



TELECOM INFRA PROJECT

MUST

IP SDN Controller NBI

Technical Requirements

OOPT



Authors

Oscar González de Dios

Network Architect, Telefónica
oscar.gonzalezdedios@telefonica.com

Luis-Angel Munoz

IP & SDN Network Architect, Vodafone
luis-angel.munoz@vodafone.com

Maria Vazquez

IP & SDN Network Architect, Vodafone
maria.vazquez@vodafone.com

Ndifor Luc-Fabrice Ngwa

Senior Engineer, Fixed network and Technology Management, MTN
luc-fabrice.ndifor@mtn.com



Contributors

Arturo Mayoral López De Lerma

Programme Manager, Facebook.
adelerma@fb.com

Fritz-Joachim Westphal

Network Architect, Job Role, DT Organisation.
fritz-joachim.westphal@telekom.de

Philippe NIGER

Network Architect, Orange.
philippe.niger@orange.com



Change Tracking

Date	Revision	Author(s)	Comment
19/05/2021	V1.0	-- --	Final Version
--	--	--	--

Table of Contents

Authors	1	
Contributors	3	
Change Tracking	4	
Table of Contents	5	
List of Figures	9	
List of Tables	10	
1 Document Scope	14	
1.1 Use cases	14	
2 Yang Data Models of IP SDN Domain Controllers	17	
2.1 IP Controller Models Yang Data models	17	
2.1.1 Service Provisioning		18
2.1.2 Topology Models		19
2.1.3 Inventory Models		20
2.1.4 Traffic Engineering Models		20
3 RESTCONF/YANG Protocol considerations	22	
3.1 Root tree discovery	23	
3.2 YANG models discovery	23	
3.3 Data API (REST)	24	
3.4 Query filtering	24	
3.5 Data encoding	25	
3.6 Notifications	25	
3.6.1 SSE vs Websockets		26
3.7 Methods	26	
3.8 Responses	26	
4 Service Provisioning	29	
4.1 Introduction	29	

4.1.1	Service provider Network	29
4.1.2	Use Case Features	30
4.1.3	Use Case Operations	30
4.2	L3VPN Service Provisioning	31
4.2.1	L3VPN Structure and Classification	31
4.2.2	List of parameters to configure in the devices for L3VPNs	32
4.2.3	Workflow for L3VPN Creation	33
4.2.3.1	Create VPN (VPN.L3.Add)	34
4.2.3.2	Create VPN Node (VPN.L3.Node.Add)	36
4.2.3.3	Create VPN Network Access (VPN.L3.Access.Add)	37
4.2.3.4	L3VPN for 3G/4G Services. [L3VPN/Dot1Q/None/None]	39
4.2.3.5	Use case Parameters: L3VPN for 3G/4G Services	39
4.2.3.6	Description	39
4.2.3.7	Configuration example: SDTN NBI	41
5	Network Topology	45
5.1	Introduction	45
5.1.1	Use Case Features	47
5.1.2	Use case Operation	48
5.2	Assumptions and Definitions	48
5.2.1	Layer 1	48
5.2.2	Layer 2	50
5.2.3	Layer 3 & IGP	53
5.2.4	UNI Topology	54
5.3	Model Parameters	55
5.4	Operations	57
5.4.1	Retrieve all networks:	57
5.4.2	Retrieve one specific network:	57
5.4.3	Retrieve one specific node:	58
5.4.4	Retrieve one Specific Termination Point:	58
5.4.5	Retrieve one Links:	58
5.4.6	Retrieve destination of a Link:	58
5.4.7	Retrieve source of a Link:	58
5.5	Workflow	59
5.6	Obtain and export of L2 topology using IP domain controllers (ethernet links between routers)	61
5.6.1	L2 Parameters	61
5.6.2	Operations	63
5.6.2.1	Retrieve supporting L2 network attributes:	63
5.6.2.2	Retrieve supporting L2 link attributes:	64
5.6.2.3	Retrieve One Specific Termination Point with L2 Attributes:	64
5.7	Obtain and export of end-to-end IP topology using IP domain controllers (IGP	

Topology)	64	
5.7.1 L3 Parameters		65
5.7.2 Operations		66
5.7.3 Retrieve L3 unicast network attributes:		66
5.7.4 Retrieve supporting L3 network attributes:		67
5.7.5 Retrieve supporting L3 link attributes:		67
5.7.6 Retrieve One Node with L3 attributes:		67
5.7.7 Retrieve One Specific Termination Point with L3 attributes:		67
5.8 Export potential service end points in IP topology (UNI Topology)	68	
5.8.1 UNI Parameters		68
5.8.2 UNI-Operations		69
5.8.2.1 Retrieve all attachment points from a specific node:		69
5.8.2.2 Retrieve details from an attachment point:		69
5.9 Topology Examples	69	
5.9.1 Back-to-Back Routers Interconnection		69
5.9.2 Back-to-Back Routers VLAN Interconnection		71
5.9.3 Back-to-Back Routers Interconnection with OSPF		74
6 Network Inventory	77	
6.1 Need of NE management model	77	
6.1.1 Logical inventory		79
6.1.2 Yang model to expose device information via NBI		79
6.1.3 Retrieve logical [Interfaces] inventory use case		80
6.1.3.1 Parameters description, general table and example		80
6.1.3.2 Use case operations table		80
7 Traffic Engineering and Network Optimization	82	
7.1 Architecture and application of NBI	82	
7.2 TE Use Cases	83	
7.2.1 Use Case Features		84
7.2.2 Use Case Operations		84
7.3 LSP Creation, modify and delete with RSVP-TE no constraints	85	
7.3.1 PCE Initiated LSPs		86
7.3.2 PCC Initiated LSPs		87
7.3.3 Basic LSP Creation without constraints and restoration: Parameters		88
7.3.4 TE Tunnel Basic Operations		90
7.3.5 Examples		91
7.3.5.1 Example: Create a Basic TE-Tunnel		91
7.3.5.2 Example Show details of an existing LSP by name		92
7.3.5.3 Example UPDATE details of an existing LSP: Description		93
7.3.5.4 Example UPDATE details of an existing LSP: Administrative Status		93
7.3.5.5 LSP Creation: Workflow		94
7.4 LSP Creation, modify and delete with SR no constraints	94	

7.5	LSP create, modify and delete LSPs with delay, bandwidth and hop count constraints	95
7.5.1	LSP Creation with constraints Option 1: Global Constrain Policy	96
7.5.2	LSP Creation with constraints: Tunnels Constrain Policy	98
7.5.3	TE Tunnel Operations related to constraints	100
7.5.4	Examples	101
7.5.4.1	Example: Create Global Constraint Policy- Path constrain - <i>path-metric-delay-minimum</i>	101
7.5.4.2	Example: Create TE Tunnel with Associate Constraint Global Policy (NPC set)	103
7.5.4.3	Example: Create TE Tunnel: LSP with constrain path-metric-delay-minimum	104
7.6	LSP create, modify and delete with constraints (delay, bandwidth and hop count) and explicit Path (strict and loose)	109
7.7	LSP create, modify, and delete with constrains - Protection: Redundancy 1+1	113
7.7.1	Applying Protection, Restoration & Resilience to a tunnel	114
7.7.2	TE Tunnel Operations related to protection and restoration	118
8	Glossary	120
9	References	122
10	Annex A	123
10.1	Network Element Management Yang model	123
11	TIP Document License	126
12	Disclaimers	127



List of Figures

Figure 1. Uses Cases Categories	14
Figure 2. Possible L3VPN creation using the L3NM yang definition	34
Figure 3. VPN.L3.Add	35
Figure 4. VPN.L3.Access.Add	37
Figure 5. Service topology diagram	40
Figure 6. Service provision options	41
Figure 7. Supporting Network and Supporting Node	46
Figure 8. L1 Topology Tree	50
Figure 9. IETF Network usage	56
Figure 10. Topology Discovery Workflow	60
Figure 11. Back-to-Back Routers	70
Figure 12. Back-to-Back Routers connected using VLANs	72
Figure 13. Back-to-Back Routers interconnection with OSPF	75
Figure 14. Model hierarchy for Inventory	79
Figure 15. PCE Initiated	87
Figure 16. PCC Initiated	88
Figure 17. LSP Creation	94
Figure 18. IETF TE Name Path Constrains	97
Figure 19. Tunnel Constrains	99
Figure 20. Initial scenario	114
Figure 21. Switching scenario	114

List of Tables

Table 1. Service provisioning use cases	15
Table 2. Support to inventory use cases	15
Table 3. Discovery and topology use cases	15
Table 4. Traffic engineering / PCE use cases	15
Table 5. Service provisioning use cases yang models	18
Table 6. Topology use cases yang models	19
Table 7. Inventory use cases yang models	20
Table 8. Traffic Engineering use cases yang models	20
Table 9. RETCONF Query filtering requirements	24
Table 10. Create VPN required parameters	35
Table 11. Create VPN profiles required parameters	36
Table 12. Create VPN Node required parameters	37
Table 13. Create VPN Network Access required parameters	38
Table 14. L3VPN for 3G/4G Service required parameters	39
Table 15. L2 Topology Tree	52
Table 16. UNI Topology Tree	55
Table 17. Network Topology Network required parameters.	56
Table 18. Network Topology Node required parameters.	56
Table 19. Network Topology Termination Points required parameters.	57
Table 20. Network Topology Link required parameters.	57
Table 21. L2 Topology Network required parameters.	61
Table 22. L2 Topology Node required parameters.	62
Table 23. L2 Topology Termination Points required parameters.	62
Table 24. L2 Topology Termination Points Ethernet required parameters.	63
Table 25. L2 Topology Termination Points Legacy required parameters.	63
Table 26. L2 Topology Links required parameters.	63

Table 27. L3 Topology Network required parameters.	65
Table 28. L3 Topology Nodes required parameters.	65
Table 29. L3 Topology Termination Points required parameters.	65
Table 30. L3 Topology Termination Points type IP required parameters.	66
Table 31. L3 Topology Termination Points type Unnumbered required parameters.	66
Table 31. L3 Topology Termination Points type Interface-name required parameters.	66
Table 31. L3 Topology Link required parameters.	66
Table 31. Service End Points UNI Network required parameters.	68
Table 31. Service End Points UNI Node required parameters.	68
Table 31. Service End Points UNI Service Access Points required parameters.	69
Table 32 Attributes (configurable and state) for basic Basic RSVP-TE Tunnel	90
Table 33 Basic LSP Operations	91
Table 34 Sample parameters of a Basic unprotected RSVP-TE Tunnel creation	92
Table 32 Attributes (configurable and state) for LSP Creation, modify and delete with SR no constraints	95
Table 32 Attributes (configurable and state) for LSP Creation with constraints Option 1: Global Constrain Policy	98
Table 32 Attributes (configurable and state) for LSP Creation with constraints Tunnel Constrains Policy	100
Table 35 Basic LSP Operations	101
Table 36 Sample parameters of an unprotected RSVP-TE Tunnel with associated NPC creation	104
Table 32 Attributes (configurable and state) for LSP create, modify and delete with constraints (delay, bandwidth and hop count) and explicit Path (strict and loose)-1110	
Table 32 Attributes (configurable and state) for LSP create, modify and delete with constraints (delay, bandwidth and hop count) and explicit Path (strict and loose)-2111	
Table 32 Attributes (configurable and state) for LSP create, modify and delete with constraints (delay, bandwidth and hop count) and explicit Path (strict and loose)-3111	
Table 32 Attributes (configurable and state) for LSP creation using protection options	117



Table 37 TE Tunnel Protection/RestorationOperations



Document Scope

Implementing a complete standard specification is a time-consuming process. To accelerate the adoption of the specifications, the TIP MUST OOPT subgroup has compiled the needs of multiple operators, selecting the most relevant common use cases [1]. Based on this information, technical requirements are prepared and shared with the industry in an open manner. The produced specifications will be incorporated as part of each operator processes.

1 Document Scope

Implementing a complete standard specification is a time-consuming process. To accelerate the adoption of the specifications, the TIP MUST OOPT subgroup has compiled the needs of multiple operators, selecting the most relevant common use cases [1]. Based on this information, technical requirements are prepared and shared with the industry in an open manner. The produced specifications will be incorporated as part of each operator processes.

This document provides the technical requirements of the **North Bound Interfaces (NBI)** mandatory to **be exposed by IP Domain Controllers**, which includes an IP Domain Path Computation Element.

1.1 Use cases

The use cases categories defined in MUST are depicted in Figure 1

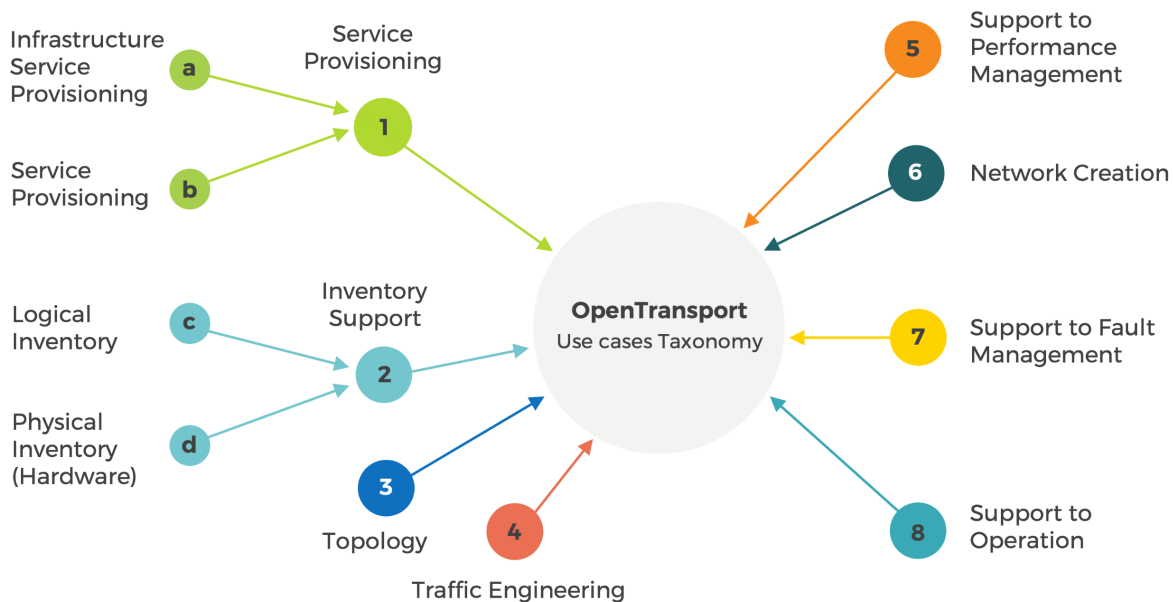


Figure 1. Uses Cases Categories

The purpose of this document is to identify, describe and compile the set of use cases as well as define the requirements, flows and details of each use cases in the IP Domain Controller. The Northbound Interface calls needed by each use case is presented here.

The following tables list the use cases covered in the document.

Service Provisioning		Section
1.1	L3VPN for 3G/4G Services. [L3VPN/Dot1Q/None/None]	4.2.3.4

Table 1. Service provisioning use cases

Support to Inventory		Section
2.1	Retrieve logical [Interfaces] Inventory	6.1.3

Table 2. Support to inventory use cases

Discovery and topology		Section
3.1	Obtain and export of end-to-end IP topology using IP domain controllers (IGP Topology)	5.7
3.2	Obtain and export of L2 topology using IP domain controllers (ethernet links between routers)	5.6

Table 3. Discovery and topology use cases

Traffic Engineering / PCE		Section
4.1	LSP Creation, modify and delete with RSVP-TE no constraints	7.3
4.2	LSP Creation, modify and delete with SR no constraints	7.4
4.3	LSP create, modify and delete with constraints (delay, bandwidth and hop count)	7.5
4.4	LSP create, modify and delete with constraints (delay, bandwidth and hop count) and explicit Path (strict and loose)	7.6
4.5	LSP create, modify and delete with constraints (delay, bandwidth and hop count) Protection: Redundancy 1+1	7.7

Table 4. Traffic engineering / PCE use cases

2

Yang Data Models of IP SDN Domain Controllers

The IP SDN Domain Controller will contain a set of network management datastores defined in Yang 1.1 [2] which will be accessible via RESTCONF [3] NBI (see section 0 for RESTCONF/YANG Protocol considerations). Current specification is NOT NMDA Compliant. Note that it is forecasted in future version of the specification to upgrade to NMDA.

2 Yang Data Models of IP SDN Domain Controllers

The IP SDN Domain Controller will contain a set of network management datastores defined in Yang 1.1 [2] which will be accessible via RESTCONF [3] NBI (see section 0 for RESTCONF/YANG Protocol considerations). Current specification is NOT NMDA Compliant. Note that it is forecasted in future version of the specification to upgrade to NMDA.

The Domain Controller will offer (as per current specification):

- Network Service modules. These modules will be used as datastores of Network Services. Hence, they can be used for CRUD operation on the services. These modules are defined based on Network Service Data models.
- Topology modules. These modules will be used to export, in read-only mode information about the network topology at different layers. These modules are defined with IETF Topology Yang models.
- Traffic engineering modules. These modules will be used for CRUD operations on TE tunnels and TE topologies
- Inventory

The Path Computation Element of the Domain Controller will offer (as per current specification):

- TE Tunnels modules. These modules will be used for CRUD operations on TE tunnels.
- TE Topology module. This module will be used to query the TE topology
- Path Computation module. These modules will be used for performing path computations on the PCE. (This will be specified in the next version).

2.1 IP Controller Models Yang Data models

The set of YANG data models used in these documents is described below.

2.1.1 Service Provisioning

A Service Network Model is a YANG module that describes a Network Service in the Service Provider Network. It contains information of the Service Provider network and might include allocated resources.

The service models that need to be supported in MUST specification are listed below. Note that, in the first specification, only a subset of the envisioned models is considered, see the foot notes:

Model	RFC / Draft
L3NM¹	https://tools.ietf.org/html/draft-ietf-opsawg-l3sm-l3nm-08
VPN2 common	https://tools.ietf.org/html/draft-ietf-opsawg-vpn-common-07
Routing-Policy³	https://tools.ietf.org/html/draft-ietf-rtgwg-policy-model-09
Acces Control List⁴	https://tools.ietf.org/html/rfc8519
BGP-Policy-Extension⁵	https://tools.ietf.org/html/draft-ietf-idr-bgp-model-08
L2NM⁶	https://tools.ietf.org/html/draft-ietf-opsawg-l2nm-01

Table 5. Service provisioning use cases yang models

¹ When the L3NM document becomes RFC the specification for L3NM will be considered for update to minimize interface changes. Current spec is based on -08 version which is in the Last Call process.

² When the VPN common document becomes RFC the specification for L3NM will be considered for update to minimize interface changes. Current spec is based on -07 version which is in the Last Call process.

³ This model will be used in the next version of the specification in advanced VPN use cases

⁴ This model will be used in the next version of the specification in advanced VPN use cases

⁵ This model will be used in the next version of the specification in advanced use cases



2.1.2 Topology Models

The topology models that need to be supported in current specification are:

Model	RFC / Draft
IETF Network Topology	https://tools.ietf.org/html/rfc8345
IETF L3 Network Topology	https://tools.ietf.org/html/rfc8346
IETF L2 Network Topology	https://tools.ietf.org/html/rfc8944
IETF UNI Topology	https://tools.ietf.org/html/draft-ogondio-opsawg-uni-topology-00
IETF TE Topology	https://datatracker.ietf.org/doc/rfc8795/

Table 6. Topology use cases yang models

2.1.3 Inventory Models

The model that needs to be supported for inventory purposes in the MUST specification is the following:

Model	RFC / Draft
Augmentation of IETF Network Topology⁶	No available online. Description of the augments in section 6.

Table 7. Inventory use cases yang models

In this case, as there is no full compliant yang model in the standards, the proposed extensions will be taken to the relevant standard body.

2.1.4 Traffic Engineering Models

The traffic engineering models that need to be supported in the MUST specification are:

Model	RFC / Draft
IETF Traffic Engineering Tunnels and Interfaces	<a href="https://tools.ietf.org/html/draft-ietf-teas-yang-te-26<sup>7</sup>">https://tools.ietf.org/html/draft-ietf-teas-yang-te-26⁷
Traffic Engineering Common YANG Types	https://tools.ietf.org/html/rfc8776
YANG Data Model for requesting Path Computation⁸	https://tools.ietf.org/html/draft-ietf-teas-yang-path-computation-12

Table 8. Traffic Engineering use cases yang models

⁶ This model will be used in the next version of the specification when describing the L2VPNs

⁷ There is NO available Network Inventory model. Hence, it is proposed to augment the standard IETF Network Topology with the relevant device model

⁸ When TE document becomes RFC, the specification for Traffic Engineering will be considered for update to minimize API changes. Current version is based on -26 version and is in the last call process.

⁹The path computation function will be included in future releases. Here it is included as reference.

3

RESTCONF/YANG Protocol considerations

MUST prescribes the use of RESTCONF [RFC 8040] as the transport protocol for all the defined management operations in the SDN architecture NBIs.

3 RESTCONF/YANG Protocol considerations

MUST prescribes the use of RESTCONF [RFC 8040] as the transport protocol for all the defined management operations in the SDN architecture NBIs.

RESTCONF is a HTTP-based protocol that provides a programmatic interface for accessing data defined in YANG 1.1 [2] and using the data store concepts defined in the Network Configuration Protocol (NETCONF) [4].

The RESTCONF specification consists of the following resources:

- **{+ RESTCONF}/data (Data API):** Create/Retrieve/Update/Delete (CRUD) based API for the data trees defined in the information model YANG files of section 2. Current specification of the IP NBI Specification does not mandate to support all optional elements in the Yang models. Each use case specifies the minimum paths and elements to support the use case.
- **{+ RESTCONF}/operations (Operations API):** Current version of the IP NBI Specification does not require to implement any operation. Future versions might contain operations.
- **{+ RESTCONF}/data/ietf-RESTCONF-monitoring:RESTCONF-state/streams (Notifications API):** Notifications implementation of RESTCONF protocol is defined in <https://tools.ietf.org/html/rfc8040#section-6.3>.
- **{+ RESTCONF}/yang-library-version:** This mandatory leaf identifies the revision date of the "ietf-yang-library" YANG module that is implemented by this server.

