ETSI Plugtests Test Plan V1.0 (2019-01)

1st mWT SDN Plugtests Event
Sophia Antipolis, France
21 – 24 January 2019

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex – FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 – NAF 742 C
Association à but non 2tandardi enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Copyright Notification

Contents

List of Figures

1 Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

2 Scope

The present document defines a test plan with the purpose of supporting the first mWT Plugtests™ event. It contains:

- *a conventions clause summarizing all pro-formas and common rules for conduction the Plugtests event;*
- *the overall architecture describing the network including controllers, interfaces and applications;*
- *the configurations (CFG) summarizing the valid configurations derived from the overall architecture. A valid configuration is a specific subset of the overall architecture to which a given group of test descriptions applies used during test sessions;*
- *the Test Summary listing all test objectives. A Test Description (TD) will be developed for each test objective.*
- *the Test Descriptions (TD) compiling all the information required to execute a test. They describe all the steps required to achieve a test objective;*
- *the Interoperability Feature Statements (IFS) identifying the features which a Device Under Test (DUT) supports, including those which are optional and those which are conditional on the support of other features. The IFS are used to select applicable TDs for each test session.*

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references,only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Informative references

The following referenced documents assist the user with regard of the Plugtests preparation.

[1] I2RS Topology Model: https://tools.ietf.org/html/rfc8345

[2] TE Topology Model: https://tools.ietf.org/html/draft-ietf-teas-yang-te-topo

[3] MW Topology Model: https://tools.ietf.org/html/draft-ye-ccamp-mw-topo-yang

[4] Ethernet Topology Model: https://tools.ietf.org/html/draft-zheng-ccamp-client-topo-yang

[5] Ethernet Service Model: https://tools.ietf.org/html/draft-zheng-ccamp-otn-client-signal-yang

[6] Restconf protocol: https://tools.ietf.org/html/rfc8040

[7] YANG Module Library: https://tools.ietf.org/html/rfc7895

[8] Plugtest Wiki: https://wiki.plugtests.net/wiki/mWT-Plugtests/index.php/Main_Page

[9] Code Forge repository: https://forge.etsi.org/gitlab/sdn/mwt

[10] Working documents referenced in this document: https://wiki.plugtests.net/wiki/mWT-Plugtests/index.php/Testing_Information

Note: for the standards in draft status, the version mentioned in Annex C shall be used as a baseline. If the most recent version is published too late for a Participant to be implemented for this Plugtest, it may be accepted that the Domain Controller complies with an earlier version.

3 Definitions and Abbreviations

3.1 Definitions

Def1 TODO

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

DC	Domain Controller
DUT	Device Under Test
GE	Gigabit Ethernet
IFS	Interoperability Feature Statement
mmW	Millimetre wave
MW	Microwave
NBI	Northbound Interface
NE	Network Element
NMS	Network Management System
SBI	Southbound Interface
SDN	Software Defined Network
TD	Test Description

4 Conventions

4.1 Common Rules

- The Domain Controller and its physical domain shall be provided as a unified bundle. A bundle may be formed by more than one component provider, but they shall be in agreement and jointly represent a single point of responsibility towards the Plugtest.

- Only one instance of the Postman software tool will be used to manage the network via the NBI.

4.2 Test Description pro-forma

Test Descriptions compile all the information required to execute a test. They describe all the steps required to achieve a test objective. The following information is provided with each Test Description:

- *Identifier: A unique identifier is assigned to each Test Description. The usage of a well-defined naming convention allowing to put the TD into context (Functional Group, Feature, etc.) is recommended.*
- *Test Objective: Description of the objective of the TD (what).*
- *Configuration: Reference to the applicable configuration(s).*
- *References: Reference to the base specification(s) which describe the feature being tested.*
- *Applicability: List of items in the IFS that need to be supported by the DUTs in order to be able to execute the test.*
- *Pre-test conditions: Specific conditions that need to be met by the DUT prior to start executing the test sequence. It can include information about configuration, and/or initial state of the DUT.*
- *Test Sequence: Detailed description of the steps that are to be followed in order to achieve the stated test purpose. These steps are specified in a clear and unambiguous way but without placing unreasonable restrictions on how the step is performed. Clarity and precision are important to ensure that the step can followed exactly. The lack of restrictions is necessary to ensure that the test can apply to a range of different types of implementation.*

