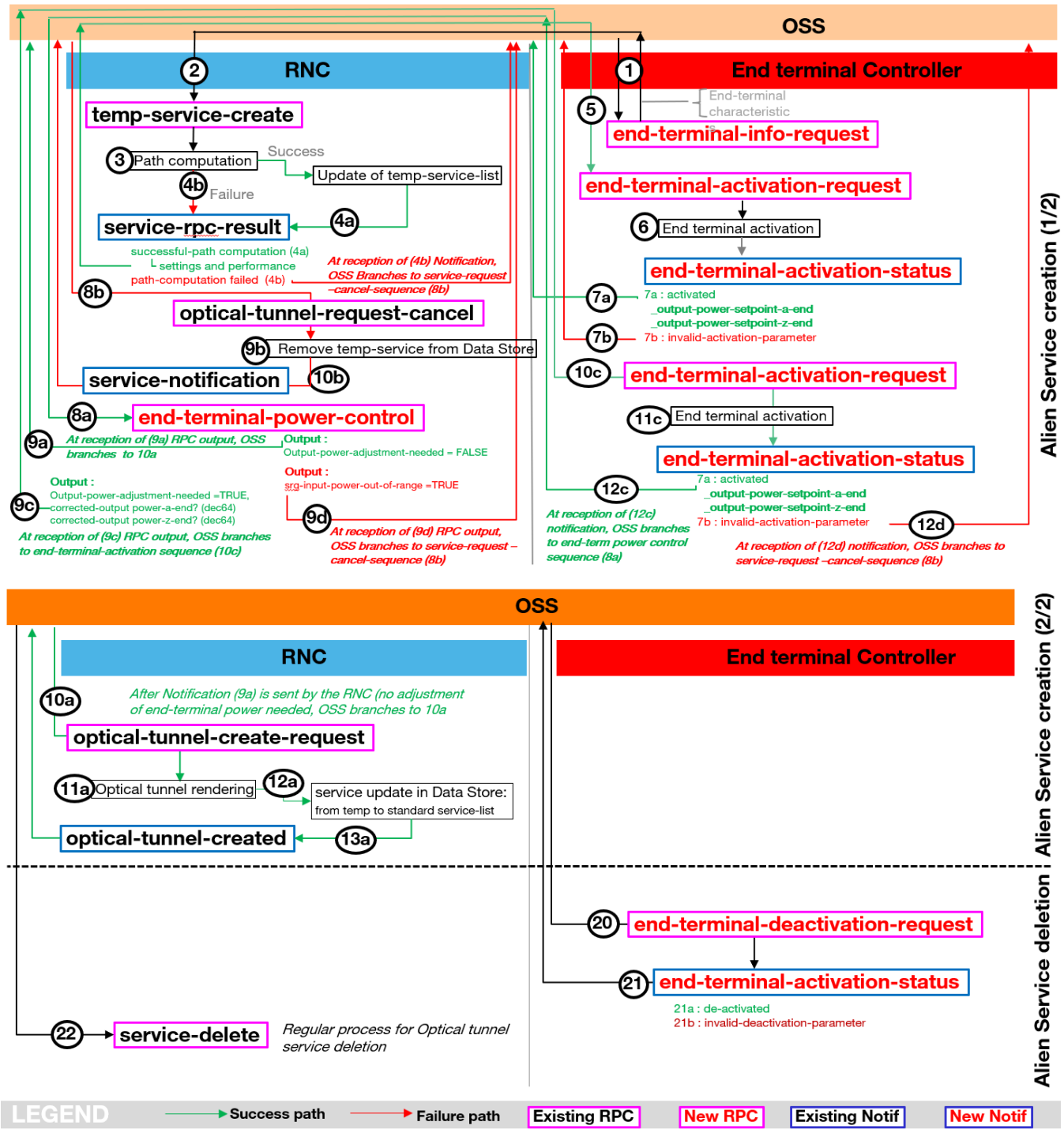
8.2 Alien wavelength service creation/DELETION

An alien wavelength service is based on transponders/external-pluggable, that are managed through an ETC, and may not follow OpenROADM optical specifications. It is provisioned on top of a ROADM infrastructure configured by the RNC as an OLS.

The transceivers (pluggable, integrated WDM interface, alien transponder…) are not handled trough the RNC, but the RNC is responsible (directly or indirectly) for the path computation.

Transceiver optical specifications may not follow OpenROADM standard, but they should be handled through an operational mode (openroadm/specific) that is described in the catalog: this implies the operational mode has been injected in the catalog using the dedicated RPC (add-openroadm-operational-modes-to-catalog / add-specific-operational-modes-to-catalog), so that the RNC knows what are the transceiver expected performances.

An alien wavelength service creation/deletion will be implemented through the sequence of actions described bellow.



Description of the flows implemented during service creation/deletion in the scope of an alien wavelength use case. ("New RPC" stands for post R7.1 RPCs)

Alien wavelength service creation flow:

1- The hierarchical controller (H-SDNC) sends an end-terminal-info-request to the End terminal Controller (ETC) to retrieve information about the end-terminal transceiver which could be an alien transponder, or a pluggable in a router. The

requested information includes the operational modes supported by the end terminal as well as its spectral range of operation.