The Network Management Datastore Architecture (NMDA) defined in [RFC8342] is the target implementation for RESTCONF NBIs of the MUST group in all the technology domains (IP, MW, Optical transport.). The inclusion of NMDA for the defined use cases of the IP SDN Domain controller in this deliverable will be considered in future releases.

3.1 Root tree discovery

The RESTCONF API **{+RESTCONF}** root resource can be discovered by getting the *"/.well-known/host-meta"* resource ([RFC 6415]) and using the <Link> element containing the *"RESTCONF"* attribute.

The client might send the following:

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

The server might respond as follows:

```
HTTP/1.1 200 OK
Content-Type: application/xrd+xml
Content-Length: nnn

<XRD xmlns='http://docs.oasis-open.org/ns/xri/xrd-1.0'>
  <Link rel='RESTCONF' href='/RESTCONF'/>
</XRD>
```

3.2 YANG models discovery

RESTCONF utilizes the YANG library defined in RFC 7895 [5] to allow a client to discover the YANG module conformance information for the server, in case the client wants to use it.

The mandatory **{+RESTCONF}/yang-library-version** resource is used to clearly identify the version of the YANG library used by the server.

The server MUST implement the "ietf-yang-library" module, which MUST identify all of the YANG modules used by the server, within the "modules-state" list resource. The modules set resource is located at:

- {+RESTCONF}/data/ietf-yang-library:modules-state

3.3 Data API (REST)

Current NBI specification states that the support of the RESTCONF Data API is mandatory. However, to facilitate the implementation, based on the use cases described in the document, and the yang models in section 2, a subset of the possible paths and a subset of the possible parameters is requested.

3.4 Query filtering

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

Name	Methods	Description
content	GET, HEAD	Select config and/or non-config data resources
depth	GET, HEAD	Request limited subtree depth in the reply content
fields	GET, HEAD	Request a subset of the target resource contents
filter	GET, HEAD	Boolean notification filter for event stream resources
start-time	GET, HEAD	Replay buffer start time for event stream resources
stop-time	GET, HEAD	Replay buffer stop time for event stream resources

Table 9. RESTCONF Query filtering requirements

The specific use of these query parameters will be detailed in the different Use Cases Low Level Design (LLDs) included in section.

3.5 Data encoding

NBI implementation MUST support JSON encoding formats in accordance with [Section 3.2 of \[RFC 8040\]](#).

The solution adhering to this specification MUST support media type "application/yang-data+json". This MUST be advertised in the HTTP Header fields "Accept" or "Content-Type" of the corresponding HTTP Request/Response messages.

The YANG data nodes in bodies and responses are encoded in json according to the specifications of [RFC7951](#).

3.6 Notifications

The solution adhering to this specification must support all YANG-defined event notifications included in the information models included in posterior sections of this document.

The solution implementing the RESTCONF server must expose its supported notification streams by populating the "RESTCONF-state/streams" container definition in the "ietf-RESTCONF-monitoring" module defined in [Section 9.3 of \[RFC 8040\]](#). The streams resource can be found at:

- `{+RESTCONF}/data/ietf-RESTCONF-monitoring:RESTCONF-state/streams`

The RESTCONF server MUST support at least the NETCONF event stream as defined in [Section 3.2.3 of \[RFC5277\]](#) with JSON encoding format being supported and advertised as two different location resources as described in [Section 6.2 of \[RFC 8040\]](#).

The RESTCONF server MUST support the RESTCONF Notifications subscription mechanism is defined in [Section 6.3 of \[RFC 8040\]](#).

The solution must support the “filter” Query Parameter, as defined in [Section 4.8.4 of \[RFC 8040\]](#), to indicate the target subset of the possible events being advertised by the RESTCONF server stream. The “filter” query parameter URI SHALL be listed in the “capability” leaf-list defined in [Section 9.3 of \[RFC 8040\]](#), to advertise the server capability of supporting the “filter” query parameter. This resource shall be located at:

- {+RESTCONF}/data/ietf-RESTCONF-monitoring:RESTCONF-state/capabilities

3.6.1 SSE vs Websockets

The RESTCONF standard defines the Server Sent Events (SSE) [W3C.REC-xml-20081126] as the standard protocol for RESTCONF stream notification service.

Server Sent Events (SSE) [W3C.REC-eventsource-20150203] is required as the target protocol for notifications in the NBI.

3.7 Methods

In the MUST NBI specification, GET, POST, PATCH and DELETE methods are specified to solve the use cases.

In current version of the specification, it is requested a “Plain PATCH” as described in [Section 4.6.1 of \[RFC 8040\]](#).

3.8 Responses

The RESTCONF Server MUST implement the following responses in the RESTCONF data resources ({+RESTCONF}/data/) :

Responses for GET Operations

200 OK	The content is successfully sent
400 Bad Request	The request is invalid (multiple reasons)
401 Unauthorized	The user is not authorized to read the content
404 Not Found	The target resource instance does not exist



Responses for POST Operations

201 Created	The content is successfully created
400 Bad Request	The request is invalid (multiple reasons)
401 Unauthorized	The user is not authorized to create the content
404 Not Found	The target resource instance does not exist
409 Conflict	The target resource instance already exist

Responses for PATCH Operations

204 No Content	The content is successfully modified.
400 Bad Request	The request is invalid (multiple reasons)
403 Forbidden	The user is not authorized to modify the content
404 Not Found	The target resource instance does not exist

Responses for DELETE Operations

204 No Content	The content has been successfully deleted
400 Bad Request	The request is invalid (multiple reasons)
403 Forbidden	The user is not authorized to delete the content
404 Not Found	The target resource instance does not exist

4

Service Provisioning

The use cases in this category are aimed at helping in the provisioning the network connectivity services that are implemented in a Service Provider Network. The IP SDN domain controller will expose APIs based on RESTCONF/Yang to allow the creation of the connectivity service in the network in a vendor-agnostic fashion. Note that the orchestration of the service happens on top of the SDN layer.



4 Service Provisioning

4.1 Introduction

The use cases in this category are aimed at helping in the provisioning the network connectivity services that are implemented in a Service Provider Network. The IP SDN domain controller will expose APIs based on RESTCONF/Yang in order to allow the creation of the connectivity service in the network in a vendor-agnostic fashion. Note that the orchestration of the service happens on top of the SDN layer.

The Operators from MUST have prioritized the APIs to manage the provisioning of Virtual Private Networks. The IP Virtual Private Network (VPN) defined in RFC 4364 provides a multipoint, routed service to the customer to carry traffic over an IP/MPLS core. The L3VPNs are widely used in the deployments of 3G/4G, fixed and enterprise services principally because several traffic discrimination policies can be applied in the network to transport and guarantee the right SLAs to the transport service customers. The Virtual Private Networks are also capable of proving layer 2 connectivity to the customers (which can be either internal or external).

4.1.1 Service provider Network

The network over which the services are implemented is based on IP/MPLS routers. In order to facilitate the comprehension of the examples thought the document, we provide a typical layer distribution of the IP/MPLS network devices, which includes the following roles:

- Cell Site Gateways.
- Aggregator PE
- Access routers
- P routers
- Interconnection routers.

Throughout the document, we will also use the terminology according to the hierarchical level (HL1 for interconnection router, HL2 for the P routers, HL3/4/5 for the access, aggregators, and cell site gateways accordingly).

The access and cell site gateways usually form regions concentrated in an aggregator router, in flexible hierarchy levels depending on the topology, that as a naming convention start in HL5 for a Cell Site, HL3 for Access Routers working as intermediate hubs towards the Aggregator PE. This aggregation device has the Autonomous System Border Router (ASBR) role as well as Inline Router reflector for its Region. Thus, to forward the traffic from the L3VPN services, the ASBR routers from each region establish an eBGP session against the core routers. This session exports the Router ID plus Label information of all the routers in the region using BGP-LU. Additionally, there is another eBGP session between the Access PE hubs (HL3) of the region and the core Router-Reflectors to export the VPNv4 routes from each VPN service. This eBGP session requires a mandatory a Next-Hop-Unchanged configuration to avoid network loops or misconfigured paths. All this control plane setup allows the creation of an end-to-end LSP from the access HL5 to the platforms without changes in the configuration during the service provisioning.

Additionally, to deploy any of the VPN services the network must fulfill the following basic requirements:

- IGP connectivity established between access routers and cell site (HL3 and HL5).
- LDP / RSVP session between access routers and cell site (HL3 and HL5).
- MPLS enable session between access routers and cell site (HL3 and HL5).
- MP-BGP (family vpnv4, ipv4)

4.1.2 Use Case Features

Features

BASIC CREATION

- Service type
 - L3VPN
 - L2VPN
- **NE-ID** and **NE-Interface**
- CE-PE Routing protocol Selection
- CE-PE Access Encapsulation
- Underlay Transport Selection
 - RSVP-TE / SR-TE
 - LDP

ADVANCE CREATION

- Multicast Support
- Service Binding/Mapping

UPDATE

- Change Service Status (**Up & Down**)
- Change Service Parameters

4.1.3 Use Case Operations

Operations

CREATE

POST

Create a LxVPN services, nodes and accesses using several topologies.
 Create a VPN node in a particular VPN Service
 Create an access to a particular VPN Node.
 Create CE-PE routing parameters
 Create accesses ethernet encapsulation, requirements.
 Create QoS relationships of a service.

READ

GET

Retrieve a LxVPN Services, Nodes and Accesses Status
 Retrieve a VPN Node in a particular VPN Service
 Retrieve Accesses parameters

UPDATE

PATCH

Update a LxVPN Services, Nodes and Accesses Status
 Update a VPN Node in a particular VPN Service
 Update accesses parameters such as IP routing and ethernet encapsulation.

DELETE

DEL

Delete all the LxVPN services, all nodes or all accesses.
 Delete specific LxVPN services, nodes or all accesses.
 Delete a VPN node in a particular VPN Service
 Delete accesses parameters

4.2 L3VPN Service Provisioning

The IP SDN Use cases consider in this deliverable are:

ID	Use case Title	Section
1.1	L3VPN for 3G/4G Services [L3VPN/Dot1Q/None/None]	4.2.3.4

4.2.1 L3VPN Structure and Classification

The VPN services can be classified using its operational characteristics, thus we have defined the following structure to identify each possible variation of the VPN service. Additionally, we have included an identifier of the Use case (**Use case Definition**) to map the relationship between the commercial offer and the real deployment:

- **Use case Definition:** Type of services deployed in the network.
- **Functional Parameters:** Used to find common structures in the configured Services. This classification has the following structure:

- **VPN Service Type:** Type of service configured
 - L2VPN
 - L3VPN
- **End Point Connection Type:** Encapsulation details used in the CE-PE connection.
 - None
 - Dot1q
 - QinQ
 - L2VPN (ipipe/epipe)
 - VPI/VCI
 - CEM
 - Loopback
- Routing Protocol used to connect the CE: Routing details used in the CE-PE connection.
 - Direct
 - Static
 - BGP***
 - OSPF*
 - ISIS*
 - RIP*
- QoS policies applied in the service:
 - None
 - QoS Overwriting
 - CIR / PIR policies

4.2.2 List of parameters to configure in the devices for L3VPNs

The L3VPN creates a virtual routing network instance (VRF) in each of the nodes involved in service deployment. This, routing instance allows the routing information propagation between the sites involved in the service.

The **L3VPN** parameters configured in the cell site router is the following if the VPN does not exist:

```
+ Service Name (VPNID)
+ Autonomous system (if not already commissioned)
+ Router ID (usually is the loopback address)
+ Route Distinguisher
+ Import / Export Target
+ Operational status / Administrative status
```



```
+ End Points (1...n)
  + Interface
  + Encapsulation
  + VLAN ID
  + IP Address / mask
  + MTU
  + Administrative Status
+ Policies
+ Qos
  + Ingress / Egress Shaping
  + Policing
+ CE-Connectivity ()
  + Routing Protocols
+ Topology (e.g., hub-spoke, in our case)
+ Topology-role
```

The same kind of validations must be done in the aggregation layer.

4.2.3 Workflow for L3VPN Creation

The Northbound interface between systems and network shall integrate only mandatory information that is held at the upper layer or that is requested by the customer service. As day one previous requirements for VPN creation, SITE “customer service layer” information to be assigned for each connection is already translated into network nodes and its PE termination points, including new site-bearers provisioning for the Site.

For this specific use case, the parameters and request will have the following four steps:

1. Create VPN:
 - a. Add VPN common attributes.
 - b. Add Import/Export Profiles
2. Create every VPN-Node (VRF) and provide the common parameters. Specify the network element in which the VPN-Node is deployed. Provide the node role

Attach each interface to the VPN-Node (VRF) by creating a new instance of vpn-network access. Create the connection protocols over each of the VPN Network accesses.

The corresponding workflow is as depicted in **Figure 2** while all the intermediate steps are described in the following subsections.

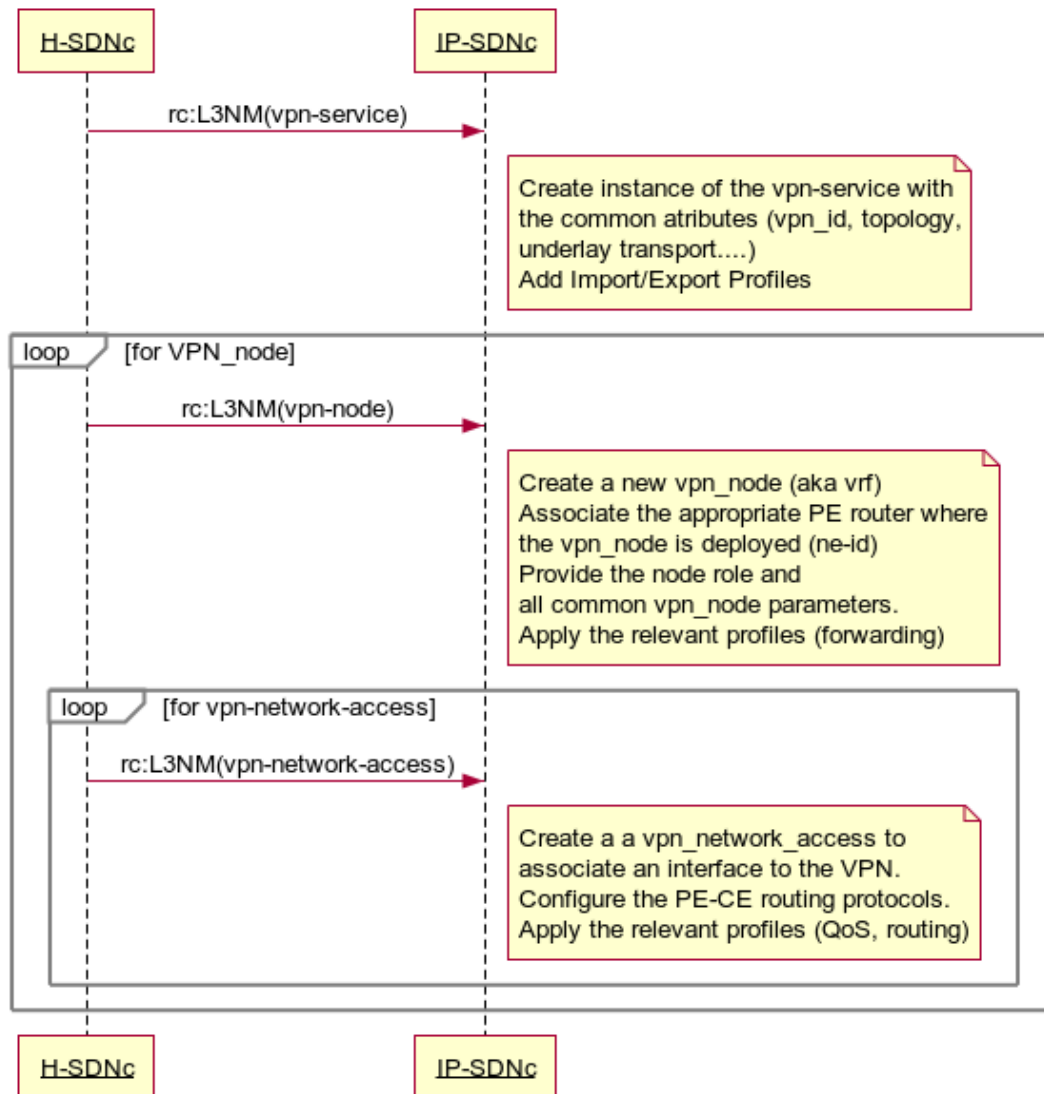


Figure 2. Possible L3VPN creation using the L3NM yang definition

4.2.3.1 Create VPN (VPN.L3.Add)

The table and figure below describe the parameters required for the creation of the new VPN service in the SDN layer. This request will not necessarily generate any direct configuration to be sent to the network elements, as it will only serve as a notification for

the controller to allocate resources for the new service. If information such as VPN-nodes or import/export profiles is specified at the creation of the VPN service, the SDN layer can use this information to push this configuration to the network elements. The information includes a VPN ID, a customer ID and a VPN service topology.

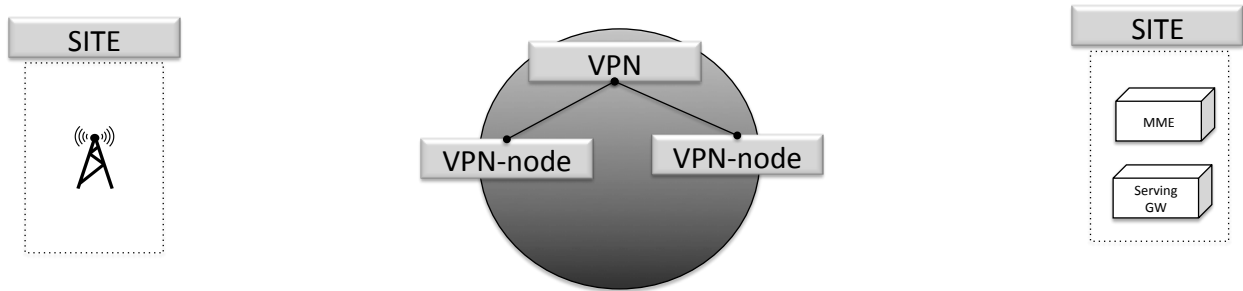


Figure 3. VPN.L3.Add

The set of parameters and the responsibilities are the following:

Parameter	Responsible	Type	Description
VPN ID	NBI	vpn-id	Reference to an IP VPN.
Customer Name	NBI	string	Name of the customer that actually uses the VPN service. In the case that any intermediary (e.g., Tier-2 provider or partner) sells the VPN service to their end user on behalf of the original service provider (e.g., Tier-1 provider), the original service provider may require the customer's name to provide smooth activation/commissioning and operation for the service.
vpn-type	NBI	identityref	For the L3NM, this is typically set to BGP/MPLS L3VPN
VPN service topology	NBI	identityref	VPN service topology.
vpn-description	NBI	string	Textual description of a VPN service.

Table 10. Create VPN required parameters

Additionally, the vpn-instance-profiles, like import/export profiles, can be added when creating the VPN service, using the following parameters. This parameters can be assigned at operator choice, either from NBI or defined by the SDN Domain network controller (NET) (e.g.: RD assignment modes are supported: direct assignment,

automatic assignment from a given pool, automatic assignment, and no assignment):

Parameter	Responsible	Type	Description
profile-id	NBI/NET	string	Unique identifier e.g. for an import/export profile defined so that the controller can identify RTs and RDs to be configured for a given VRF in one or multiple VPN nodes.
role	NBI	identityref	role of the VPN instance profile in the VPN (e.g. Hub-role)
Local-autonomous-system	NBI	inet:as-number	AS number of the VRF
RD	NBI/NET	rt-types:route-distinguisher	Route distinguisher
Address-family	NBI/NET	identityref	It can be set to IPv4, IPv6, or dual-stack.
Vpn-target/route-target	NBI/NET	rt-types:route-target	Set of route-targets to match for import and export routes to/from VRF. Route target value
Vpn-target/route-target-type	NBI/NET	rt-types:route-target-type	Set of route-targets to match for import and export routes to/from VRF. Route target type (import/export/both).
Maximum routes	NBI/NET	uint32	Defines 'maximum-routes' for the VRF. Maximum prefixes that the VPN node can accept for a given address family and routing protocol

Table 11. Create VPN profiles required parameters

4.2.3.2 Create VPN Node (VPN.L3.Node.Add)

VPN nodes (typically known as VRFs or network instance in the devices) can be included on VPN creation if the locations (PE routers) are known in advance. To do so, the following parameters are configurable:

Parameter	Responsible	Type	Description
Vpn-node-ID	NBI/NET	vpn-common:vpn-id	Is an identifier that uniquely identifies a node that enables a VPN network access.
NE-ID	NBI/NET	string	Unique identifier for a network element where to instantiate the VRF. This identifier may be a string, a UUID, an IP address, etc.
Description	NBI/NET	string	Textual description of a VPN node.

Router-ID	NBI	rt-types:router-id	Indicates a 32-bit number that is used to uniquely identify a router within an Autonomous System.
local-autonomous-system	NBI	inet:as-number	AS number of the VRF, if different from the base instance.
active-vpn-instance-profiles/profile-id	NBI	leafref	Reference to the active VPN instance profile for this VPN node (e.g., profile id of hub-role)
group-id	NBI	string	The 'group-id' is used to associate, e.g., redundancy or protection constraints with VPN nodes belonging to the same group
Status/ admin-status	NBI	identityref	Tracks the status of a node involved in a VPN service.