Table 1: Test Description pro-forma

Interoperability Test Description			
Identifier	*Unique test description ID: TD_AB_XXX_00. Follows a well-defined naming convention*		
Test Objective	*a concise summary of the test reflecting its purpose and allowing readers to easily distinguish this test from any other test in the document*		
Configuration	*Reference to the applicable configuration*		
References	*List of references to the base specification clause(s), use case(s), requirement(s), etc. which are either used in the test or define the functionality being tested*		
Applicability	*List of features and capabilities in the IFS which are required to be supported by the DUTs in order to execute this test*		
Pre-test conditions	*List of test specific pre-conditions that need to be met by the DUT including information about configuration, i.e. precise description of the initial state of the DUTs prior to start executing the test sequence* • •		
Test Sequence	**Step**	**Type**	**Description**
	1	***<Request>***	*Step description*
	2		
	3		
	4		
	5		
	6		

The Steps in the Test Sequence can be of different type, depending on their purpose:

- *A stimulus corresponds to an event that triggers a specific action on a FUT, like sending a message for instance;*
- *A configure corresponds to an action to modify the FUT or SUT configuration;*
- *An IOP check consists of observing that one FUT behaves as described in the standard: i.e. resource creation, update, deletion, etc. For each IOP check in the Test Sequence, a result can be recorded;*
- *The overall IOP Verdict will be considered OK if all the IOP checks in the sequence are OK.*

4.3 Interoperability Feature Statement (IFS)

The Interoperable Feature Statement (IFS) identifies the 9tandardized features of a DUT. These features can be mandatory, optional or conditional (depending on other features), and depend on the role played by the DUT. The IFS can also be used as a pro-forma by a vendor to identify the features that its DUT will support when interoperating with corresponding features from other vendors. The annex of the present document defines the IFS.

5 Architecture

5.1 Reference SDN Architecture

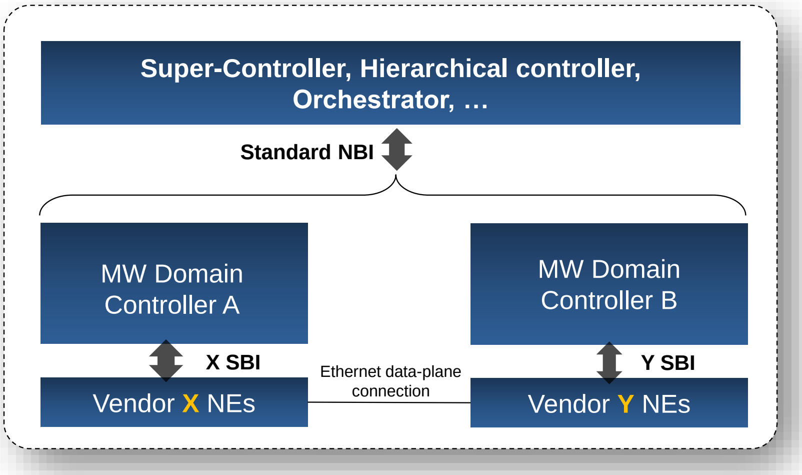


Figure 1 Generic Multi-Domain, Multi-Vendor SDN Architecture

With reference to Figure 1, This Plugtest focuses on the DC's NBI, regardless of the specific overall architecture choices made in layers above the Domain Controller.

Also, as an explicit choice, nothing is specified or required regarding the DC's SBI (i.e. the interface between a DC and its managed NEs), regarding protocols, data models etc.

The basis for the definition of the NBI whose interoperability shall be tested by this Plugtest are the use of the Restconf protocol (RFC 8040) and the YANG DM library provided by IETF (RFCs and relevant drafts, as specified in Annex C IETF Data Model Selection).

As depicted in Figure 2, In order to simplify the test specification and implementation, the interoperability testing will by unanimous agreement of the mWT ISG be performed by using an API Development and Testing environment, namely the Postman system.

Tests will be performed by exploiting the automation (scripting) capability of Postman, with a single set of scripts being jointly developed specifically for this Plugtest by the Participants and stored in the Plugtest's Forge code repository [9].

Specifying a single set of scripts and the expected format and content of the related responses by the DCs, it will be possible to univocally determine the compliance of the DCs to the relevant standards and confirm the multi-domain interoperability of the systems under test and the specified NBI.