2- The hierarchical controller (H-SDNC) sends a temp-service-create request to the RNC with the following attributes:

- Connection-type = optical tunnel
- Information on pluggable previously provided for A and Z end by the ETC

```
| | | +requesting-interface-properties
| | |   +---w supported-operational-modes* [preference]
| | |   | +---w preference                (Provides the level of preference associated with this mode)
| | |   | +---w operational-mode-id?      (Unique identifier of the mode)
| | |   +---w min-frequency?              (Minimum frequency the pluggable can be set to)
| | |   +---w max-frequency?              (Maximum frequency the pluggable can be set to)
| | |   +---w min-granularity?            (Minimum channel spacing required for correct frequency tuning)
```

3- The RNC Path Computation Engine (PCE) tries to find a path with an operational mode that best fit the request (highest preference). For this, it points to the operational-mode-catalog to get the physical parameters corresponding to the supported operational modes (tries first the preferred one).

4a- If it finds a path, the performances of the path and the settings are saved in the Data Store (temporary-service-list) temp-service-list/services/…topology & backup-topology aToZ/zToA /computation results … och/otsi resource). The RNC returns a service-rpc-result notification (notification-type: successful-path-computation-result-resource-reserved) with the results of the path computation providing the ETC with information about expected performances and settings to apply to the pluggable :

- Settings: frequency, width, optical-operational-mode-id, min/max-output-power
- Performances: osnr, gsnr

4b-In case the PCE fails to find a path, the RNC returns a service-rpc-result notification (notification-type : path-computation-failed) which terminates the process for the RNC.

5- If provided parameters returned through service-rpc-result notification to the H-SDNC fit with the ETC (such as an IP-SDNC) expectations, the ETC sets the pluggable according to the provided settings through an end-terminal-activation-request RPC

6- Configuration for activation is applied to the end-terminal optics.

7 –After the ETC as attempted to set the pluggable on both A and Z ends (frequency and output power set to the relevant values); it sends an end-terminal-activation-status notification to the HLC

7a- In case of success, the ETC sets the activation-status to “activated” and provides the output-power-setpoints for both A and Z ends.

7b- In case of failure, the ETC sets the activation-status to “invalid-activation-parameter”.

8a- If the end-terminal was successfully activated, the HLC exercised and end-terminal-power-control request.

9- The RNC provides in the output of the RPC the results of its evaluation of the output-power settings.

9a- If no correction needs to be applied, output-power-adjustment needed is set to False. HLC branches to (10a)

9c- If a correction needs to be applied (which corresponds to the case where the attenuation of the fiber connecting the end-terminal-TX to the SRG-PP-RX is not in the expected range), output-power-adjustment needed is set to True and the corrected-output-power-a/z-end are provided by the RNC to the HLC so that the correction can be applied on end-terminals. HLC branches to (10c)

9d- If no correction can be applied (which corresponds to the case where the attenuation of the fiber connecting the end-terminal-TX to the SRG-PP-RX is excessive and can not be corrected applying an output-power which is in the range of the end-terminal specifications), srg-input-power-out-of-range is set to True. At reception of (9d), the HLC branches to service-request –cancel sequence (8b)

8b- If the end-terminals can not be correctly set, the HLC sends an optical-tunnel-request-cancel to the RNC.

9b- The RNC removes from the Data Store the information relative to temp-service as the optical tunnel was not created.

10b- The RNC confirms to the HLC through a service-notification that the optical-tunnel has not been created. The HLC sends an end-terminal-deactivation-request to the ETC (20). This is followed by an end-terminal-activation-status notification sent by the A and Z end devices to the HLC (21) to confirm the end-terminals have been deactivated.

10a- In case no optical power adjustment is needed on the end-terminals (9a) The HLC sends to the RNC an optical-tunnel-create request. The common-id identifies the temporary service that shall be used as reference for the creation of the optical tunnel service. This RPC also confirms the frequency the pluggable shall be tuned to and provides information about the bandwidth at -3 and -10 dB (full-bandwidth-at-3/10-dB) that may be required by the ROADM device for correct power adjustment.

11a- The RNC renders the optical tunnel and updates the Service Data Store: the service is moved from the temp-list to the standard service-list. This ends the service creation process.

13a- The RNC sends to the HLC an optical-tunnel-created notification to confirms that the optical tunnel was successfully created.

10c- In case some power adjustment is needed on the end-terminals (9c), the output of the end-terminal-power-control is processed by the RNC to notify the HLC. The HLC exercise a new end-terminal-activation-request so that ETC can update the configuration of end-terminals and the output power can be adjusted.

11c- The end-terminal configuration is updated.

12c- In case of successful update of the end-terminal configuration, the ETC sends back an end-terminal-activation-status notification to the HLC providing information on the new output-power-setpoints. At reception of the notification, the HLC branches to end-terminal-power-control sequence (8a).

12d- In case the end-terminal configuration update fails, the ETC sends back an end-terminal-activation-status notification to the HLC stating that activation-parameter are invalid. At reception of this notification the HLC branches to the service-request-cancel sequence

Alien wavelength service deletion flow

20- An alien service deletion starts with the HLC exercising an end-terminal-deactivation-request to the ETC.

21- The ETC confirms whether the end-terminal could be deactivated or not sending back to the HLC an end-terminal-activation-status notification. In case of success, the activation-status is set to de-activated. In case of failure, an invalid-deactivation-parameter is returned.

22- If the end-terminal could be de-activated, the HLC exercise a regular service-delete RPC so that the RNC can delete the standard service corresponding to the optical tunnel.

Note : no specific flow has been planned to handle the case where the end-terminal could not be de-activated. This use case shall be handled through existing set of RPC and notifications by the HLC.