Table 12. Create VPN Node required parameters

4.2.3.3 Create VPN Network Access (VPN.L3.Access.Add)

The activation of a new VPN network access (e.g., new client port to be included in a VRF, and therefore described inside a vpn-node) requires some parameters to be sent from the systems layer. These include the ID of the new access, description, the port-id (reference to the node, device, and interface for the connection), encapsulation of the client traffic, routing protocols, AS number (not necessary for our use case, with direct routing), IP addresses to be configured for the PE-CE link. This information will trigger configuration in the interface of the corresponding interface in the PE router.

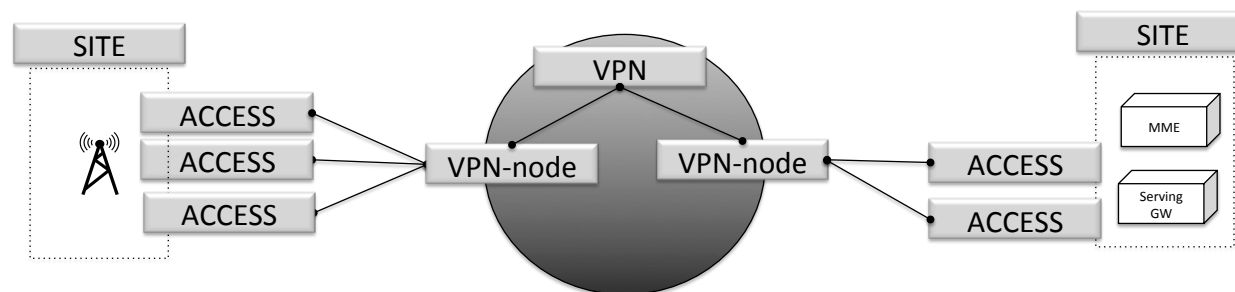


Figure 4. VPN.L3.Access.Add

Parameter	Responsible	Type	Description
vpn-network-access/id	NBI	vpn-	Identifier for the VPN network

		common:vpn-id	access.
port-id	NBI	vpn-common:vpn-id	Indicates the physical port on which the VPN network access is bound
description	NBI	string	Includes a textual description of the VPN network access
vpn-network-access-type	NBI	identityref	Describes the type of connection: loopback, point-to-point or multipoint, IRB (connection coming from l2vpn service, 'l2vpn-id' would be used to identify it).
Status	NBI	identityref	Only administrative can be configured.
Connection	NBI		Represents and groups the set of Layer 2 connectivity from where the traffic of the L3VPN in the VPN Network access is coming
Connection/encapsulation/type	NBI	identityref	dot1q, qinq, etc. parameters.
Connection/l2-tunnel-service/Pseudowire/vcid (if necessary)	NBI	uint32	If connection (vpn-network-access) is a pseudowire, any necessary parameter will be enclosed here (e.g. Virtual circuit ID).
Ip-connection/[ipv4/ipv6]/[dhcp/static]	NBI	inet:ipv4-address	Protocol for IP allocation. Usually defined as static and IPs enclosed under the "address" container.
Service/Bandwidth	NBI	uint64	Input and output bandwidth for the service.
Service/QoS [standard profile]	NBI	string	Reference to a standard profile for QoS, defined at the vpn-profiles.
Routing-protocols/ospf Address-family Area-address Metric MTU Security	NBI	vpn-common:rtg-ospf	Configuration for CE-PE routing protocol, when ospf is chosen.
Routing-protocols/bgp ASnumber address-family Neighbor Multihop Security	NBI	vpn-common:rtg-bgp	Configuration for routing protocol, when bgp is chosen.
Routing-protocols/static/ipvX-prefixes/lan next-hop lan-tag	NBI		Configuration for routing protocol, when static is chosen.

Table 13. Create VPN Network Access required parameters

4.2.3.4 L3VPN for 3G/4G Services. [L3VPN/Dot1Q/None/None]

Number	2.1
Name	L3VPN for 3G/4G Services. [L3VPN/Dot1Q/None/None]
Technologies involved	IP
Brief description	L3VPN services are widely deployed in the IP/MPLS networks; These services consume several logical resources (RD, RT, VLANs, IP Address, etc.) to be deployed correctly. The maintenance is done in the network daily and several areas get involved.

4.2.3.5 Use case Parameters: L3VPN for 3G/4G Services

Name	1.1 L3VPN for 3G/4G Services
Service Type	L3VPN
End Point Connection Type	Dot1Q
Routing Protocol used to Connect the CE	Direct
QoS policies applied in the service	None

Table 14. L3VPN for 3G/4G Service required parameters

4.2.3.6 Description

In the **L3VPN for 3G/4G Services** the nodeB are directly connected to the cell site (HL5 layer). The cell site acts as a first aggregation layer, for nodes that share the same geographical location. The connections between the next Access Router layer (HL3) and cell site layer is made in a ring topology. The HL3 receives and aggregates traffic from the rings of the same state (geographical location). In the HL5 a L3VPN is created to receive the interfaces of each nodeB. Three interfaces are created in the HL5, each in a different IP Address and VLAN range:

- **PKI interface:** Used for PKI authentication.
- S1 Interface: Users Traffic
- Sync Interface: Clock signaling

The VPNID, targets and distinguisher are derived from standard naming conventions.

Additional logical resources such as VLANs, IP Addresses are defined by the network planning team during the service design. The Access Router acts as a Route Reflector for all the cell sites HL5 of the region. For that reason, specific filtering rules must be configured in the MP-BGP session.

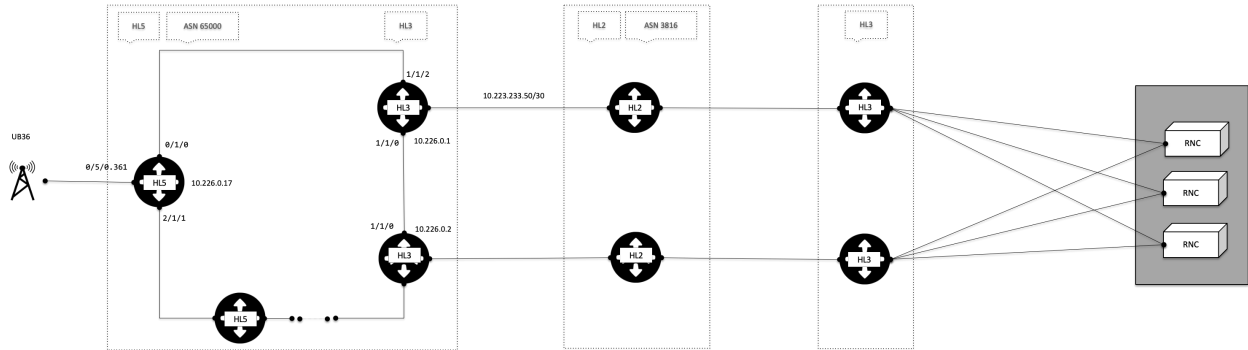


Figure 5. Service topology diagram

To deploy this service several conditions must be evaluated,

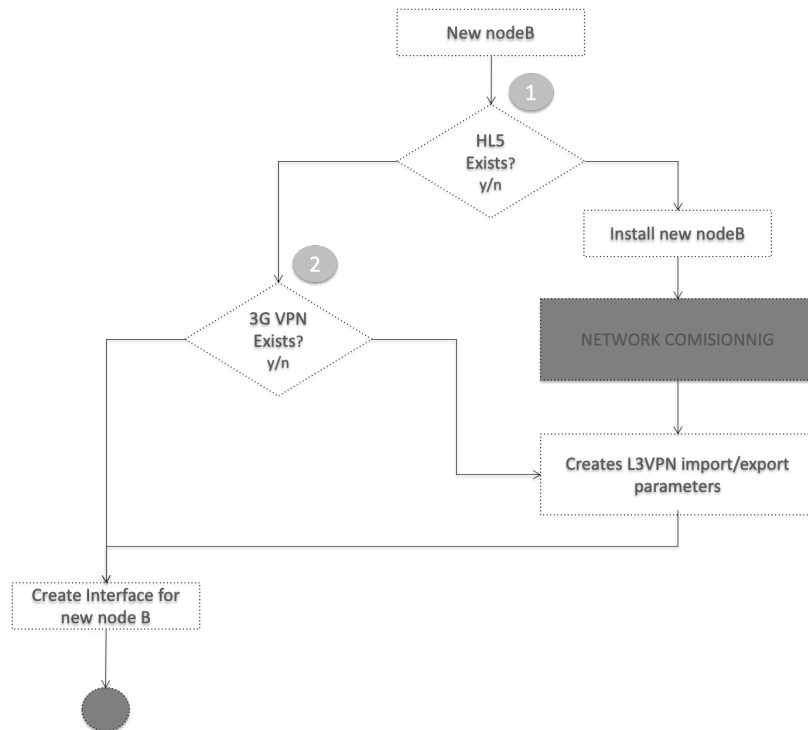


Figure 6 show the different scenarios:

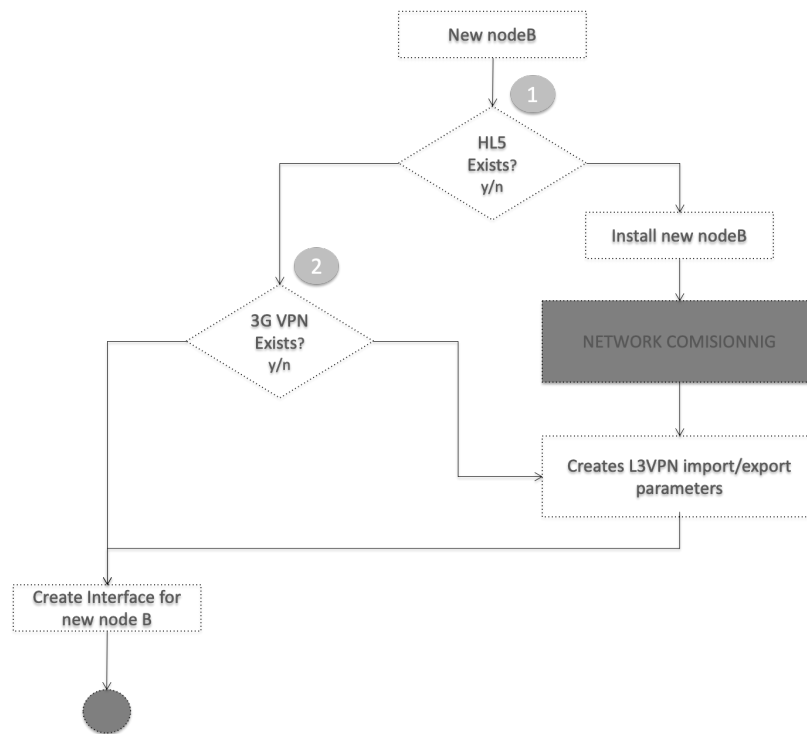


Figure 6. Service provision options

1. Is the HL5 cell site router already deployed in the network?
 - a. **No.** Some commissioning process may be done. This Includes Physical/Logical activation of the device, interfaces, and protocols.
 - b. **Yes.** Is the L3VPN already configured at the HL5 receiving the nodeB?
 - i. **No.** Full configuration may be done.
 - ii. **Yes.** Just the interface parameters must be included in the configuration.

4.2.3.7 Configuration example: SDTN NBI

For this specific use case the set of parameters and RESTCONF request messages are described in detail with an example of each parameter (Site and bearer CE side details are abstracted, considered as available as a pre-requisite):

4.2.3.7.1 Create VPN (VPN.L3.Add)

Parameter	Responsible	Service Example
VPN ID	NBI	550087400
Customer Name	NBI	4G_MUST
VPN service topology	NBI	any-to-any
Description	NBI	4G_MUST

Parameter	Responsible	Service Example
profile-id	NBI/NET	ie_00
role	NBI	any-to-any
Local-autonomous-system	NBI	65001
RD	NBI/NET	65100:87400023
Vpn-target/route-target	NBI/NET	65010:17400
Vpn-target/route-target-type	NBI/NET	both
Maximum routes	NBI/NET	-

Parameter	Responsible	Service Example
Vpn-node-ID	NBI/NET	550087400_10.226.0.23
NE-ID	NBI/NET	10.226.0.23
Description	NBI/NET	4G_MUST
Router-ID	NBI	10.226.0.23
active-profile	NBI	ie_00
Status/administrative	NBI	UP

4.2.3.7.2 Create Access (VPN.L3.Access.Add)

Parameter	Responsible	Service Example
vpn-network-access-id	NBI	PKI-MLB0832-550087400
port-id	NBI	1/4/6
description	NBI	S1-MLB0832-GUATEQUE CENTRO
vpn-network-access-type	NBI	Point-to-point
Status/administrative	NBI	UP
Connection/ Ethernet encapsulation	NBI	Vlan Dot1q 1017

Connection/l2-tunnel-service/ Pseudowire/vcid (if necessary)	NBI	-
Ip-connection/[ipv4/ipv6]/ [dhcp/static]	NBI	10.99.84.69/30
Service/Bandwidth	NBI	-
Service/QoS [standard profile]	NBI	-
Routing-protocols/ospf Address-family Area-address Metric MTU Security	NBI	-
Routing-protocols/bgp ASnumber address-family Neighbor Multihop Security	NBI	-
Routing-protocols/static/ipvX-prefixes/ lan next-hop lan-tag	NBI	None

5

Network Topology

Topology collection is a critical use case for every Operation because the network topology is an abstract representation of the physical nodes, links and network interconnections. It is crucial to get and graphically represent network information, such as ...



5 Network Topology

5.1 Introduction

Topology collection is a critical use case for every Operation because the network topology is an abstract representation of the physical nodes, links and network interconnections. It is crucial to get and graphically represent network information, such as:

- Structure (Connectivity and Paths)
- Performance (Available bandwidth per link)
- Availability of physical and logical resources

Currently, the topology representations are limited to the scope of each of the network vendors i.e. Each NMS has its particular/proprietary network view. Sometimes Dummy devices from a third party can be included to simulate the interconnection of the networks. However, nowadays obtaining a unified view of the entire IP network is not possible.

In MUST, the topology has been defined as a hierarchical structure composed of several layers. Each layer has its own schema and parameters. The topology represents not only physical but also logical components of the network. The set of network parameters of each layer is particularly defined based on their technological requirements (i.e., VLANs are only represented in the Layer 2 Topology). In this layered schema, there is no direct relationship between layers. The only interconnection between two layers uses the supported-on parameters. In this case, one layer can be supported in another (i.e., the upper layer is supported over the lower one as depicted in Figure 7).

Some concepts that must be considered are the following:

- **Network:** The data model contains a container with a list of networks. Each network is captured in its own list entry, distinguished via a network-id. A network has a certain type, such as L2, L3, OSPF, or IS-IS. A network can even have multiple types simultaneously. The Network contains an inventory of nodes that are part of

the network.

- **Supporting Network:** A network can in turn be part of a hierarchy of networks, building on top of other networks. Any such networks are captured in the list "supporting-network". A supporting network is, in effect, an underlay network.
- **Node:** The node represents an abstraction of the device for the network of which it is a part. The nodes of a network are captured in their own list. Each node is identified relative to its containing network by a node-id.
- **Supporting Node:** In cases where the same device or entity takes part in multiple networks or at multiple layers of a networking stack, the same device or entity will be represented by multiple nodes, one for each network. The relationship between those nodes is the supporting node.

These concepts are depicted in Figure 7. There are three networks (L3, L2 and L1). Each network has one node (a representation of the physical node-A). The upper layer L3 is supported on Layer 2 and consequently the Node A of the layer 3 is supported on the Node A of the layer 2.

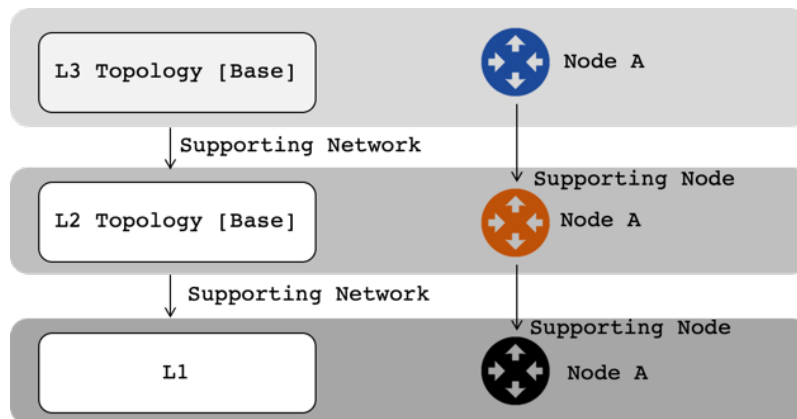


Figure 7. Supporting Network and Supporting Node

Based on this definition, within MUST, the network controller must export the topological information based on the following definitions:

- There must be at least four networks (topology levels).
- The controller must support the correlation between networks using the supported-on reference at network, node and termination point level.
 - **Network #1 (L1).** This network is the closest view of the “physical” representation of the network. L1 topology is composed by nodes, links and termination points. The termination points have the Interdomain reference.
 - **Network #2 (L2).** L2 is composed by nodes, links and termination points. It contains the Ethernet links and the router interfaces.
 - **Network #3 (L3).** The L3 topology contains the IP links, IGP links and the network interfaces. It is composed by nodes, links and termination points.
 - **Network #4 (UNI).** This network has the feasibility information.

Each layer and is described in detail in the following sections, covering the following set of use cases:

ID	Use case Title	Section
3.1	Obtain and export of end-to-end IP topology using IP domain controllers (IGP Topology)	Error! Reference source not found.Error! Reference source not found.
3.2	Obtain and export of L2 topology using IP domain controllers (ethernet links between routers)	5.6
3.3	Export potential service end points in IP topology	5.8

5.1.1 Use Case Features

Features

BASIC RETIEVAL

- Collect Networks, Nodes and Links from the network.
- Collect layered view of the network.
 - Physical
 - L2 – Ethernet
 - L3 – IP
 - TE – Tunnels

BASIC CREATION

- Networks Creation
- Physical properties creation on the base layer topology



5.1.2 Use case Operation

Operations	
READ	CREATE
GET	POST
<div>List all the networks.</div> <div>Lists the nodes of a Network.</div> <div>List the links of a Network</div> <div>List the Termination points of a Node</div> <div>List all the termination points of the network</div> <div>List all the Ethernet Links of a Node</div> <div>List all the IP links of a node</div> <div>List all the TE links of a Node</div> <div>List all the LSPs of a TE Tunnel.</div> <div>Get relationships between Ethernet, IP and TE Links.</div> <div>Get the LLDP neighbors information from a node.</div> <div>Get the IGP neighbors information from a node.</div> <div>Get the BGP neighbors information from a node.</div> <div>Get multicast attributes from a node.</div>	
<div>Assign a geo-location of a node.</div> <div>Create physical interconnections</div>	

5.2 Assumptions and Definitions

As described previously the network controller must be able to export the network topology in at least four layers. In this section, each of these layers, their minimal requirements, data tree and supporting reference are described.

5.2.1 Layer 1

This layer represents the closest view of the physical nodes and ports of the network and is the one who has connections with the Transport layer (FO/DWDM). The Layer 1 network topology YANG module is designed to be generic and applicable to different L1 technologies. It can be used to describe the physical with logical (attributes) for L1 network topologies. The main parameters in this Layer are the following:

Network: The name of the network must be “l1topology”. There is no supporting network

in this network view.

Network Type: The L1 Network uses the TE topo YANG module. Thus, the network type must be TE-type.

Node: Represents the physical routers of the network. There is no supporting node in this network view.

Termination-point: Represent the physical ports of the network used for interconnection between nodes and other devices in the network. There is no supporting termination point in this network view.

Inter-domain-plug-id: Binary number used to reference the connection between IP and Optical Domain.

Links: Links between domains.

The tree of the data model is described in **Figure 8. L1 Topology Tree**

```

module: ietf-network
  +--rw networks
    +--rw network* [network-id]
      +--rw network-id          network-id
      +--rw network-types
augment /nw:networks/nw:network/nw:network-types:
  +--rw te-topology!
  +--rw supporting-network* [network-ref]
    | +--rw network-ref      -> /networks/network/network-id
  +--rw node* [node-id]
    +--rw node-id          node-id
    +--rw supporting-node* [network-ref node-ref]
      +--rw network-ref    -> ../../../../supporting-
network/network-ref
      +--rw node-ref       -> /networks/network/node/node-id
augment /nw:networks/nw:network/nw:node:
  +--rw te-node-id?   te-types:te-node-id
  +--rw te!
    +--ro oper-status?          te-types:te-oper-
status
    +--ro geolocation
      | +--ro altitude?      int64
      | +--ro latitude?     geographic-coordinate-degree

```

```

|   +--ro longitude?    geographic-coordinate-degree
+--rw termination-point* [tp-id]
   +--rw tp-id          tp-id
   +--rw supporting-termination-point* [network-ref node-ref
tp-ref]
      +--rw network-ref    -> ../../../../nw:supporting-
node/network-ref
      +--rw node-ref       -> ../../../../nw:supporting-
node/node-ref
      +--rw tp-ref        ->
/nw:networks/network[nw:network-id=current()/../network-
ref]/node[nw:node-id=current()/../node-ref]/termination-point/tp-id
augment /nw:networks/nw:network/nw:node/nt:termination-point:
   +--rw te-tp-id?    te-types:te-tp-id
   +--rw te!
      +--rw admin-status?          te-types:te-
admin-status
      +--rw name?                  string
      +--rw inter-domain-plug-id?  binary
augment /nw:networks/nw:network:
   +--rw link* [link-id]
      +--rw link-id          link-id
      +--rw supporting-link* [network-ref link-ref]
      +--rw network-ref      -> ../../../../nw:supporting-
network/network-ref
      +--rw link-ref         -> /nw:networks/network[nw:network-
id=current()/../network-ref]/link/link-id

```

Figure 8. L1 Topology Tree

The draft that supports this network topology is [\[Reduced Draft-ietf-teas-yang-te-topo\]](#).