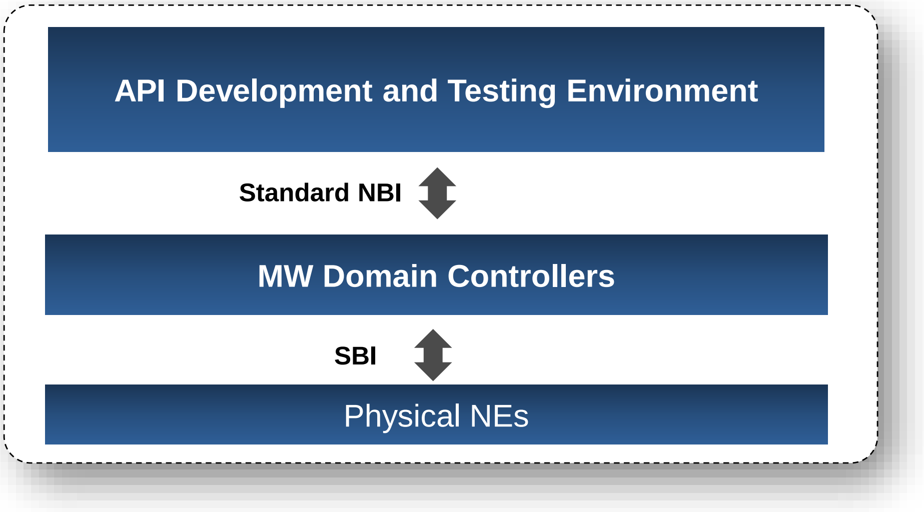


Figure 2 PlugTest SDN Architecture

5.2 Test Network Architecture

5.2.1 Logical Topology

Figure 3 shows the logical topology of the Plugtest network.

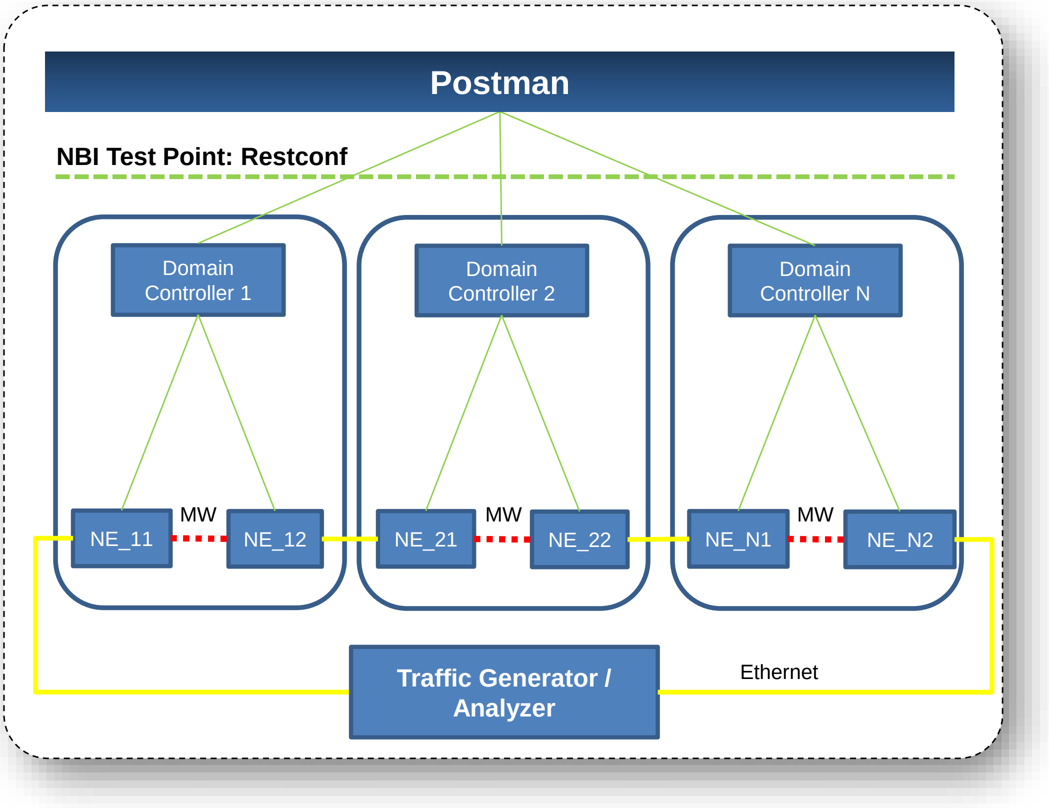


Figure 3 Logical Topology of the Test Network

- Each Domain contains exactly one MW (mmW) link. The physical connection between the two RF units of the radio link within one domain shall be realized with coaxial cable or waveguide plus attenuators, no antenna and no free space radiation is allowed
- MW links are arranged in a linear topology
- Each link is connected to the adjacent one via an Ethernet cable, the first and the last NE in the total chain are connected to a packet traffic generator / analyzer via Ethernet cables
- The connection between the Domain controller and its MW link is internal to the domain and completely taken care for by the respective Participant
- All naming of attributes is indexed to the Domain “number” (1 to N for a total of N Domains) in order to simplify the script execution

- Figure 4 and Figure 5 show the general and detailed physical structure of the test network.