5.2.2 Layer 2

The L2 layer represents the Link-Layer or Ethernet connection. The Layer 2 network topology YANG module is designed to be generic and applicable to Layer 2 networks built with different L2 technologies. It can be used to describe both the physical and the logical (virtual) L2 network topologies. Parameters of this Layer are the following:

- MAC address information
- Maximum Transmission Unit (MTU)
- Link Aggregation (LAG)
- VLAN ID

- Pseudowires (PW)

The L2 Topology must have the following considerations to be constructed and exported:

- Intra-Interconnections (Connections between domains):
 - With another IP domain: **YES** (for example an ETHERNET LINKS between the Nokia domain and the Huawei domain)
 - With access layer (fixed DSLAMs or mobile eNODEBs): **NO**
 - With platforms (i.e. mobile core): **NO**
 - With clients: **NO**
- Supported connections
 - Ethernet links: **YES** with its basic characteristics (speed)
 - On what it is supported: **YES** (it would be the relationship with the lower layer)
 - What goes above: **NO**
 - Point-to-point links only. There is no L2 switching.
 - LAG: **YES**
 - VLAN: **YES** (understanding a link between two nodes that is a VLAN)
 - QinQ: **NO**
- Supported interfaces
 - Those associated with the base router only with its basic characteristics (the rest of parameters, to the inventory)
 - VLAN / tag for the case of VLAN
 - LAG Interface
 - Ethernet Links: **YES** with its basic characteristics.
 - On what it is supported: **YES** (it would be the relationship with the lower layer)
 - What goes above: **NO**
- Subscriptions to changes: **YES**

```
module: ietf-l2-topology
augment /nw:networks/nw:network/nw:network-types:
  +--rw l2-network!
augment /nw:networks/nw:network:
```

```

+--rw l2-network-attributes
  +--rw name?    string
augment /nw:networks/nw:network/nw:node:
+--rw l2-node-attributes
  +--rw name?          string
  +--rw description?   string
  +--rw management-address*  inet:ip-address
  +--rw sys-mac-address?  yang:mac-address
  +--rw management-vid?   dot1q-types:vlanid {VLAN}?
augment /nw:networks/nw:network/nt:link:
+--rw l2-link-attributes
  +--rw name?    string
  +--rw flag*    link-flag-type
  +--rw rate?    decimal64
  +--rw delay?   uint32
  +--rw srlg*    uint32
augment /nw:networks/nw:network/nw:node/nt:termination-point:
+--rw l2-termination-point-attributes
  +--rw description?          string
  +--rw maximum-frame-size?   uint32
  +--rw (l2-termination-point-type)?
    | +--:(ethernet)
    | | +--rw mac-address?      yang:mac-address
    | | +--rw eth-encapsulation? identityref
    | | +--rw lag?              boolean
    | | +--rw member-link-tp*   leafref
    | | +--rw mode?             neg-mode
    | | +--rw port-vlan-id?     dot1q-types:vlanid {VLAN}?
    | | +--rw vlan-id-name* [vlan-id] {VLAN}?
    | | | +--rw vlan-id        dot1q-types:vlanid
    | | | +--rw vlan-name?     string
    | | +--rw qinq* [svlan-id cvlan-id] {QinQ}?
    | | | +--rw svlan-id       dot1q-types:vlanid
    | | | +--rw cvlan-id       dot1q-types:vlanid
    | | +--rw vxlan {VXLAN}?
    | | | +--rw vni-id?       vni
    | +--:(legacy)
    | | +--rw layer-2-address?  yang:phys-address
    | | +--rw encapsulation?    identityref
  +--ro tp-state?              enumeration

```

Table 15. L2 Topology Tree

The draft that supports this network topology is [\[draft-ietf-i2rs-yang-l2-network-topology\]](#).

5.2.3 Layer 3 & IGP

The L3 layer represents the Network layer and IGP layer. The L3 Topology must have the following considerations to be constructed and exported:

- Intra-Interconnections:
 - With another IP domain: **YES** (for example a ETHERNET LINKS between the Nokia domain and the Huawei domain)
 - With access layer (fixed DSLAMs or mobile eNODEBs): **NO**
 - With platforms (i.e. mobile core): **NO**
 - With clients: **NO**
- IGP Parameters
 - IGP Protocol: OSPF and IS-IS
 - Area ID: **YES**
 - Router Role: **YES**
 - IGP authentication: **NO**
 - Adjacencies: **NO**
- Supported connections
 - IP links: **YES** with its basic characteristics (metric, bandwidth)
 - On what it is supported: **YES** (it would be the relationship with the lower layer)
 - What goes above: **NO**
- Supported interfaces
 - Those interfaces and sub-interfaces associated with the base router only with its basic characteristics (the rest of parameters are part of the inventory)
- Subscriptions to changes: **YES**

```
module: ietf-l3-unicast-topology
augment /nw:networks/nw:network/nw:network-types:
  +--rw l3-unicast-topology!
augment /nw:networks/nw:network:
  +--rw l3-topology-attributes
    +--rw name?    string
    +--rw flag*    l3-flag-type
augment /nw:networks/nw:network/nw:node:
  +--rw l3-node-attributes
    +--rw name?    inet:domain-name
```

```

    +--rw flag*          node-flag-type
    +--rw router-id*     rt-types:router-id
    +--rw prefix* [prefix]
        +--rw prefix     inet:ip-prefix
        +--rw metric?    uint32
        +--rw flag*      prefix-flag-type
augment /nw:networks/nw:network/nt:link:
    +--rw l3-link-attributes
        +--rw name?      string
        +--rw flag*      link-flag-type
        +--rw metric1?   uint64
        +--rw metric2?   uint64
augment /nw:networks/nw:network/nw:node/nt:termination-point:
    +--rw l3-termination-point-attributes
    +--rw (termination-point-type)?
        +--:(ip)
        |   +--rw ip-address*      inet:ip-address
        +--:(unnumbered)
        |   +--rw unnumbered-id?   uint32
        +--:(interface-name)
        +--rw interface-name?     string

```

L3 Topology Tree

The draft that supports this network topology is [\[RFC8345\]](#) [\[RFC8346\]](#).

5.2.4 UNI Topology

The Represents an abstracted view of the Service Provider network. Contains the points from which VPN services can be attached. This layer should contain all the termination points (used and unused) of the network.

The parameters of this Layer are the following:

- Nodes
- Service Attachment Points (User Network Interface)

The data model proposed augments ietf-network model by adding the concept of service-attachment-points. The service-attachment-points are an abstraction of the points to which network services such as L3 VPNs or L2 VPNs can be attached. The service orchestration layer does not need to know about the internals of the network. Hence, the abstraction need is to be able to get the set of nodes, and the attachment points associated with the nodes from which network services can be requested.

The set of parameters defined the UNI-topology are:

```
module: ietf-uni-topology
  augment /nw:networks/nw:network/nw:network-types:
    +--rw uni-topology!
  augment /nw:networks/nw:network/nw:node:
    +--rw service-attachment-points
      +--rw service-attachment-point* [attachment-id]
        +--rw attachment-id          nt:tp-id
        +--rw name?                  string
        +--ro type?                  identityref
        +--rw admin-status?          boolean
        +--rw oper-status?           enumeration
        +--rw encapsulation-type?    Identityref
```

Table 16. UNI Topology Tree

The draft that supports this network topology is [\[draft-ogondio-opsawg-uni-topology-00\]](#).

5.3 Model Parameters

The topology representation of the Network in this document is supported on the RFC8345. In this RFC a basic network representation is done using an abstract network model ("ietf-network") with the set of parameters described in this section. These parameters are common for all the networks described in this document (L1, L2, L3, and UNI) because all the topology representations augmented this basic module as depicted in figure 27 (the dotted lines represent augmentations of the model). The specific parameters of each network are described in detail after in the document.

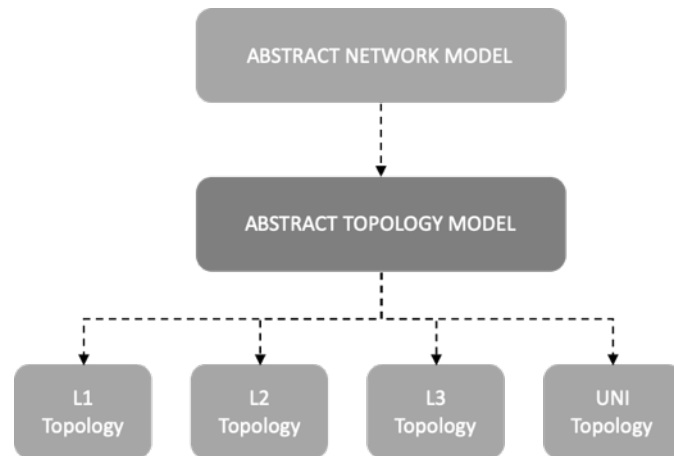


Figure 9. IETF Network usage

- **Network:** It's the main container and it groups all the topology elements (Nodes, TP and Links).

Attribute	Allowed Values/Format	Format	Responsible	Mod
network-id	value: "string"	UUID	Controller	RW
network-types	value: "string"	'l1topology l2topology l3topology uni'	Controller	RW
node	List of {node}	-	-	RW

Table 17. Network Topology Network required parameters.

- **Nodes:** Represents all the physical network elements deployed. It has all the possible termination points.

Attribute	Allowed Values/Format	Format	Responsible	Mod
node-id	value: "string"	NE_ID	Network	RW
termination-point	List of {termination-point}	-	Network	RW

Table 18. Network Topology Node required parameters.

- **Termination Points:** All the ports or L3 Interfaces used for interconnection between nodes in the network.

Attribute	Allowed Values/Format	Format	Responsible	Mod
tp-id	value: "string"	Port-ID	Controller	RW
supporting-termination-point	List: [network-ref node-ref tp-ref]	-	Controller.	RW

Table 19. Network Topology Termination Points required parameters.

- **Links:** Interconnection between nodes.

Attribute	Allowed Values/Format	Format	Responsible	Mod
link-id	value: "string"	"Node_Src + Port_Src" -- "Node_Dst + Port_Dst"	Controller	RW
source	Leaf of {source-node, source-tp}	NE_ID,Port ID	Controller	RW
destination	Leaf of {dest-node, dest-tp}	NE_ID,Port ID	Controller	RW

Table 20. Network Topology Link required parameters.

5.4 Operations

The operations allowed in the RFC are the following:

5.4.1 Retrieve all networks:

```
GET /restconf/data/ietf-network:networks/
HTTP/1.1
Host: example.com
Accept: application/yang-data+json
```

5.4.2 Retrieve one specific network:

```
GET /restconf/data/ietf-  
network:networks/network={NetID} HTTP/1.1  
Host: example.com  
Accept: application/yang-data+json
```

5.4.3 Retrieve one specific node:

```
GET /restconf/data/ietf-  
network:networks/network={NetID}/node={NodeId}  
HTTP/1.1  
Host: example.com  
Accept: application/yang-data+json
```

5.4.4 Retrieve one Specific Termination Point:

```
GET /restconf/data/ietf-  
network:networks/network={NetID}/node={NodeId}/nt:terminatio  
n-point={Tp-Id} HTTP/1.1  
Host: example.com  
Accept: application/yang-data+json
```

5.4.5 Retrieve one Links:

```
GET /restconf/data/ietf-  
network:networks/network={NetID}/nt:link={link-  
id} HTTP/1.1  
Host: example.com  
Accept: application/yang-data+json
```

5.4.6 Retrieve destination of a Link:

```
GET /restconf/data/ietf-  
network:networks/network={network-  
id}/nt:link={link-id}/nt:destination/ HTTP/1.1  
Host: example.com  
Accept: application/yang-data+json
```

5.4.7 Retrieve source of a Link:

```
GET /restconf/data/ietf-  
network:networks/network={network-  
id}/nt:link={link-id}/nt:source/ HTTP/1.1  
Host: example.com  
Accept: application/yang-data+json
```



5.5 Workflow

The workflow of topology collection as well as the permitted operations is depicted in **Figure 10**. Topology Discovery Workflow. The first operation (1) retrieves the list of Network references included in the `/restconf/data/ietf-network:network` (2). For each Network reference found, operation (3) is repeated to obtain a Topology object instance filtered subtree up to one level (4).

For each node found in operation (4), operation (5) is repeated to retrieve node object (6) and its Terminations-points.

For each Network reference discovered in operation (2), operation (7) is repeated to retrieve each `/restconf/data/ietf-network:network={NET_ID}/ietf-network-topology:link={LINK_ID}` object (8).

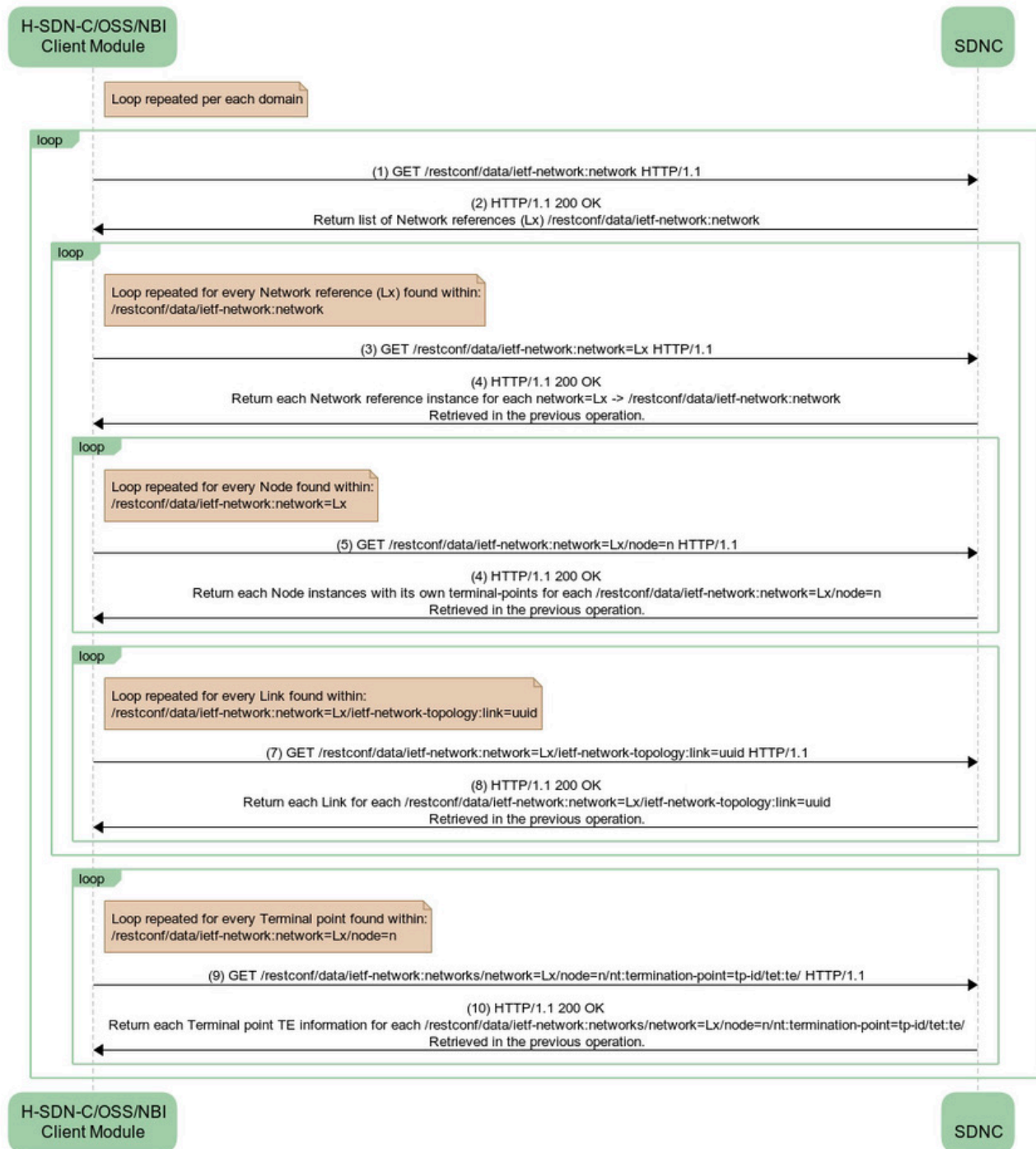


Figure 10. Topology Discovery Workflow

5.6 Obtain and export of L2 topology using IP domain controllers (ethernet links between routers)

In this section, the details to Obtain and export of L2 topology using IP domain controllers (ethernet links between routers)

Number	C.9
Name	Obtain and export of L2 topology using IP domain controllers (ethernet links between routers)
Brief description	L2 Topology must represent the Ethernet links in the network. Including the Nodes, the termination points and the links. This topology uses the basic topology representation of the IETF but augmented with the specific parameters to support VLANs, LAGs and Ethernet link information.

5.6.1 L2 Parameters

Some augmentations were done to cover all the link-layer parameters. The set of augmentation of each Topology elements is the following:

- Network

Attribute	Allowed Values/Format	Format	Responsible	Mod
Network-id	value: "string"= L2Topology	l2topology'	Controller	RW

Table 21. L2 Topology Network required parameters.

- Nodes

Attribute	Allowed Values/Format	Format	Responsible	Mod
Node-id	value: "string"= NE-ID	NE-ID (ipaddress)	Controller	RW
l2-node-attributes/name	value: "string"=NE-Name	NE-Name (hostname)	Controller	RW
management-address	value: "string" = NE-IP address		Network	RO
sys-mac-address	value: "string"		Network	RO
management-vid	value: "uint16" : range "0..4095"		Network	RO

Table 22. L2 Topology Node required parameters.

- Termination Points

Attribute	Allowed Values/Format	Format	Responsible	Mod
description	value: "string"= Port Description		Network	RW
maximum-frame-size	value: "uint32"		Network	RW
l2-termination-point-type	"value": "ethernet" "legacy"		Network	RW
tp-state	value:"uint16" - 0: "the termination point is in forwarding state"		Network	RO
	value:"uint16" - 1: "the termination point is in blocking state"		Network	
	value:"uint16" - 2: "the termination point is in down state"		Network	
	value:"uint16" - 3: "the termination point is in other state"		Network	

Table 23. L2 Topology Termination Points required parameters.

- Termination Points: Type = Ethernet

Attribute	Allowed Values/Format	Format	Responsible	Mod
mac-address	"value": "string"		Network	RW
eth-encapsulation	Type = "value": "string"		Network	RW
lag	Type = "value": "boolean"	Yes/No	Controller	RW
member-link-tp	Type = "value": "string"		Network	RW
mode	neg-mode		Network	RW
port-vlan-id	"value": "uint16" : range "0..4095"		Network	RW
qinq	List = [svlan-id "value": "uint16" : range "0..4095",		Network	RW

	cvlan-id "value": "uint16" : range "0..4095"]			
--	---	--	--	--

Table 24. L2 Topology Termination Points Ethernet required parameters.

- Termination Points:Type= legacy

Attribute	Allowed Values/Format	Format	Responsible	Mod
mac-address	"value": "string" : '[0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){5}'		Network	RW
eth-encapsulation	"value": "string"		Network	RW

Table 25. L2 Topology Termination Points Legacy required parameters.

- Links

Attribute	Allowed Values/Format	Format	Responsible	Mod
name	"value": "string"	"Node_Src + Port_Src" -- "Node_Dst + Port_Dst"	Controller	RW
rate	"value": "decimal64" : range "1 .. 3.14 10 20..max"		Network	RW
delay	"value": "uint32"		Network	RW

Table 26. L2 Topology Links required parameters.

5.6.2 Operations

Network and **Node** information can be retrieved in the same way as explained in section **Error! Reference source not found.** The additional operation supported to get specific L2 Information are:

5.6.2.1 Retrieve supporting L2 network attributes:

```
GET /restconf/data/ietf-
network:network/network={NetId}/l2t:l2-network-
attributes/ HTTP/1.1
Host: example.com
Accept: application/yang-data+json
```

5.6.2.2 Retrieve supporting L2 link attributes:

```
GET /restconf/data/ietf-
network:networks/network={NetId}/nt:link={link-
id}/l2t:l2-link-attributes/ HTTP/1.1
Host: example.com
Accept: application/yang-data+json
```

5.6.2.3 Retrieve One Specific Termination Point with L2 Attributes:

```
GET /restconf/data/ietf-networks
:networks/network={NetId}/node={node-
id}/nt:termination-point={tp-id}/l2t:l2-
termination-point-attributes/ HTTP/1.1
Host: example.com
Accept: application/yang-data+json
```

5.7 Obtain and export of end-to-end IP topology using IP domain controllers (IGP Topology)

In this section, the details to obtain and export of L3 topology using IP domain controllers (IP links between routers).

Number	C.8
Name	Obtain and export of end-to-end IP topology using IP domain controllers (IGP Topology)
Brief description	L3 topology must represent the IP links in the network. Including the Nodes, the termination points and the links. This topology uses the basic topology representation of the IETF but augmented with the specific parameters to support IP Addressing, Metrics and IGP information. The L3 (nodes and Links) are supported in L2 topology (nodes and links)-

5.7.1 L3 Parameters

Some augmentations were done to cover all the link-layer parameters. The set of augmentation of each topology elements is the following:

- Network

Attribute	Allowed Values/Format	Format	Responsible	Mod
name	value: "string" = L3Topology	l3topology'	Controller	RW
area-id	value: "uint32"		Controller	RW

Table 27. L3 Topology Network required parameters.

- Nodes

Attribute	Allowed Values/Format	Format	Responsible	Mod
name	value: "string" = NE-ID		Controller	RW
router-id	value: "NE-IP Address" '[0-255],[0-255],[0-255],[0-255]'		Network	RW
Supporting-node	Node-id	Leafref to the L2 Supporting Node.	Controller	RW

Table 28. L3 Topology Nodes required parameters.

- Termination Points

Attribute	Allowed Values/Format	Format	Responsible	Mod
termination-point-type	value: "string" = "ip","unnumbered","interface-name"		Controller	RW
Supporting-termination-point	Tp-id	Leafref to the L2 Supporting termination point.	Controller	RW

Table 29. L3 Topology Termination Points required parameters.

- Termination Points:Type= ip

Attribute	Allowed Values/Format	Format	Responsible	Mod
-----------	-----------------------	--------	-------------	-----

ip-address	value: "string" '[0-255],[0-255],[0-255],[0-255]'		Network	RW
------------	---	--	---------	----

Table 30. L3 Topology Termination Points type IP required parameters.

- Type= unnumbered

Attribute	Allowed Values/Format	Format	Responsible	Mod
unnumbered-id	value: "uint32"		Network	RW

Table 31. L3 Topology Termination Points type Unnumbered required parameters.

- Type= interface-name

Attribute	Allowed Values/Format	Format	Responsible	Mod
interface-name	value: "string"		Network	RW

Table 32. L3 Topology Termination Points type Interface-name required parameters.

- Links

Attribute	Allowed Values/Format	Format	Responsible	Mod
name	value: "string"	"Node_Src + Port_Src" -- "Node_Dst + Port_Dst"	Controller	RW
metric1	value: "uint64"		Network	RW
metric2	value: "uint64"		Network	RW

Table 33. L3 Topology Link required parameters.

5.7.2 Operations

The topology discover use case consists on the following operation and it is important to indicate that **Network** and **Node** information can be retrieved in the same way as explained in **section** Error! Reference source not found.:

5.7.3 Retrieve L3 unicast network attributes:



```
GET /restconf/data/ietf-  
network:networks/network={network-id}/network-  
types/l3t:l3-unicast-topology/ HTTP/1.1  
Host: example.com  
Accept: application/yang-data+json
```

5.7.4 Retrieve supporting L3 network attributes:

```
GET /restconf/data/ietf-networks:networks  
/network={network-id}/l3tp:l3-topology-attributes/  
HTTP/1.1  
Host: example.com  
Accept: application/yang-data+json
```

5.7.5 Retrieve supporting L3 link attributes:

```
GET /restconf/data/ietf-networks:networks  
/network={NetId}/nt:link={link-id}/l3tp:l3-link-  
attributes/ HTTP/1.1  
Host: example.com  
Accept: application/yang-data+json
```

5.7.6 Retrieve One Node with L3 attributes:

```
GET /restconf/data/ietf-  
network:networks/network={network-  
id}/node={node-id}/l3tp:l3-node-attributes/  
HTTP/1.1  
Host: example.com  
Accept: application/yang-data+json
```

5.7.7 Retrieve One Specific Termination Point with L3 attributes:

```
GET /restconf/data/ietf-  
network:networks/network={network-  
id}/node={node-id}/nt:termination-point={tp-
```

```
id}/l3tp:l3-termination-point-attributes/
HTTP/1.1
Host: example.com
Accept: application/yang-data+json
```

5.8 Export potential service end points in IP topology (UNI Topology)

In this section, the details to obtain and export potential end points in the network are the following:

Number	C.10
Name	Export potential service end points in IP topology (UNI Topology)
Brief description	The UNI topology represents the feasibility topology and has all the potential end point in the network.

5.8.1 UNI Parameters

The UNI network view must have a feasibility view of the network, for example:
The set of model parameters are the following:

- Network

Attribute	Allowed Values/Format	Format	Responsible	Mod
name	value: "string" = L3Topology	Uni-topology	Controller	RW
Network-type	Value:"string"	"uni-topology"	Controller	RW

Table 34. Service End Points UNI Network required parameters.

- Nodes

Attribute	Allowed Values/Format	Format	Responsible	Mod
name	value: "string" = NE-ID		Controller	RW
Supporting-node	Node-id	Leafref to the L2 Supporting Node.	Controller	RW

Table 35. Service End Points UNI Node required parameters.

- Service Access Points

Attribute	Allowed Values/Format	Format	Responsible	Mod
attachment-id	nt:tp-id		Controller	RW
name	string		Controller	RW
type?	identityref		Controller	RW
admin-status?	boolean		Controller	RW
oper-status?	enumeration		Controller	RW
encapsulation-type?	identityref		Controller	RW

Table 36. Service End Points UNI Service Access Points required parameters.

5.8.2 UNI-Operations

The model permits the following operations:

5.8.2.1 Retrieve all attachment points from a specific node:

```
GET /restconf/data/networks/ietf-
networks:network={network-id}/node={node-
id}/uni:service-attachment-points/ HTTP/1.1
Host: example.com
Accept: application/yang-data+json
```

5.8.2.2 Retrieve details from an attachment point:

```
GET /restconf/data/ietf-
network:networks/network={network-
id}/node={node-id}/uni:service-attachment-
points/uni:service-attachment-
point={attachment-id}/ HTTP/1.1
Host: example.com
Accept: application/yang-data+json
```

5.9 Topology Examples

In this section some possible scenarios are described.

5.9.1 Back-to-Back Routers Interconnection

This is the simplest scenario where two routers are connected back-to-back using FO or DWDM as depicted in Figure 11. Back-to-Back Routers .

In this scenario the requirements are:

- Each router has a dedicated port.
- Logically there is not encapsulation set in the port.
- Interfaces in the router can have an IP address assigned (mandatory to describe IP topology)
- LLDP can be or not be enabled in the interface
- STP is disabled in the interfaces.
- Ports are administrative UP and Link is established between the nodes.

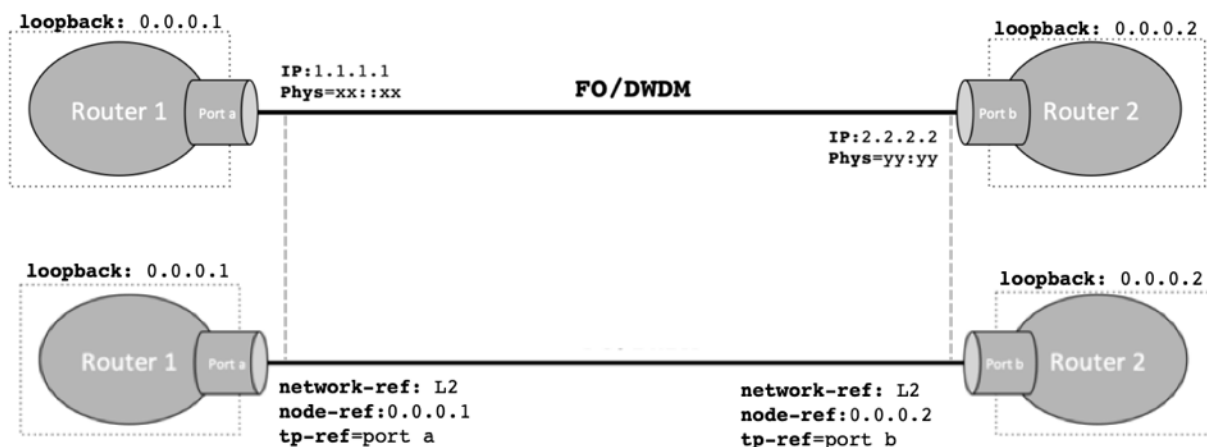


Figure 11. Back-to-Back Routers

The expected L2 Topology of this Network is:

Nodes		Termination Points		Termination Points		Links	
Values		Values		Values		Values	
Name	Router 1	tp-id	a	tp-id	b	link-id	Router 1,a,Router 2,b
management-address	0.0.0.1	description	Port a	description	Port b	name	Link a to b
Name	Router 2	maximum-frame-size	1518	maximum-frame-size	1518	source	"source-node": "Router 1", "source-tp": "a"
management-address	0.0.0.2	l2-termination-point-type	ethernet	l2-termination-point-type	ethernet	destination	"dest-node": "Router 2", "dest-tp": "b"
		tp-state	0	tp-state	0	rate	1000
		mac-address	xx:xx	mac-address	yy:yy	delay	12
		eth-encapsulation	none	eth-encapsulation	none		
		lag	FALSE	lag	FALSE		
		mode	auto-neg	mode	auto-neg		

The expected L3 Topology of this Network is:

Nodes		Termination Points		Termination Points		Links	
Values		Values		Values		Values	
Name	Router 1	tp-id	a	tp-id	b	link-id	Router 1,a,Router 2,b
router-id	0.0.0.1	ip-address	1.1.1.1	ip-address	2.2.2.2	name	Link a to b
Name	Router 2	supporting-termination-point	network-ref: L2	supporting-termination-point	network-ref: L2	source	"source-node": "Router 1", "source-tp": "a"
router-id	0.0.0.2		node-ref: 0.0.0.1		node-ref: 0.0.0.2	destination	"dest-node": "Router 2", "dest-tp": "b"
			tp-ref: Port a		tp-ref: Port a	metric1	100

5.9.2 Back-to-Back Routers VLAN Interconnection

In this scenario two routers are connected back-to-back using FO or DWDM and Virtual Interfaces (VLAN 1 and VLAN 2) are deployed between the nodes as depicted in Figure 12.

In this scenario the requirements are:

- Each router has a dedicated port.
- Logically is an encapsulation set in the port (dot1q or qinq).
- Virtual Interfaces in the router can have an IP address assigned (mandatory to describe IP topology)
- LLDP can be or not be enabled in the interface
- STP is disabled in the interfaces.
- Ports are administrative UP and Link is established between the nodes.

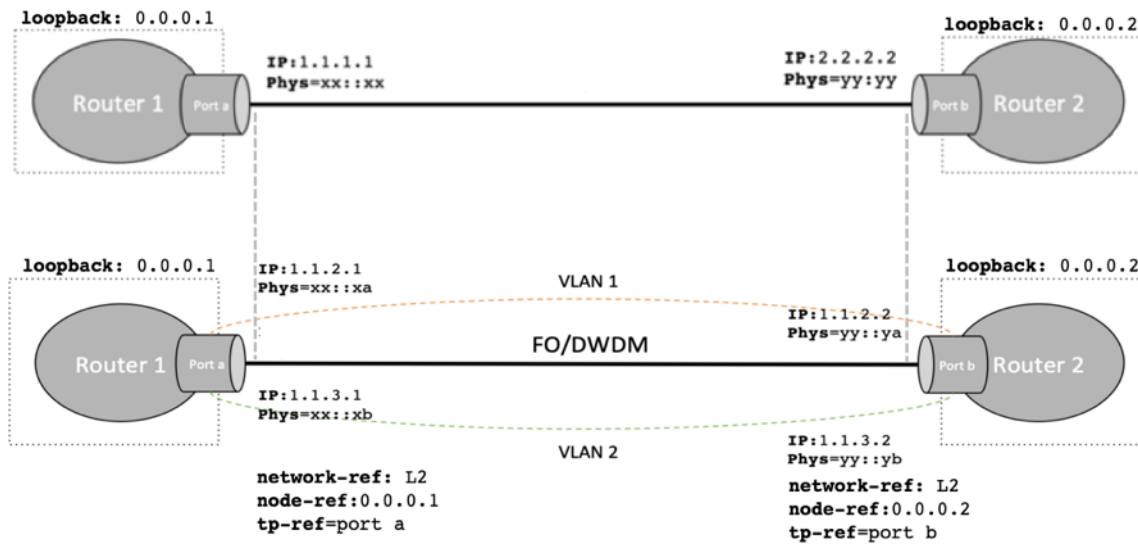


Figure 12. Back-to-Back Routers connected using VLANs

The expected L2 Topology of this Network is:

Nodes		Termination Points		Termination Points		Links	
Values		Values		Values		Values	
Name	Router 1	tp-id	a	tp-id	b	link-id	Router 1,a,Router 2,b
management-address	0.0.0.1	description	Port a	description	Port b	name	Link a to b
Name	Router 2	maximum-frame-size	1518	maximum-frame-size	1518	source	"source-node": "Router 1", "source-tp": "a"
management-address	0.0.0.2	l2-termination-point-type	ethernet	l2-termination-point-type	ethernet	destination	"dest-node": "Router 2", "dest-tp": "b"
		tp-state	0	tp-state	0	rate	1000
		mac-address	xx:xx	mac-address	yy:yy	delay	12
		eth-encapsulation	none	eth-encapsulation	none		
		lag	FALSE	lag	FALSE		
		mode	auto-neg	mode	auto-neg		
		tp-id	VLAN 1	tp-id	VLAN 1	link-id	Router 1,a,vlan 1,Router 2,b,vlan 1
		description	Port a VLAN 1	description	Port b VLAN 1	name	Link a to b VLAN 1
		mac-address	xx:xa	mac-address	yy:ya	source	"source-node": "Router 1", "source-tp": "a-vlan 1"
		eth-encapsulation	dot1q	eth-encapsulation	dot1q	destination	"dest-node": "Router 2", "dest-tp": "b-vlan 1"
		port-vlan-id	1	port-vlan-id	1		
		tp-id	VLAN 2	tp-id	VLAN 2	link-id	Router 1 a,vlan 2,Router 2 b,vlan 2
		description	Port a VLAN 2	description	Port b VLAN 2	name	Link a to b VLAN 2
		mac-address	xx:xb	mac-address	yy:yb	source	"source-node": "Router 1", "source-tp": "a-vlan 2"
		eth-encapsulation	dot1q	eth-encapsulation	dot1q	destination	"dest-node": "Router 2", "dest-tp": "b-vlan 2"
		port-vlan-id	2	port-vlan-id	2		

The expected L3 Topology of this Network is:

Nodes		Termination Points		Termination Points		Links	
Values		Values		Values		Values	
Name	Router 1	tp-id	a	tp-id	b	link-id	Router 1,a,Router 2,b
router-id	0.0.0.1	ip-address	1.1.1.1	ip-address	2.2.2.2	name	Link a to b
Name	Router 2	supporting-termination-point	network-ref: L2	supporting-termination-point	network-ref: L2	source	"source-node": "Router 1", "source-tp": "a"
router-id	0.0.0.2		node-ref: 0.0.0.1		node-ref: 0.0.0.2	destination	"dest-node": "Router 2", "dest-tp": "b"
			tp-ref: Port a		tp-ref: Port b	metric1	100

tp-id	VLAN 1	tp-id	VLAN 1	link-id	Router 1,a,vlan 1,Router 2,b,vlan 1
ip-address	1.1.2.1	ip-address	1.1.2.2	name	Link a to b VLAN 1
supporting-termination-point	network-ref: L2	supporting-termination-point	network-ref: L2	source	"source-node": "Router 1", "source-tp": "a-vlan 1"
	node-ref: 0.0.0.1		node-ref: 0.0.0.2	destination	"dest-node": "Router 2", "dest-tp": "b-vlan 1"
	tp-ref: Port a VLAN 1		tp-ref: Port b VLAN 1	metric1	100

tp-id	VLAN 2	tp-id	VLAN 2	link-id	Router 1,a,vlan 2,Router 2,b,vlan 2
ip-address	1.1.3.1	ip-address	1.1.3.2	name	Link a to b VLAN 2
supporting-termination-point	network-ref: L2	supporting-termination-point	network-ref: L2	source	"source-node": "Router 1", "source-tp": "a-vlan 2"
	node-ref: 0.0.0.1		node-ref: 0.0.0.2	destination	"dest-node": "Router 2", "dest-tp": "b-vlan 2"
	tp-ref: Port a VLAN 2		tp-ref: Port b VLAN 2	metric1	100

5.9.3 Back-to-Back Routers Interconnection with OSPF

This is the simplest scenario where two routers are connected back-to-back using FO or DWDM as depicted in Figure 13.

In this scenario the requirements are:

- Each router has a dedicated port.
- Logically there is not encapsulation set in the port.
- Interfaces in the router can have an IP address assigned (mandatory to describe IP topology)
- LLDP can be or not be enabled in the interface
- STP is disabled in the interfaces.
- Ports are administrative UP and Link is established between the nodes.
- OSPF protocol is configured.

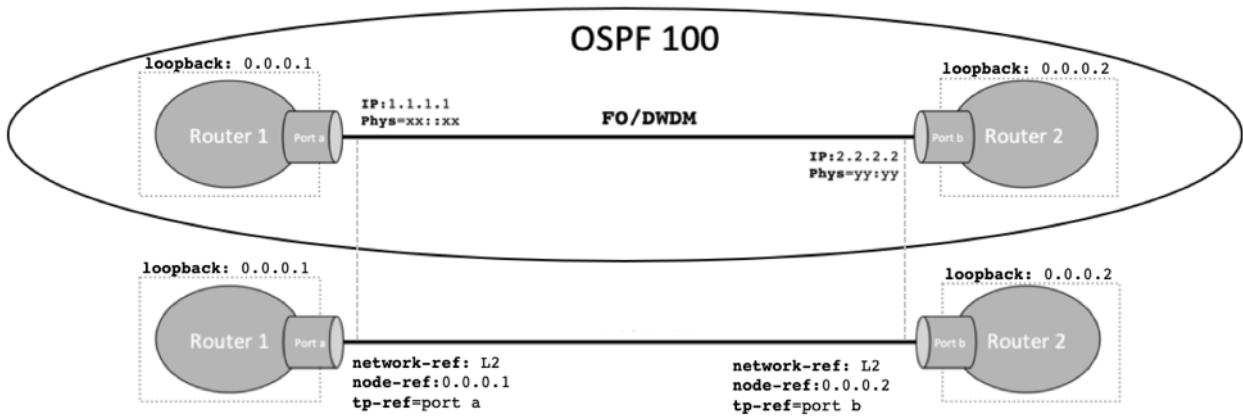


Figure 13. Back-to-Back Routers interconnection with OSPF

The expected L2 Topology of this Network is:

Nodes		Termination Points		Termination Points		Links	
Values		Values		Values		Values	
Name	Router 1	tp-id	a	tp-id	b	link-id	Router 1,a,Router 2,b
management-address	0.0.0.1	description	Port a	description	Port b	name	Link a to b
Name	Router 2	maximum-frame-size	1518	maximum-frame-size	1518	source	"source-node": "Router 1", "source-tp": "a"
management-address	0.0.0.2	l2-termination-point-type	ethernet	l2-termination-point-type	ethernet	destination	"dest-node": "Router 2", "dest-tp": "b"
		tp-state	0	tp-state	0	rate	1000
		mac-address	xx:xx	mac-address	yy:yy	delay	12
		eth-encapsulation	none	eth-encapsulation	none		
		lag	FALSE	lag	FALSE		
		mode	auto-neg	mode	auto-neg		

6

Network Inventory

Inventory management in a Service Provider Environment is critical. Networks are in a constant state of change because events, occurring at any time, change the status of the network, impact the future topology of the network, and affect operations and business processes. SDN offers complementary functions to inventory systems by providing real time inventory information to the OSS layer...



6 Network Inventory

Inventory management in a Service Provider Environment is critical. Networks are in a constant state of change because events, occurring at any time, change the status of the network, impact the future topology of the network, and affect operations and business processes. SDN offers complementary functions to inventory systems by providing real time inventory information to the OSS layer. Inventory information can be classified by the following categories:

- **Topology:** Topology inventory is the topmost element in the hierarchy. The topology model defines “networks” and “nodes”. The topology model is used as a base to retrieve other information such as hardware inventory parameters. The topology use cases were presented in the previous section
- **Hardware Inventory:** Hardware inventory refers to the physical components of each node. These components might be line cards, transceivers, ports etc.
- **Logical Inventory:** Logical inventory refers to layer 3 and layer 2

6.1 Need of NE management model

In the case of network inventory retrieval, the network controller would not include any abstraction. The inventory requires the use of device-oriented Yang data models. However, at the time of writing, there is no IETF standard covering fully the needs of the management of device information from the controller perspective.

The detected gap to fulfil the needs of the Inventory is covered by augmenting the Network Topology and using the schema-mount definition done in RFC8528. All the device parameters will be retrieved from the NE devices by the SDN controller and exposed using the network topology model. Thus, the network topology model (IETF 8345) is augmented with mount points to include the logical and physical parameters of the device.

For that purpose, the TIP MUST team has defined the necessary augmentations and the mount points. The outcome will be taken to the relevant standard body. The code is

documented in annex A.

Mount-Points: YANG Schema Mount identifies mount points by name within a module. This definition allows for the definition of mount points whose schema can be shared across data models. As OpenConfig has independent definitions for the logical (interfaces, network instances, VLANs) and physical (system, platform), several mount points can be required for the whole inventory support. The set of mount points available in the current specification are the following:

Mount-Point	Data model
Interfaces-root	openconfig-interfaces

Network Topology Augmentations: Each mount point requires a specific augmentation to be added by the network controller to the network topology. The augmentations were defined at the IETF network topology node level. The following is an example type-specific augmentation:

```
augment /nw:networks/nw:network/nw:node {
  description
    "Augment used to define attach the node configuration";
  container commissioning-configs {
    uses system-config;
    uses protocols-config;
    uses resources-config;
    uses interfaces-config;
  }
}
```

```
module: ietf-network
+--rw networks
  +--rw network* [network-id]
    +--rw network-id          network-id
    +--rw node* [node-id]
      | +--rw node-id          node-id
      | | +--rw ne-mgmt-nm:commissioning-configs
      | | +---+
      |
```

The model hierarchy is represented in Figure 14 while the “tip-ne-commissioning” model tree is shown on the next section.