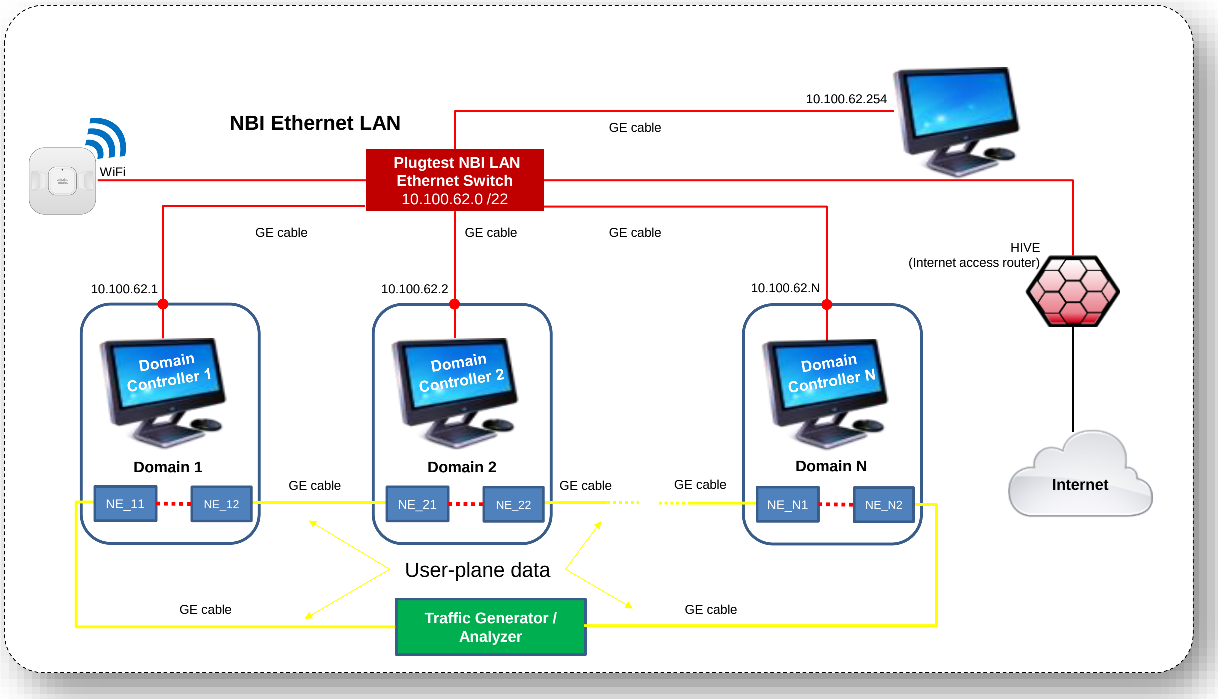


Figure 4 Physical Structure of the Test Network (All Domains Connected)

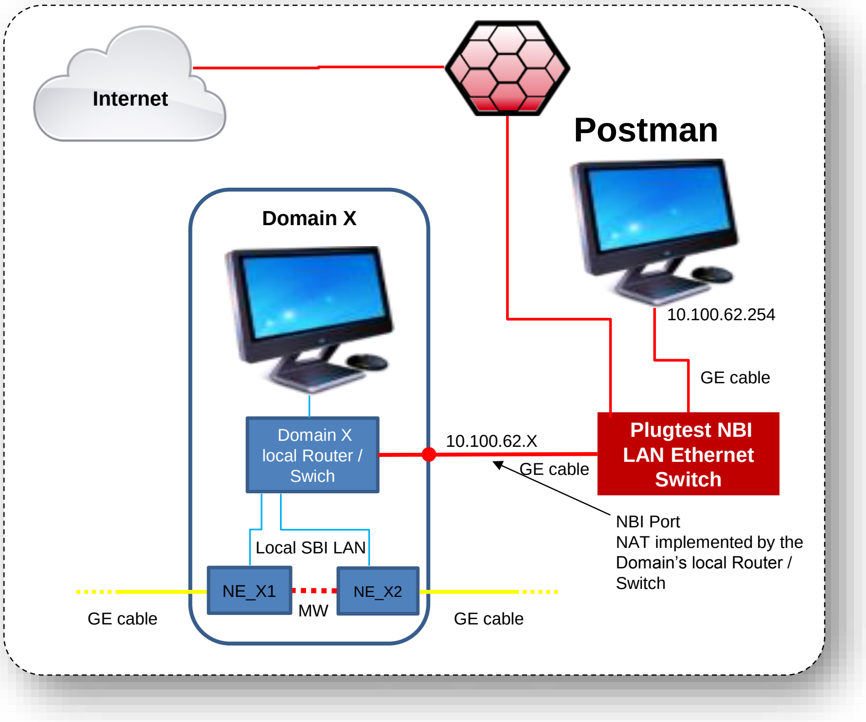

Internet
Postman
Domain X
Domain X local Router / Swich
10.100.62.254
GE cable
10.100.62.X
GE cable
Plugtest NBI LAN Ethernet Switch
NBI Port NAT implemented by the Domain's local Router / Switch
Local SBI LAN
NE_X1
NE_X2
MW
GE cable
GE cable

Figure 5 Physical Structure of the Test Network (Detail of one Domain)

- The LAN used to connect the Domain Controllers to the Postman is a simple Ethernet LAN with a single Ethernet switch
- The NBI IP addressing plan is static, based on private IP (e.g. 10.100.62.X, where X is the given Domain unique assigned number – from 1 to N if there is a total of N Domains)
- NAT is used at the Domains' NBI Port, in order to completely isolate the Postman NBI LAN
- The Domain's own Router / Switch is complete responsibility of the Domain's owner
- The single Postman instance used for all testing is running on a common, dedicated computer. This computer is connected via Ethernet cable to the NBI LAN (with internet access for Plugtest's Forge [9], Wiki [8] etc.)

5.2.2 IP Addressing

All IP addresses (controllers and equipment) are assigned statically. Specifically, ETSI's DHCP server will not assign any address in the range 10.100.62.0 to 10.100.62.254 on the test network.

Participant	Static range	NBI IP	Microwave m-plane IP *
Unassigned	10.100.62.10/22 to .19 10.100.62.110/22 to .119	10.100.62.10/22	10.100.62.11/22 10.100.62.12/22
Ceragon	10.100.62.20/22 to .29 10.100.62.120/22 to .129	10.100.62.20/22	10.100.62.21/22 10.100.62.22/22
Ericsson	10.100.62.30/22 to .39 10.100.62.130/22 to .139	10.100.62.30/22	10.100.62.31/22 10.100.62.32/22
Huawei	10.100.62.40/22 to .49 10.100.62.140/22 to .149	10.100.62.40/22	10.100.62.41/22 10.100.62.42/22

Participant	Static range	NBI IP	Microwave m-plane IP *
Intracom	10.100.62.50/22 to .59 10.100.62.150/22 to .159	10.100.62.50/22	10.100.62.51/22 10.100.62.52/22
NEC	10.100.62.60/22 to .69 10.100.62.160/22 to .169	10.100.62.60/22	10.100.62.61/22 10.100.62.62/22
Nokia	10.100.62.70/22 to .79 10.100.62.170/22 to .179	10.100.62.70/22	10.100.62.71/22 10.100.62.72/22
SIAE	10.100.62.80/22 to .89 10.100.62.180/22 to .189	10.100.62.80/22	10.100.62.81/22 10.100.62.82/22
Spirent TestCenter Laptop**	10.100.62.253/22	-	-
Postman	10.100.62.254/22	-	-