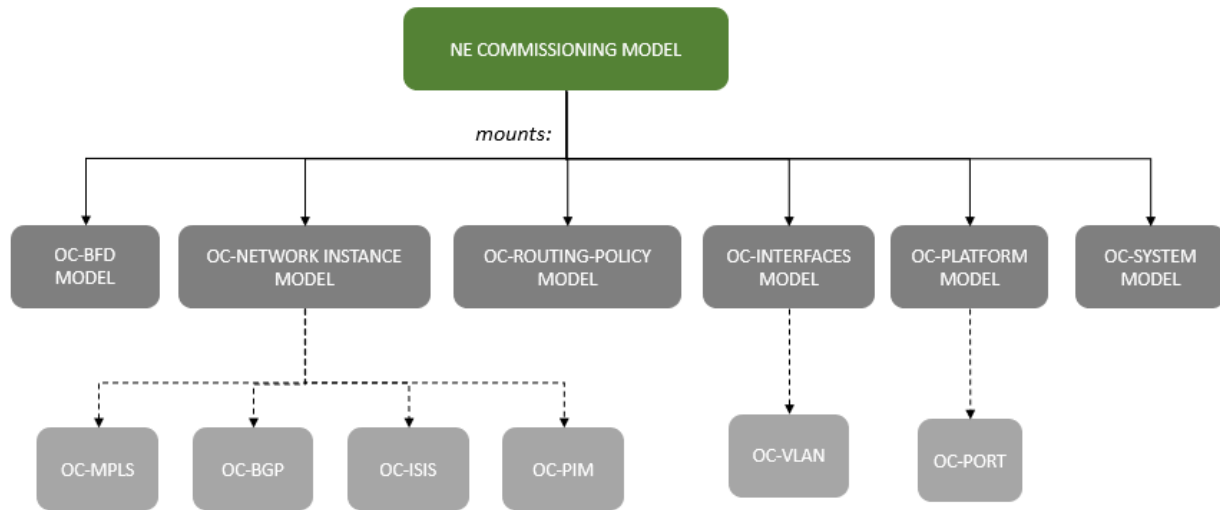


Figure 14. Model hierarchy for Inventory

6.1.1 Logical inventory

Logical inventory refers to the set of non-hardware resources which are present on the devices. Configuration inventory should be synchronized as well as pure physical resources so that network information databases are up to date and assurance can be guaranteed.

6.1.2 Yang model to expose device information via NBI

The Yang model used to retrieve the logical information from the uses schema-mount to expose to NBI the device-specific configuration provided by OpenConfig models on the SBI. Each query/change in the interfaces attributes requires using the Network topology and the tip-ne-commissioning/interfaces-config data path to access the available configuration and state model features. The tip-ne-commissioning yang files contains the mount point definitions to allow the network controller to add the whole parameters of the openconfig-interfaces data model at the node level.

```
module: ietf-network
  +--rw networks
    +--rw network* [network-id]
      +--rw network-id          network-id
      +--rw node* [node-id]
        | +--rw node-id          node-id
        | | +--rw ne-mgmt:commissioning-configs
        | | +--rw ne-mgmt:ne-interfaces
        | | +--rw ne-com:logical-interfaces
```

6.1.3 Retrieve logical [Interfaces] inventory use case

Number	2.1
Name	Retrieve logical interfaces inventory
Brief description	The use case here should focus on retrieving the information regarding all the logical or virtual interfaces of a specific equipment such as subinterfaces, VLAN interfaces, tunnel interfaces and other non-physical interfaces.

6.1.3.1 Parameters description, general table and example

As it was mentioned in previous sections, inventory parameters exposed to the NBI interface are those detailed in the SBI. Because of that, the concrete list or the params related to this use case can be found on the [MUST_SBI_IP] document.

6.1.3.2 Use case operations table

Operation	Description
GET /restconf/data/ietf-network:networks/network={network-id}/node={node-id}/ne-commissioning-ne:commissioning-configs/interfaces/ne-interfaces/interfaces-root/	Retrieve all the interface configuration parameters from one node

7

Traffic Engineering and Network Optimization

Traffic engineering (TE) allows the enforcement of traffic steering flows by leveraging onto MPLS tunnels or Segment Routing paths. This permits to increase the efficiency on the use of the network resources by properly mapping the traffic flows to the available resources, and improve network management, including troubleshooting, to overcome difficult failure situations...



7 Traffic Engineering and Network Optimization

Traffic engineering (TE) allows the enforcement of traffic steering flows by leveraging onto MPLS tunnels or Segment Routing paths. This permits to increase the efficiency on the use of the network resources by properly mapping the traffic flows to the available resources, and improve network management, including troubleshooting, to overcome difficult failure situations. Increasingly complex network scenarios such as large single domain environments, multi-domain or multi-layer networks require the usage of algorithms for efficiently computing end-to-end paths.

This complexity is driving the need for a dedicated SDN controller, which will perform path computations and be adaptive to network changes. The **Path Computation Element (PCE)** function allows performing complex constrained based path calculation over a network graph representation. The centralized path computations introduced by the PCE, improves the application of TE policies in MPLS and GMPLS networks by mitigating race conditions inherent of distributed systems.

Based on these functionalities, the main purpose of traffic engineering use cases in is to reduce overall operating costs through more efficient use of network resources, including link occupation, traffic rerouting, network availability, and other components. Essentially, traffic engineering actions address the need to prevent situations where some parts of the network are overloaded, while other parts of the network remain underused and thus ensure the most appropriate path for network traffic and allow the implementation of mechanisms for protecting traffic against network failures.

7.1 Architecture and application of NBI

Within the required architecture for Traffic Engineering, the Path Computation Element (PCE) Function is a module that interacts with the rest of control and management modules. To guarantee interoperability, the PCE is required to use standard interfaces to communicate with the rest of the modules of the SDN control layer.

The Northbound Interface will be used to deal with **both RSVP-TE signaled LSP tunnels and Segment Routing Paths**. Throughout the document we will use LSP generically to refer to both.

Upon request of RESTCONF NBI the controller shall be able to steer traffic according to policies across candidate paths that can be either MPLS tunnels or SR paths according to (draft-ietf-spring) Segment Routing Policy Architecture. The candidate paths and the policies association can be done in different ways, but PCEP is a good option to make it dynamic and active-stateful (draft-ietf-pce-segment-routing-policy-cp-01). LSP paths association extensions in PCEP are described in (RFC 8697) as described in the SBI specification document. Some other approaches for SBI policy for SR are considered such as yang models in IETF (e.g., draft-ietf-spring-sr-policy-yang) but needed to be considered in RESTCONF NBI yang data model requests.

7.2 TE Use Cases

There are several needs now to operate the network. However, one of the major tasks is to optimize and to control the way the traffic flows through the network during a failure. The TE use cases described in this section are defined to facilitate the network operation and optimize the network occupancy.

The lists of use cases identified for traffic engineering that are covered in this document are:

Traffic Engineering / PCE		Section
4.1	LSP Creation, modify and delete with RSVP-TE no constraints	7.3
4.2	LSP Creation, modify and delete with SR no constraints	7.4
4.3	LSP create, modify and delete with constraints (delay, bandwidth and hop count)	7.5
4.4	LSP create, modify and delete with constraints (delay, bandwidth and hop count) and explicit Path (strict and loose)	7.6
4.5	LSP create, modify and delete with constraints (delay, bandwidth and hop count) Protection: Redundancy 1+1	7.7

The Traffic Engineering use cases will be enhanced to consider the traffic steering into the tunnels and the association to services.

7.2.1 Use Case Features

Current version of the Traffic Engineering NBI supports the following main features:

BASIC CREATION	Source & Destination	
	Name	
	Unidirectional or bidirectional(co-routed)	
	Signaling	RSVP-TE/SR
CONSTRAINTS	Specify max value for:	Hop count
		Delay
		BW
RESILIENCE/PROTECTION	Nothing	
	1+1 (hot standby) (2LSPs established)	
	Restoration (ANY:if any change occurs the restoration takes place)	
UPDATE	Change Tunnel description	
	Change Tunnel Admin Status(Up & Down)	

7.2.2 Use Case Operations

In summary, the main operations needed in the Traffic Engineering API are listed below. Note that, though the use cases, an additional set of READ operations is indicated to facilitate clients querying relevant parts of the data model (such as protection, restoration, details of a path..)

CREATE

- POST** Create a TE-Tunnel
- POST** Create global NPC set

UPDATE

- PATH** Update a TE-Tunnel

READ

- GET** List all TE Tunnels
- GET** Show details of a TE-Tunnel
- GET** List globally defined NPC sets
- GET** Show global NPC set details

DELETE

- DEL** Delete a TE-Tunnel
- DEL** Delete a global NPC set

7.3 LSP Creation, modify and delete with RSVP-TE no constraints

Number	4.1
Name	LSP Creation, modify and delete with RSVP-TE no constraints
Brief description	Currently the LSP administration is usually managed manually. So, the creation, modification or deletion of LSPs depends of the knowledge and skills of the operations teams that depending on the necessity, make the necessary changes to fulfil the optimum performance on the network. This use case motivates the automatic creation and administration of LSPs in the network using a centralized network controller. The first use cases covers the triggering of the LSP creation via the northbound interface of the controller. The LSP will be signaled in the network via RSVP-TE and no constraints nor resiliency requirements are included.

The Traffic Engineering (TE) Use cases are covered in an incremental manner, from the simplest one to those more complex. The first TE use case is about the creation of LSPs from the NBI of the Network Controller without any type of constraint or restriction. However, the request will include the possibility to indicate the type of signaling of the LSP in the network. In the use case the desire is that the path created between two end points is signaled via RSVP-TE in the MPLS network.

The request of the creation can be:

- Triggered by the operator, e.g., due to a planning process, or to overcome a problem in the network.
- Triggered from an operator's planning tool.
- Triggered automatically by the own SDN controller.



- Triggered by a Hierarchical SDN Controller, e.g., to orchestrate the associations of LSPs to VPNs.

The triggers mentioned before are only mentioned as examples and don't impact on the NBI of the Controller/PCE. Note that if the PCE is implemented as a separate piece, it will need to implement the NBI. However, the specification does not state such implementation/deployment decisions.

Two types of operation are possible, selectable by policy in the controller.

- PCE-Initiated LSPs
- PCC-Initiated LSPs

The NBI will NOT indicate if the LSP is PCC or PCE initiated. The operation will be able to choose the desired behavior for all the LSPs created externally as mentioned. The level of interoperability at PCEP level may restrict temporarily the use of PCE-Initiated LSPs. The interaction of the PCE with the devices is covered in the SBI specification. Regardless PCC or PCE initiated, the LSPs created via the North Bound Interface will be delegated to the PCE.

7.3.1 PCE Initiated LSPs

In the first type of LSP, the PCE module receives the petition in RESTCONF (1) and internally does the translation to the PCEP Initiate message (details in the SBI document) The PCE module interacts with the network elements using PCEP (2) and sends the PCEP Init and Create messages to deploy the LSPs in the network.

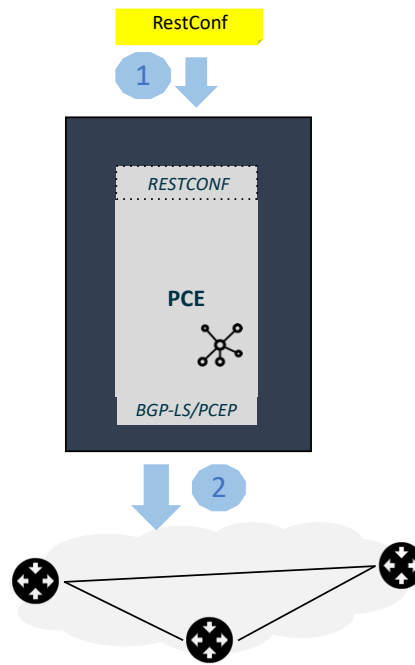


Figure 15. PCE Initiated

7.3.2 PCC Initiated LSPs

In the second type of operation, the Controller, after receiving the LSP creation request using RESTCONF (1), proceeds to configure the NE which is the source of the LSP using NETCONF/Yang (please refer to the SBI document for details) (2). The source NE requests the LSP computation to the PCE using PCEP. The PCE returns the computed path to the NE (3).

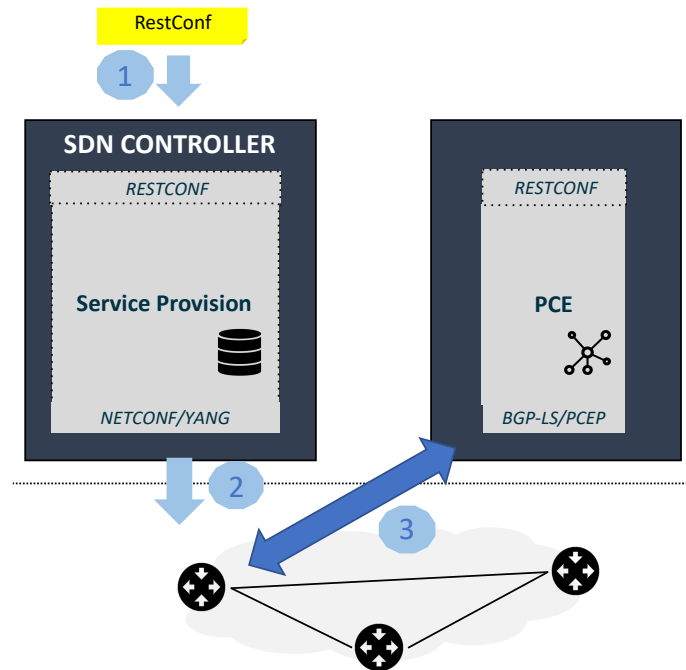


Figure 16. PCC Initiated

7.3.3 Basic LSP Creation without constraints and restoration: Parameters

The procedure to create a LSP tunnel is to instantiate a new “tunnel” entry in the tunnels list of the ietf-te yang module. When the controller receives the new tunnel entry, one or more LSPs will be instantiated depended on the requested parameters. Typically, in a bidirectional tunnel with no protection, two LSPs will be created.

The set of parameters for the basic LSP creation are the following, some of them are OPTIONAL. That means they are not required to deploy the tunnel on the network, however, are relevant for the documentation and troubleshooting purposes. This specification focuses on the parameters for the IETF yang model that have been considered relevant for the use case.

The parameters required to create a basic tunnel are:

- **Name:** Unique identifier for each Tunnel. Must be provided in the NBI
- **Admin State:** This is a configuration Parameter. Reflects the administrative status

of the Tunnel on the Network, i.e., Up or Down (the intention of the network operator). If it is not provided in the NBI, the default value “tunnel-admin-state-up” is used.

- **Description:** Tunnel description. Optional parameter that can be provided in the NBI request.
- **Encoding:** This parameter is optional. The original use of MPLS was to encapsulate IP packets, and thus MPLS is often described as Layer 2.5. However, more recent specifications describe the encapsulation of Layer 2 frames as well (RFC3471 for GMPLS). In this use case the encoding supported is only for IP packets, so the only accepted value is “lsp-encoding-packet”. The rest of values will not be accepted. The default value is lsp-encoding-packet (in case the value is not provided in the NBI).
- **Source:** Tunnel source. In the MUST IP NBI specification, the source parameter is mandatory. It is represented as 4 octets in dotted-quad notation. Typically, the loopback address of the router where the tunnel will be originated is used as source. The source address MUST refer to the Node-id in the L3 Topology (see L3 Topology chapter)
- **Destination:** Tunnel destination. The destination address MUST be included to the L3 Topology Node-id (see L3 Topology chapter). In the MUST IP NBI specification, the destination parameter is mandatory.
- **Bidirectional.** The parameter is optional, with default value “false” in the standard yang model. In the MUST IP NBI specification, the simple LSP use cases is unidirectional.
- **Signaling Type:** The Tunnel can be implemented using different signaling protocols. The possible choices are RSVP (for RSVP-TE signaled LSP) SR (to use segment routing). In this use case, **RSVP-TE is used**

In addition, there are a set of parameters that are not sent in the NBI, but can be read querying the controller and are part of the Yang module:

- **Identifier:** The identifier is autogenerated by the controller and is retrieved by querying the tunnel. In case of RSV-TE Tunnels, the identifier will reflect the Tunnel ID as defined in REF RFC3209. Note that according to the IETF yang the parameter can be set. However, our recommendation from MUST IP NBI specification is, for simplicity, to auto-generate the number in the controller or in

the node.

- **Operational State:** This is a **Read-Only Parameter**. Reflects the status of the Tunnel on the Network, i.e.: up or down.

Attribute	Format	Allowed Values	Provided by	Description
name	string	-	NBI	Key. Name of the Tunnel
identifier?	uint16	-	Controller/PCE	Read only. numeric identifier of the tunnel
description?	string	-	NBI	Optional tunnel description
encoding?	identityref	"lsp-encoding-packet"	NBI (DEF)	LSP encoding type
admin-state?	identityref	"tunnel-admin-state-up" "tunnel-admin-state-down"	NBI (DEF)	TE tunnel administrative state.
source?	te-types:te-node-id	NE_ID	NBI	NE-ID of the tunnel source
destination?	te-types:te-node-id	NE_ID	NBI	NE-ID of the tunnel destination
signaling-type?	identityref	"path-setup-rsvp"	NBI	Type of signaling protocol selected.
bidirectional	boolean	True False	NBI (DEF)	If the Tunnel is bidirectional.
operational-state?	identityref	"tunnel-state-up" "tunnel-state-down"	Controller/PCE	Read only parameter. This reflects the tunnel state in the network.

Table 37 Attributes (configurable and state) for basic Basic RSVP-TE Tunnel

7.3.4 TE Tunnel Basic Operations

The following section describes how to create/retrieve an LSP using the IETF-TE YANG model with the recommended parameters.

The basic operations are:

DESCRIPTION	OPERATION
Create a TE-Tunnel	POST /restconf/data/ietf-te:te/tunnels/
Show details of a TE-Tunnel	GET /restconf/data/ietf-te:te/tunnels/tunnel={name}/
List all TE Tunnels	GET /restconf/data/ietf-te:te/tunnels/
Delete a TE-Tunnel	delete /restconf/data/ietf-te:te/tunnels/tunnel={name}/
Show the Lsp state	GET /restconf/data/ietf-te:te/lsp-state/lsp={source},{destination},{tunnel-id},{lsp-id},{extended-tunnel-id}/
Show Primary Paths of an specific TE-Tunnel	GET /restconf/data/ietf-te:te/tunnels/tunnel={name}/primary-paths/
Returns the Computed Path Error Information of primary path in a TE Tunnel	GET /restconf/data/ietf-te:te/tunnels/tunnel={name}/ primary-paths/primary-path={primary-path-name}/computed-path-error-infos/
Update the Tunnel Name	PATCH /restconf/data/ietf-te:te/tunnels/tunnel={name}/
Update the Tunnel Status	PATCH /restconf/data/ietf-te:te/tunnels/tunnel={name}/

Table 38 Basic LSP Operations

7.3.5 Examples

7.3.5.1 Example: Create a Basic TE-Tunnel

In the following example a RSVP-TE signaled Tunnel with no restrictions and with no protection/restoration can be created with the following set of required parameters:

Attribute	Format	Sample Value
name	string	Example_LSP_Tunnel
description?	String	Example_LSP_Tunnel
encoding?	Identityref	te-types:lsp-encoding-packet
admin-state?	Identityref	te-types:tunnel-admin-state-up
source?	Te-types:te-node-id	10.0.0.1

destination?	Te-types:te-node-id	10.0.0.2
bidirectional	boolean	false
signaling type?	Identityref	te-types:path-setup-rsvp

Table 39 Sample parameters of a Basic unprotected RSVP-TE Tunnel creation

```
POST /restconf/data/ietf-te:te/tunnels HTTP/1.1
Host: example.com
Accept: application/yang-data+json
{
  "ietf-te:tunnel": [
    {
      "name": "Example_LSP_Tunnel",
      "encoding": "te-types:lsp-encoding-packet",
      "admin-state": "te-types:tunnel-admin-state-up",
      "source": "10.0.0.1",
      "destination": "10.0.0.2",
      "bidirectional": "false",
      "signaling-type": "te-types:path-setup-rsvp"
    }
  ]
}
```

7.3.5.2 Example Show details of an existing LSP by name

This query can be used to get the information of a given LSP and to know if the tunnel was successfully created by checking the “operational-state” leaf.

```
GET /restconf/data/ietf-  
te:te/tunnels/tunnel=Example_LSP_Tunnel HTTP/1.1  
Host: example.com  
Accept: application/yang-data+json  
{  
  "ietf-te:tunnel": [  
    {  
      "name": "Example_LSP_Tunnel",  
      "encoding": "te-types:lsp-encoding-packet",  
      "admin-state": "te-types:tunnel-admin-state-up",  
      "source": "10.0.0.1",  
      "destination": "10.0.0.2",  
      "bidirectional": "false",  
      "signaling-type": "te-types:path-setup-rsvp"  
    }  
  ]  
}
```

7.3.5.3 Example UPDATE details of an existing LSP: Description

This query can be used to update the information of a given LSP.

```
PATCH /restconf/data/ietf-  
te:te/tunnels/tunnel=Example_LSP_Tunnel HTTP/1.1  
Host: example.com  
Accept: application/yang-data+json  
{  
  "ietf-te:tunnel": [  
    {  
      "description": "LSP_Tunnel",  
    }  
  ]  
}
```

7.3.5.4 Example UPDATE details of an existing LSP: Administrative Status

```

PATCH /restconf/data/ietf-
te:te/tunnels/tunnel=Example_LSP_Tunnel HTTP/1.1
Host: example.com
Accept: application/yang-data+json
{
  "ietf-te:tunnel": [
    {
      "admin-state": "te-types:tunnel-admin-state-up"
    }
  ]
}

```

7.3.5.5 LSP Creation: Workflow

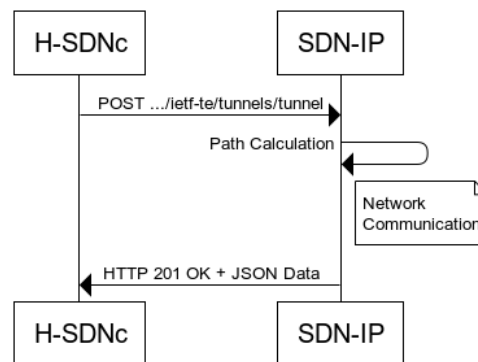


Figure 17. LSP Creation

7.4 LSP Creation, modify and delete with SR no constraints

Number	4.2
Name	LSP create, modify and delete with constrains and Segment Routing signaling
Brief Description	In this use case the LSP is establish based on constraint-based routing performed by the PCC (router/NE) or a external PCE. The LSP will use SEGMENT ROUTING as signalling protocol and won't have any kind of protection

The set of parameters required to create a Tunnel using SR and a MAX-HOP restriction are the following:

Attribute	Allowed Values/Format	Format
tunnel* [name]	string	Simple_LSP
name	string	Simple_LSP
description?	string	Simple_LSP_with_named_path
encoding?	identityref	te-types:lsp-encoding-ethernet
admin-state?	identityref	te-types:tunnel-admin-state-up
source?	te-types:te-node-id	10.0.0.1
destination?	te-types:te-node-id	10.0.0.2
bidirectional	boolean	false
signaling type?	identityref	te-types:path-setup-sr

Table 40 Attributes (configurable and state) for LSP Creation, modify and delete with SR no constraints

The sample POST call is the following:

```
POST /restconf/data/ietf-te:te/tunnels HTTP/1.1
Host: example.com
Accept: application/yang-data+json
{
  "ietf-te:tunnel": [{
    "name": "Simple_LSP",
    "encoding": "te-types:lsp-encoding-packet",
    "admin-state": "te-types:tunnel-admin-state-up",
    "source": "10.0.0.1",
    "destination": "10.0.0.2",
    "bidirectional": "false",
    "signaling-type": "te-types:path-setup-sr"
  }]
}
```

7.5 LSP create, modify and delete LSPs with delay, bandwidth and hop count constraints

Number	4.3
Name	LSP create, modify and delete with constraints and Segment Routing signaling
Brief	In this use case the LSP is established based on constraint-based

Description	routing performed by the PCC (router/NE) or an external PCE. The result of Constrained Shortest Path First (CSPF) algorithm will be the path that meets the constraints or the set (combination) of constraints considered, in this case delay, BW and hop limit. The LSP will use RSVP or SR as signaling protocol and won't have any kind of protection.
--------------------	--

MPLS tunnels creations go beyond of just provision and signal some paths on the network. The great advantage of the PCE deployment in the network is the possibility to use certain criteria to optimize the network and maximize the links occupancy. Thus, some constrains can be applied during the path creation.

This constrains can have several types and characteristics:

- **Delay:** Provision the path with an upper bound for the delay.
- **Bandwidth:** Reserve/guarantee certain BW in the network to support the path traffic.
- **Hop Count:** Create a path with a strict number of hops or minimize the number of hops a path can cross.
- **Signaling Type:** The changes in the constrains can be implemented using different signaling protocols.
- **Protection Type:** Constrains include the several redundancy types.

The IETF-TE model supports the selection of a set of specific constrains to be applied to a specific tunnel. Two usage options are available.

- **First,** create a global Constrain Policy. That policy can be reused between several tunnels.
- **Second,** apply the constraints directly over the tunnel.

7.5.1 LSP Creation with constraints Option 1: Global Constrain Policy

Inside the Global variables of the IETF-TE module a Named Path Constraint (NPC) set can be created. A NPC set is a global policy to be reused. As depicted in Figure 18 NPC set has the set of parameters to optimize the tunnel deployment in the network. This policy can be reused and is referenced by the Tunnel module using a leaf-ref with the name of the policy.

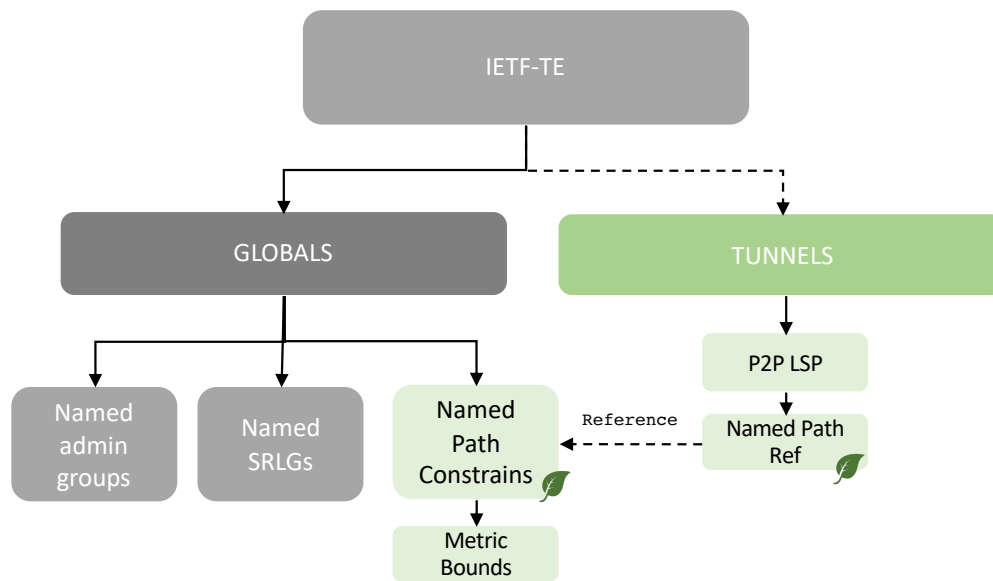


Figure 18. IETF TE Name Path Constrains

The named path constrain has several possible configurations including changes of the signaling or protection type. However, this use case restricts the usage of this policy to the metric bounds container which has the following possible constrains:

- **Metric Type:** Metric-types bound for the TE path.
 - **path-metric-delay-minimum:** Refers to the Unidirectional minimum link delay.
 - **path-metric-hop:** Hop path metric
 - **path-metric-residual-bandwidth:** Unidirectional Residual Bandwidth, which is defined to be Maximum Bandwidth minus the bandwidth currently allocated to LSPs.
 - **path-metric-te:** TE path metric
- **Upper Bound:** Upper bound on end-to-end TE path metric. A zero indicate an unbounded upper limit for the specific metric-type

The details of the constrain policy the following parameters are the minimum required:

Attribute	Allowed Values/Format	Format	Resp	Description
named-path-constraint*	[name]		NBI	KEY. Unique name to identify the policy
metric-type	identityref	path-metric-delay-minimum path-metric-hop path-metric-te	NBI	Identifies an entry in the list of metric-types bound for the TE path.
upper-bound?	uint64		NBI	Upper bound on end-to-end TE path metric. A zero indicate an unbounded upper limit for the specific metric-type

Table 41 Attributes (configurable and state) for LSP Creation with constraints Option 1: Global Constrain Policy

Once the NPC (policy) is formed, the tunnel must be created using the corresponding parameters (described in section 7.3.3) for the LSP creation and using a leaf-ref named-path-constraint to apply the policy to the tunnel.

7.5.2 LSP Creation with constraints: Tunnels Constrain Policy

Second option allows the selection of the optimization algorithm and restriction policy directly on each of the tunnel components during its creation; Instead of using a global policy. As depicted in Figure 19, the tunnel CAN have a primary and secondary path. In Each of the Paths the optimization metric policies that can be applied are the following:

- **Metric-Type:** The same as described in previous version.
- **Algorithm:**
 - **objective-function-type:** Objective function entry
- **path-computation-method:** The method used for computing the path, either locally computed, queried from a server or not computed at all (explicitly must be configured).

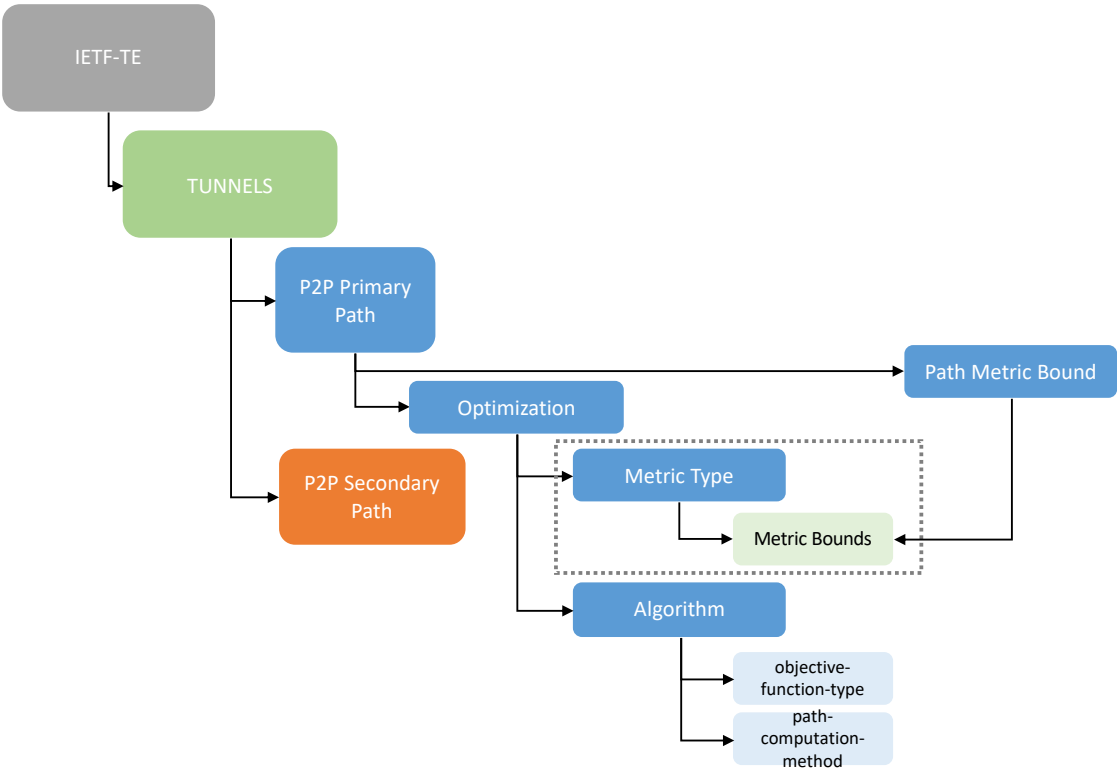


Figure 19. Tunnel Constrains

The details of the constrain TE Tunnel parameters used to apply constrains are:

Attribute	Allowed Values/Format	Format	Resp	Description
objective-function-type	list	of-maximize-residual-bandwidth of-minimize-load-path of-minimize-cost-path of-minimize-agg-bandwidth-consumption of-minimize-load-most-	NBI	Objective function entry

		loaded-link of-minimize- cost-path-set		
path-computation-method	list	path-externally-queried	NBI	The method used for computing the path, either locally computed, queried from a server or not computed at all (explicitly must be configured).
metric-type	identityref	path-metric-delay-minimum path-metric-hop path-metric-te	NBI	Identifies an entry in the list of metric-types bound for the TE path.
upper-bound?	uint64		NBI	Upper bound on end-to-end TE path metric. A zero indicate an unbounded upper limit for the specific metric-type
path-computation-server	ip-address		NBI	IP Address of the PCE
use-path-computation	boolean	True False	NBI	Boolean

Table 42 Attributes (configurable and state) for LSP Creation with constraints Tunnel Constrains Policy

7.5.3 TE Tunnel Operations related to constraints

In addition to the basic LSP operations defined in Table 37, where the only difference is that more parameters are required, two additional calls are needed:

DESCRIPTION	OPERATION
Create global NPC set	POST /restconf/data/ietf-te:te/globals/named-path-constraints/
Delete a Named Path Constraint set given its name.	DELETE /restconf/data/ietf-te:te/globals/named-path-constraints/named-path-constraint={name}/
Show global NPC set	GET /restconf/data/ietf-te:te/globals/named-path-

details	constraints/named-path-constraint={name}/
List globally defined NPC sets	GET /restconf/data/ietf-te:te/globals/named-path-constraints/
Return list of LSPs associated to a primary path of a TE Tunnel	GET /restconf/data/ietf-te:te/tunnels/tunnel={name}/primary-paths/primary-path={primary-path-name}/lsps/
Returns secondary Paths of an specific TE-Tunnel	GET /restconf/data/ietf-te:te/tunnels/tunnel={name}/secondary-paths/
Returns the Computed Path Error Information of secondary path in a TE Tunnel	GET /restconf/data/ietf-te:te/tunnels/tunnel={name}/secondary-paths/secondary-path={secondary-path-name}/computed-path-error-infos/
Returns the LSPs associated to the secondary path in a TE Tunnel	GET /restconf/data/ietf-te:te/tunnels/tunnel={name}/secondary-paths/secondary-path={secondary-path-name}/lsps/
Delete a Named Path Constraint set given its name.	DELETE /restconf/data/ietf-te:te/globals/named-path-constraints/named-path-constraint={name}/

Table 43 Basic LSP Operations

7.5.4 Examples

7.5.4.1 Example: Create Global Constraint Policy- Path constrain - *path-metric-delay-minimum*

As mentioned in 7.5.1 the first step is the creation of the named-path-constraints. In this example the **metric-type** is *path-metric-delay-minimum*. The policy has the same name and the *upper-bound* has a value of 30ms.

Attribute	Allowed Values/Format	Example
named-path-constraint*	[name]	path-metric-delay-minimum
metric-type	identityref	te-types:path-metric-delay-minimum
upper-bound?	uint64	30

The sample call is the following:

```
POST /restconf/data/ietf-te:te/globals/named-path-constraints
HTTP/1.1
Host: example.com
Accept: application/yang-data+json
{
  "ietf-te:named-path-constraint": [
    {
      "name": "path-metric-delay-minimum",
      "signaling-type": "te-types:path-setup-rsvp",
      "path-metric-bounds": {
        "path-metric-bound": {
          "metric-type": "te-types:path-metric-delay-
minimum",
          "upper-bound": "30"
        }
      }
    }
  ]
}
```

If other metric type is selected the path-metric-bounds/metric-type MUST be changed. For example:

Attribute	Allowed ValuesFormat	Example
named-path-constraint*	[name]	path-metric-delay-minimum
metric-type	identityref	te-types:path-metric-hop
upper-bound?	uint64	30

The sample POST is the following:

```

POST restconf/data/ietf-te:te/globals/named-path-constraints
HTTP/1.1
Host: example.com
Accept: application/yang-data+json
{
  "ietf-te: named-path-constraint":{
    "name":"path-metric-hop",
    "signaling-type":"te-types:path-setup-rsvp",
    "path-metric-bounds":{
      "path-metric-bound":{
        "metric-type":"te-types:path-metric-hop",
        "upper-bound":"30"
      }
    }
  }
}

```

7.5.4.2 Example: Create TE Tunnel with Associate Constraint Global Policy (NPC set)

The next step is the association of the previously created path constrain in the TE-Tunnel creation. For this, you must use the Tunnel creation procedure described in the previous section and reference the named-path-constraint within the P2P LSP.

Attribute	Allowed Values/Format	Format
tunnel* [name]	string	Simple_LSP
name	string	Simple_LSP
description?	string	Simple_LSP_with_named_path
encoding?	identityref	te-types:lsp-encoding-ethernet
admin-state?	identityref	te-types:tunnel-admin-state-up
source?	te-types:te-node-id	10.0.0.1
destination?	te-types:te-node-id	10.0.0.2
src-tp-id?	yang:hex-string	00:00:00:01
dst-tp-id?	yang:hex-string	00:00:00:02
signaling type?	identityref	RSVP
primary-path/name	string	Simple_LSP_1

primary-path/named-path-constraint	identityref	path-metric-delay-minimum
use-path-computation	boolean	True

Table 44 Sample parameters of an unprotected RSVP-TE Tunnel with associated NPC creation

7.5.4.3 Example: Create TE Tunnel: LSP with constrain path-metric-delay-minimum

As mentioned before, the tunnel can be created and at the same time, the optimization and restrictions can be applied directly to the Tunnel/LSP. The set of required parameters are:

```
POST /restconf/data/ietf-te:te/tunnels HTTP/1.1
Host: example.com
Accept: application/yang-data+json
{
  "ietf-te:tunnel":[
    {
      "name":"Simple_LSP",
      "encoding":"te-types:lsp-encoding-packet",
      "description":"Simple_LSP_with_named_path",
      "admin-state":"te-types:tunnel-admin-state-up",
      "source":"10.0.0.1",
      "destination":"10.0.0.2",
      "signaling-type":"te-types:path-setup-rsvp",
      "bidirectional":"false",
      "primary-paths":[
        {
          "primary-path:{
            "name":"Simple_LSP_1",
            "use-path-computation":"True",
            "named-path-constraint":"path-metric-delay-
minimum"
          }
        }
      ]
    }
  ]
}
```

Attribute	Allowed ValuesFormat	Format
tunnel* [name]	string	Simple_LSP

name	string	Simple_LSP
description?	String	Simple_LSP_with_named_path
encoding?	Identityref	te-types:lsp-encoding-packet
admin-state?	Identityref	te-types:tunnel-admin-state-up
source?	Te-types:te-node-id	10.0.0.1
destination?	Te-types:te-node-id	10.0.0.2
bidirectional	boolean	false
signaling type?	Identityref	path-setup-rsvp
primary-path/name	string	Path1
primary-path/optimizations/optimization-metric/metric-type	identityref	path-metric-delay-minimum
use-path-computation	boolean	true
primary-paths/path-metric-bounds/metric-type	identityref	path-metric-delay-minimum
primary-paths/path-metric-bounds/upper-bound	uint16	30

The sample POST call is the following:

```

POST /restconf/data/ietf-te:te/tunnels HTTP/1.1
Host: example.com
Accept: application/yang-data+json
{
  "ietf-te:tunnel":[
    {
      "name":"Simple_LSP",
      "encoding":"te-types:lsp-encoding-packet",
      "admin-state":"te-types:tunnel-admin-state-up",
      "source":"10.0.0.1",
      "destination":"10.0.0.2",
      "bidirectional":"false",
      "signaling-type":"te-types:path-setup-rsvp",
      "primary-paths:{
        "primary-path":[
          {
            "name":"path1",
            "path-computation-method":"te-
types:path-externally-queried",
            "path-computation-server":"10.0.0.1",
            "use-path-computation":"true",
            "optimizations:{
              "optimization-metric:{
                "metric-type":"te-types:path-
metric-delay-minimum"
              },
              "objective-function:{
                "objective-function-type":"te-
types:of-minimize-agg-bandwidth-consumption"
              }
            },
            "path-metric-bounds:{
              "path-metric-bound:{
                "metric-type":"te-types:path-
metric-delay-average",
                "upper-bound":"30"
              }
            }
          }
        ]
      }
    }
  ]
}

```

As described before, the Policies can be applied to the Tunnels using Global Constraint Policies or directly during the Paths Assignment in the Tunnel Creation. In the following uses cases the second option will be described.

The set of parameters required to create a Tunnel using RSVP and a MAX-HOP restriction are the following:

Attribute	Allowed ValuesFormat	Format
tunnel* [name]	string	Simple_LSP
name	string	Simple_LSP
description?	string	Simple_LSP_with_named_path
encoding?	identityref	te-types:lsp-encoding-ethernet
admin-state?	identityref	te-types:tunnel-admin-state-up
source?	te-types:te-node-id	10.0.0.1
destination?	te-types:te-node-id	10.0.0.2
bidirectional	boolean	false
signaling type?	identityref	te-types:path-setup-rsvp
primary-path/name	string	Path1
primary-path/path-computation-method	identityref	te-types:path-externally-queried
primary-path/path-computation-server	ipaddress	10.10.10.1
primary-path/ optimizations/ optimization-metric/ metric-type	identityref	path-metric-hop
use-path-computation	boolean	true
primary-paths/ path-metric-bounds/metric-type	identityref	path-metric-hop
primary-paths/ path-metric-bounds/ upper-bound	uint16	3

The sample POST call is the following:

```

POST /restconf/data/ietf-te:te/tunnels HTTP/1.1
Host: example.com
Accept: application/yang-data+json
{
  "ietf-te:tunnel":[
    {
      "name":"Simple_LSP",
      "encoding":"te-types:lsp-encoding-packet",
      "admin-state":"te-types:tunnel-admin-state-up",
      "source":"10.0.0.1",
      "destination":"10.0.0.2",
      "bidirectional":"false",
      "signaling-type":"te-types:path-setup-rsvp",
      "primary-paths":{
        "primary-path":[
          {
            "name":"path1",
            "path-computation-method":"te-types:path-
externally-queried",
            "path-computation-server":"10.10.10.1",
            "use-path-computation":"true",
            "optimizations":{
              "optimization-metric":{
                "metric-type":"te-types:path-metric-
hop"
              },
              "objective-function":{
                "objective-function-type":"te-
types:of-minimize-agg-bandwidth-consumption"
              }
            },
            "path-metric-bounds":{
              "path-metric-bound":{
                "metric-type":" te-types:path-metric-
hop",
                "upper-bound":"3"
              }
            }
          }
        ]
      }
    }
  ]
}

```

7.6 LSP create, modify and delete with constraints (delay, bandwidth and hop count) and explicit Path (strict and loose)

Number	4.4
Name	LSP create, modify, and delete with constraints (delay, bandwidth and hop count) and explicit Path (strict and loose)
Description	An explicit-path (either SR or RVSP-TE) is triggered from an operator action. Once the operator triggers the creation of the path, it is initiated using the explicitly specified path which comes from a planning process. If the path is topologically not feasible, either because the network is partitioned or insufficient resources are available along some parts of the path, the LSP creation will fail. No alternative paths can be used. If the setup succeeds, the LSP stays on the defined path indefinitely. The initiated path in this use case does not consider protection. The explicit path can contain strict or loose hops.

In this use case, the additional constraint with regards to previous use cases is that a explicit path is provided at provisioning time. That explicit path might have been computed by the network planner or an external planning tool. In any case, the path is always going to be double checked against the PCE, so if the provided path is NOT feasible, the tunnel creation will fail. In this use case, the hops of the explicit paths are nodes (routers).

To indicate the explicit path, in the tunnel a “primary-path” object needs to be provided. Withing that primary path, the key is to provide new entries in the array “route-object-include-exclude” to provide the explicit hops. This new entries, in this use case are set as “te-types:route-include-object”, which indicates that the hop MUST be included in the path. The hops can be strict or loose. In case of loose hops, the PCE will complete the path to reach those hops, so the metric bounds constraints and optimization metric explained in the previous use cases can be optionally included. If the path is fully strict, it is recommended not to add metric bounds or optimization metric. In this version of the specification, the hops must be provided as IP addresses (the type is “numbered-node-hop”). Further options will be explored later.

The set of parameters required to create a Tunnel using with the explicit hops’ restriction are the following:

Attribute	Format	Value
tunnel* [name]	string	Simple_LSP
name	string	Simple_LSP
description?	string	Simple_LSP_with_named_path
encoding?	identityref	"te-types:lsp-encoding-ethernet"
admin-state?	identityref	te-types:tunnel-admin-state-up or te-types:tunnel-admin-state-down
source?	te-types:te-node-id	10.0.0.1 (example)
destination?	te-types:te-node-id	10.0.0.2 (example)
bidirectional	boolean	false
signaling type?	identityref	te-types:path-setup-rsvp or te-types:path-setup-sr
primary-paths	List of primary-path objects	See next table

Table 45 Attributes (configurable and state) for LSP create, modify and delete with constraints (delay, bandwidth and hop count) and explicit Path (strict and loose)-1

Attribute	Allowed Values/Format	Format
primary-path/name	string	Path1
primary-path/path-computation-method	identityref	te-types:path-externally-queried
primary-path/path-computation-server	ipaddress	10.10.10.1 (optional)
primary-path/optimizations/optimization-metric/metric-type	identityref	te-types:path-metric-hop (optional)
use-path-computation	boolean	true
primary-paths/path-metric-bounds/metric-type	identityref	te-types:path-metric-hop (optional)

primary-paths/ path-metric- bounds/ upper- bound	uint16	3 (optional)
primary-paths/ route-object- include-exclude	Object entry in list “route object- include-exclude	One entry per hop to be included. See next table for details

Table 46 Attributes (configurable and state) for LSP create, modify and delete with constraints (delay, bandwidth and hop count) and explicit Path (strict and loose)-2

Attribute	Allowed Values/Format	Values
primary-paths/ route- object-include- exclude/explicit-route- usage	identityref	“te-types:route-include-object”
primary-paths/ route- object-include- exclude/explicit-route- usage/type/numbered- node-hop	te-node-id	Ip-address
primary-paths/ route- object-include- exclude/explicit-route- usage/type/hop-type	identityref	strict or loose

Table 47 Attributes (configurable and state) for LSP create, modify and delete with constraints (delay, bandwidth and hop count) and explicit Path (strict and loose)-3

The sample POST call is the following:

```
POST /restconf/data/ietf-te:te/tunnels HTTP/1.1
Host: example.com
Accept: application/yang-data+json

{
  "ietf-te:tunnel": [
    {
      "name": "Simple_LSP",
      "encoding": "te-types:lsp-encoding-packet",
      "admin-state": "te-types:tunnel-admin-state-up",
      "source": "10.0.0.1",
      "destination": "10.0.0.2",
      "bidirectional": "false",
      "signaling-type": "te-types:path-setup-rsvp",
      "primary-paths": {
        "primary-path": [
          {
            "name": "path1",
            "path-computation-method": "te-types:path-externally-queried",
            "path-computation-server": "10.10.10.1",
            "use-path-computation": "true",
            "optimizations": {
              "optimization-metric": {
                "metric-type": "te-types:path-metric-hop"
              }
            },
            "explicit-route-objects-always": {
              "route-object-include-exclude": [
                {
                  "index": "1",
                  "explicit-route-usage": "te-types:route-include-object",
                  "numbered-node-hop": {
                    "node-id": "10.0.0.1",
                    "hop-type": "strict"
                  }
                },
                {
                  "index": "2",
                  "explicit-route-usage": "te-types:route-include-object",
                  "numbered-node-hop": {
                    "node-id": "10.0.0.1",
                    "hop-type": "strict"
                  }
                }
              ]
            }
          },
          {
            "path-metric-bounds": {
              "path-metric-bound": {
```



```
        "metric-type": " te-types:path-metric-hop",
        "upper-bound": "3"
      }
    }
  ]
}
```

7.7 LSP create, modify, and delete with constrains - Protection: Redundancy 1+1

Number	4.5
Name	LSP create, modify and delete with protection constraints and redundancy 1+1
Description	This use case refers to 1+1 protection where there is one working LSP and one protection LSP

Based on the protecting and restoration schemas defined at the beginning of this chapter, this use case has the following characteristics:

- **Protection:** 1+1 Protection
- **Restoration:** Pre-Planned LSP Restoration
- **Disjoint:** Optional
- **Bidirectional Protection:** Yes
- **Restoration-type:** ANY

Based on this a pre-signaled, preconfigured protecting LSP over dedicated resources (labeled as protection in Figure 20) is deployed at the very same time with the working LSP. At the ingress/egress PE, the normal traffic is permanently bridged to both the

working and protection LSP. At the egress node, the normal traffic is selected from the better of the two LSPs (In Figure 20 the green path).

The protection path can be disjoint and use Node, Link or SRLG restrictions to deploy the working and protecting LSPs.

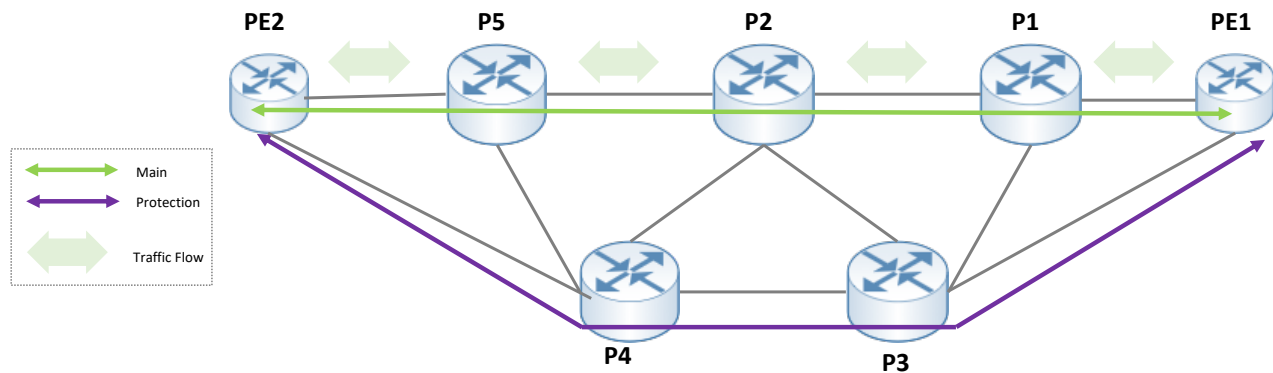


Figure 20. Initial scenario

During a failure scenario as the depicted in Figure 21, the two nodes PE 1 (egress/ingress) and PE 2 (egress/ingress) should be coordinated to switch the traffic from the primary to the protection LSP.

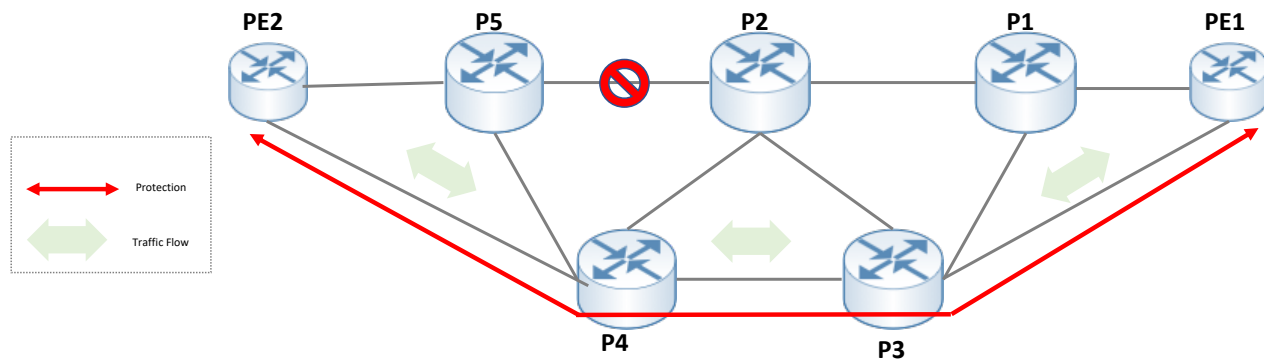


Figure 21. Switching scenario

7.7.1 Applying Protection, Restoration & Resilience to a tunnel

The model IETF-TE supports multiple protection and restoration schemas derived from the GMPLS RFCs [RFC4427]. Possible pro

- **1+1 Protection:** 1+1 protection has one working LSP and one protection LSP. At the ingress node, the traffic is permanently bridged to both the working and protection LSPs. At the egress node, the traffic is selected from the better of the two LSP. Due to the permanent bridging, the 1+1 protection does not allow an unprotected extra traffic signal to be provided.
- **1:N (N >= 1) Protection:** 1:N protection has N working LSPs/spans that carry normal traffic and 1 protection LSP that may carry extra-traffic. At the ingress, the normal traffic is either permanently connected to its working LSP or the protection LSP. At the egress node, the normal traffic is selected from either its working or protection LSP. Unprotected extra traffic can be transported over the protection LSP/span whenever the protection LSP is not used to carry a normal traffic.
- **Pre-Planned LSP Restoration:** Also referred to as pre-planned LSP re-routing. Before failure detection and/or notification, one or more restoration LSPs are instantiated between the same ingress-egress node pair as the working LSP. Note that the restoration resources must be pre-computed and must be signaled. The restoration LSP is not able to carry any extra-traffic.
- **LSP Restoration:** Also referred to as LSP re-routing. The ingress node switches the normal traffic to an alternate LSP that is signaled and fully established (i.e., cross-connected) after failure detection and/or notification. The alternate LSP is signaled from the ingress node and may reuse the intermediate node's resources of the working LSP under failure condition (and may also include additional intermediate nodes.)
- **Shared-Mesh Restoration:** "Shared-mesh" restoration is defined as a particular case of pre-planned LSP re-routing that reduces the restoration resource requirements by allowing multiple restoration LSPs (initiated from distinct ingress nodes) to share common resources (including links and nodes.).

Additionally, the TE model supports the "ALL" and "ANY" restoration schemas. Thus, in that context we are going to describe the possible options in each case:

- **ANY:** Means if a single LSP fail the restoration will be triggered.
- **ALL:** Means if all LSPs of a tunnel fails the restoration will be triggered.

The set of parameters for the LSP creation using protection options are the following:

Attribute	Allowed Values/Format	Format	Resp	Description
lsp-protection-type	list	lsp-protection-unprotected lsp-protection-bidir-1-for-1 lsp-protection-unidir-1-plus-1 lsp-protection-bidir-1-plus-1 lsp-protection-extra-traffic	NBI	Link protection type required for the links included in the computed path
lsp-protection-state	identityref	Normal signal-fail-of-protection lockout-of-protection forced-switch signal-fail signal-degrade manual-switch wait-to-restore do-not-revert failure-of-protocol	NBI	The state of the APS state machine controlling which tunnels is using the resources of the protecting LSP.
disjointness		node link srlg	NBI	The type of resource disjointness. When configured for a primary path, the disjointness level applies to all secondary LSPs. When configured for a secondary path, disjointness level overrides the one configured for the primary path.
protection-reversion-disable	boolean	true false	NBI	Disable protection reversion to working path
restoration-type	identityref	restore-all restore-any	NBI	LSP restoration type. Any= Restores when any of the LSPs is affected by a failure

				All= Restores when all the tunnel LSPs are affected by failure
restoration-scheme	ipaddress	Precomputed presingnaled preconfigured	NBI	LSP restoration scheme.
restoration-reversion-disable	boolean	true false	NBI	Disable restoration reversion to working path

Table 48 Attributes (configurable and state) for LSP creation using protection options

The set of parameters required to create a Tunnel using 1+1 Protection are the following:

Attribute	Allowed Values/Format	Format
tunnel* [name]	string	Simple_LSP
name	string	Simple_LSP
description?	string	Simple_LSP_with_named_path
encoding?	identityref	te-types:lsp-encoding-ethernet
admin-state?	identityref	te-types:tunnel-admin-state-up
source?	te-types:te-node-id	10.0.0.1
destination?	te-types:te-node-id	10.0.0.2
src-tp-id?	yang:hex-string	00:00:00:01
dst-tp-id?	yang:hex-string	00:00:00:02
bidirectional	boolean	false
Protection/enable	boolean	true
protection-type	identityref	lsp-protection-unprotected
protection-reversion-disable	boolean	false
Restoration/enable	boolean	true
restoration-type	identityref	te-types:lsp-restoration-restore-any
restoration-scheme	ipaddress	te-types:restoration-scheme-preconfigured
restoration-reversion-disable	boolean	false

The sample POST call is the following:

```
POST /restconf/data/ietf-te:te/tunnels HTTP/1.1
Host: example.com
Accept: application/yang-data+json

{
  "ietf-te:tunnel": {
    "name": "Simple_LSP",
    "encoding": "te-types:lsp-encoding-packet",
    "admin-state": "te-types:tunnel-admin-state-up",
    "preference": "1",
    "source": "10.0.0.1",
    "destination": "10.0.0.2",
    "bidirectional": "false",
    "protection": {
      "enable": "true",
      "protection-type": "te-types:lsp-protection-bidir-1-plus-
1",
      "protection-reversion-disable": "false"
    },
    "restoration": {
      "enable": "true",
      "restoration-type": "te-types:lsp-restoration-restore-
any",
      "restoration-scheme": "te-types:restoration-scheme
preconfigured",
      "restoration-reversion-disable": "false"
    },
    "signaling-type": "te-types:path-setup-rsvp"
  }
}
```

7.7.2 TE Tunnel Operations related to protection and restoration

In addition to the basic LSP operations defined in Table 37, where the only difference is that more parameters are required, two additional calls are needed:

DESCRIPTION	OPERATION
Returns the Protection properties of a TE-Tunnel	GET /restconf/data/ietf-te:te/tunnels/tunnel={name}/protection/
Show details of the Restoration properties of a TE-Tunnel	GET /restconf/data/ietf-te:te/tunnels/tunnel={name}/restoration/



Table 49 TE Tunnel Protection/RestorationOperations

8 Glossary

ABNO	Application-based network operations
ACL	Access control list
API	Application programming interface
ASBR	Autonomous System Border Route
BGP-LS	Border Gateway Protocol Link-Stat
BGP-LU	BGP Labeled Unicast
BHM	Backhaul Module
BRAS	Broadband remote access server
BSS	Business Support Systems
CDN	Content Delivery Network
CE	Customer Edge
CEM	Circuit Emulation
CGNAT	Carrier-grade NAT
CIR	Committed information rate
CRUD	Create, Read, Update and Delete
DWDM	Dense Wavelength Division Multiplexing
GMPLS	General Multi-Protocol Label Switching
GUI	Graphical user interface
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
ISIS	Intermediate System to intermediate System
L2SM	L2VPN Service Model
L3NM	L3VPN Network Model
LAG	Link aggregation group
LDP	Label Distribution Protocol
LLDP	Link Layer Discovery Protocol
LSP	Label Switch Path
LSR	Label Switching Router
MP-BGP	Multiprotocol Extensions for BGP
MPLS	Multiprotocol Label Switching
NET	Network
NBI	Northbound interface
NMDA	Network Management Datastore Architecture
ONF	Open Networking Foundation
OSPF	Open Shortest Path First
OSS	Operation Support Systems
P2M	Point-to-Multipoint

P2P	Point-to-point
PCE	Path Computation Element Path Computation Element Communication
PCEP	Protocol
PE	Provider Edge
PIR	Peak Information Rate
PKI	Public Key Infrastructure
PW	Pseudowire
QoS	Quality of service
RD	Route Distinguisher
RIP	Routing Information Protocol
RPC	Remote Procedure Call
RSVP	Resource Reservation Protocol
RT	Route Target
SBI	South Bound Interface
SDN	Software Defined Network
SDP	Session Description Protocol
SecGW	Security Gateway
SLA	Service Level Agreement
SSE	Server Sent Events
STP	Spanning Tree Protocol
TAPI	Transport API
TE	Traffic engineering
TTL	Time to live
UNI	User-Network Interface
URI	Uniform resource identifier
VCI	Virtual Circuit Identifier
VLAN	Virtual local area network
VPI	Virtual Path Identifier
VPN	Virtual Private Network
VRF	Virtual Routing and Forwarding

9 References

- [1] Telecom Infra Project, "Open Transport SDN Architecture Whitepaper", available online at https://cdn.brandfolder.io/D8DI15S7/at/jh6nnbb6bjvn7w7t5jbgm5n/OpenTransportArchitectureWhitepaper_TIP_Final.pdf last seen on 28-April-2021.
- [2] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [3] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://datatracker.ietf.org/doc/rfc8040/>.
- [4] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [5] Bierman, A., Bjorklund, M., and K. Watsen, "YANG Module Library", RFC 7895, DOI 10.17487/RFC7895, June 2016, <<https://www.rfc-editor.org/info/rfc7895>>.
- [OpenAPI] OpenAPI Specification Version 3.0.2, <<https://swagger.io/specification/>>
- [RFC 6455] Fette, I. and A. Melnikov, "The WebSocket Protocol", RFC 6455, DOI 10.17487/RFC6455, December 2011, <<https://www.rfc-editor.org/info/rfc6455>>.
- [W3C.REC-eventsourc-20150203] Hickson, I., "Server-Sent Events", World Wide Web Consortium Recommendation REC-eventsourc-20150203, February 2015 Considerations <<http://www.w3.org/TR/2015/REC-eventsourc-20150203>>.
- [MUST_SBI_IP] MUST IP SDN Controller SBI Requirements v1.0, February 2021. https://cdn.brandfolder.io/D8DI15S7/as/mfbj6nm7w38xnbvrmcnp9t6/MUST-IP-Controller-SBI-Requirements-Doc-v10_FINAL_VERSION_WEBSITE.pdf



10 Annex A

10.1 Network Element Management Yang model

```
module ne-management-nm {
  yang-version 1.1;
  namespace "yang:ne-management-nm";
  prefix ne-mgmt-nm;

  import ietf-yang-schema-mount {
    prefix yangmnt;
  }
  import ietf-network {
    prefix nw;
  }

  revision "2021-04-30" {
    description
      "Initial version";
  }

  /* Groupings */

  grouping system-config {
    container system-config {
      yangmnt:mount-point system-root {
        description
          "Mount-point relative
          to system configuration.";
      }
      description
        "Container for system configuration
        mount-point attributes.";
    }
  }

  grouping protocols-config {
    container protocols-config {
      yangmnt:mount-point protocol-root {
```

```
        description
            "Mount-point relative
            to protocols configuration.";
    }
    description
        "Container for protocol mount-point
        configuration attributes (BGP, ISIS, MPLS, PIM).";
    }
}

grouping interfaces-config {
    container ne-interfaces {
        container hardware-ports {
            yangmnt:mount-point port-root {
                description
                    "Mount-point relative
                    to physical ports configuration.";
            }
            description
                "Mount-point relative to physical ports.";
        }
        container logical-interfaces{
            yangmnt:mount-point interfaces-root {
                description
                    "Mount-point relative
                    to interfaces configuration.";
            }
            description
                "Mount-point relative to interfaces on the device
                such as the name, description or IP addressing.";
        }
        description
            "Top container for interface
            configuration attributes.";
    }
}

grouping resources-config{
    container resources-config{
        container routing-policies{
            yangmnt:mount-point routing-root{
                description
                    "Mount-point relative
                    to routing-policies configuration.";
            }
        }
    }
}
```

```
    }
    description
      "Mount-point relative to prefix list,
      BGP communities and other common policy resources.";
  }
  description
    "Top container for resources configuration attributes.";
}

/* Main blocks */

augment /nw:networks/nw:network/nw:node {
  description
    "Augment used to define attach the node configuration";
  container commissioning-configs {
    uses system-config;
    uses protocols-config;
    uses resources-config;
    uses interfaces-config;
  }
}
}
```



11 TIP Document License

By using and/or copying this document, or the TIP document from which this statement is linked, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions:

Permission to copy, display and distribute the contents of this document, or the TIP document from which this statement is linked, in any medium for any purpose and without fee or royalty is hereby granted under the copyrights of TIP and its Contributors, provided that you include the following on ALL copies of the document, or portions thereof, that you use:

1. A link or URL to the original TIP document.
2. The pre-existing copyright notice of the original author, or if it doesn't exist, a notice (hypertext is preferred, but a textual representation is permitted) of the form: "Copyright 2019, TIP and its Contributors. All rights Reserved"
3. When space permits, inclusion of the full text of this License should be provided. We request that authorship attribution be provided in any software, documents, or other items or products that you create pursuant to the implementation of the contents of this document, or any portion thereof.

No right to create modifications or derivatives of TIP documents is granted pursuant to this License. except as follows: To facilitate implementation of software or specifications that may be the subject of this document, anyone may prepare and distribute derivative works and portions of this document in such implementations, in supporting materials accompanying the implementations, PROVIDED that all such materials include the copyright notice above and this License. HOWEVER, the publication of derivative works of this document for any other purpose is expressly prohibited.

For the avoidance of doubt, Software and Specifications, as those terms are defined in TIP's Organizational Documents (which may be accessed at




<https://telecominfraproject.com/organizational-documents/>), and components thereof incorporated into the Document are licensed in accordance with the applicable Organizational Document(s).

12 Disclaimers

THIS DOCUMENT IS PROVIDED "AS IS," AND TIP MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

TIP WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

The name or trademarks of TIP may NOT be used in advertising or publicity pertaining to this document or its contents without specific, written prior permission. Title to copyright in this document will at all times remain with TIP and its Contributors. This TIP Document License is based, with permission from the W3C, on the W3C Document License which may be found at <https://www.w3.org/Consortium/Legal/2015/doc-license.html>.



Copyright © 2020 Telecom Infra Project, Inc. A TIP Participant, as that term is defined in TIP's Bylaws, may make copies, distribute, display or publish this Specification solely as needed for the Participant to produce conformant implementations of the Specification, alone or in combination with its authorized partners. All other rights reserved.

The Telecom Infra Project logo is a trademark of Telecom Infra Project, Inc. (the "Project") in the United States or other countries and is registered in one or more countries. Removal of any of the notices or disclaimers contained in this document is strictly prohibited.



TELECOM INFRA PROJECT