Table 1 IP Addressing Plan

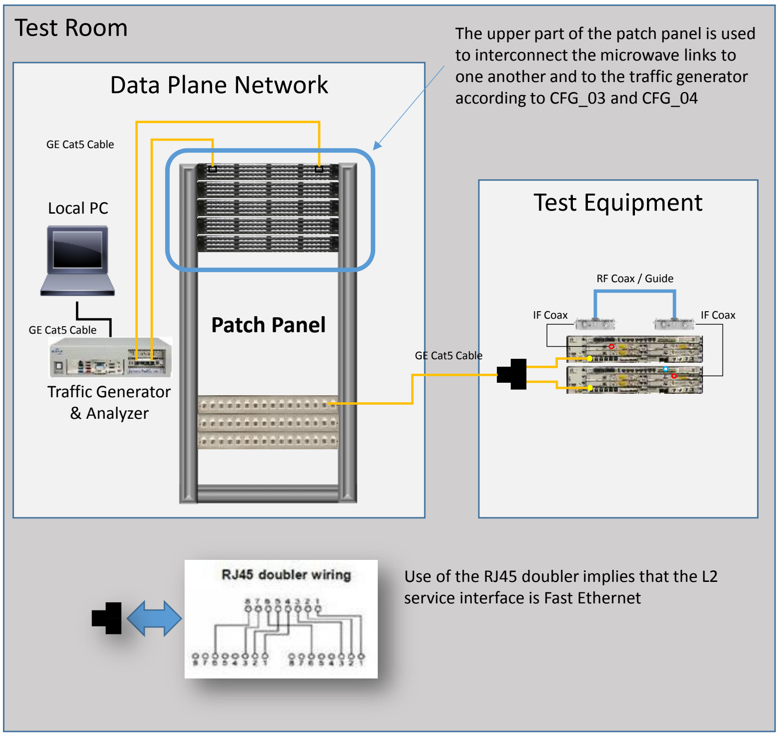
(*) Note: the IP address of each microwave radio unit is indicated for completeness, in case the SBI need to be connected via the same test network as the NBI.

(**) Note: the IP address of the computer used to manage the TestCenter is defined in case that computer needs access to internet or a local printer etc. in principle, that laptop does not need to be connected to the test network.

5.2.3 Data Plane Network

The test cases that create and delete a L2 service foresee the use of a Test Instrument (Spirent TestCenter C1) to generate the traffic, and to confirm that it is flowing correctly when the circuit is set up.

This data-plane network is closed, i.e. not connected to any other network (test network, internet etc.)



The upper part of the patch panel is used to interconnect the microwave links to one another and to the traffic generator according to CFG_03 and CFG_04

Use of the RJ45 doubler implies that the L2 service interface is Fast Ethernet

Figure 6 Data Plane Physical Interconnections

The use of the RJ-45 doubler, needed to connect to the portable patch panel provided by ETSI CTI, means that the physical interface type must be 100Base-T.

The L2 service is configured as reported in Annex D.3 Ethernet Service.

5.3 Data Model Architecture

The IETF RFC and draft DMs to be used in this Plugtest are specified in Annex C IETF Data Model Selection, including the selection of the required subsets of attributes defined therein.

Figure 7, Figure 8 and Figure 9 depict a simplified DM topology overview as shall be used in this Plugtest.

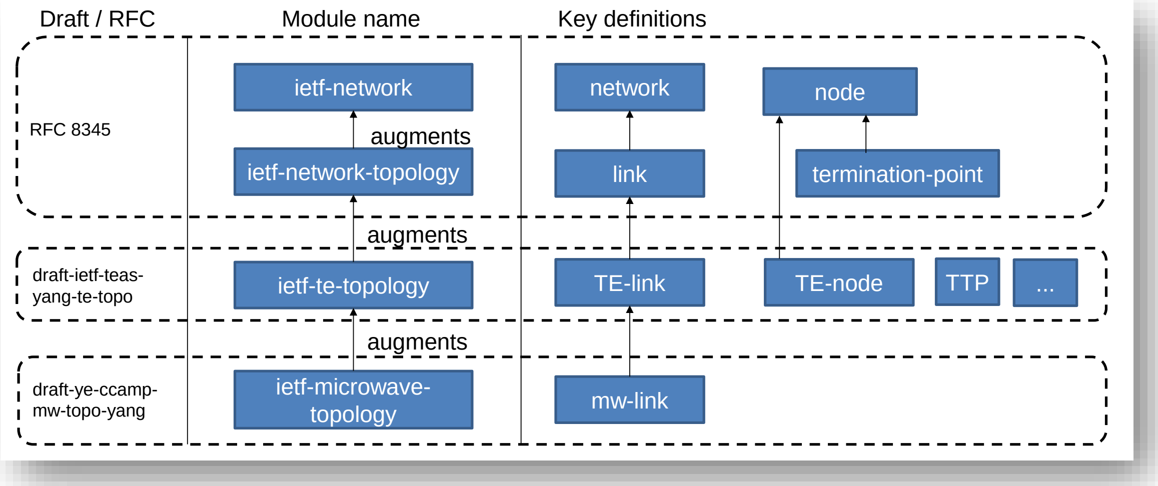


Figure 7 IETF Microwave Topology Models

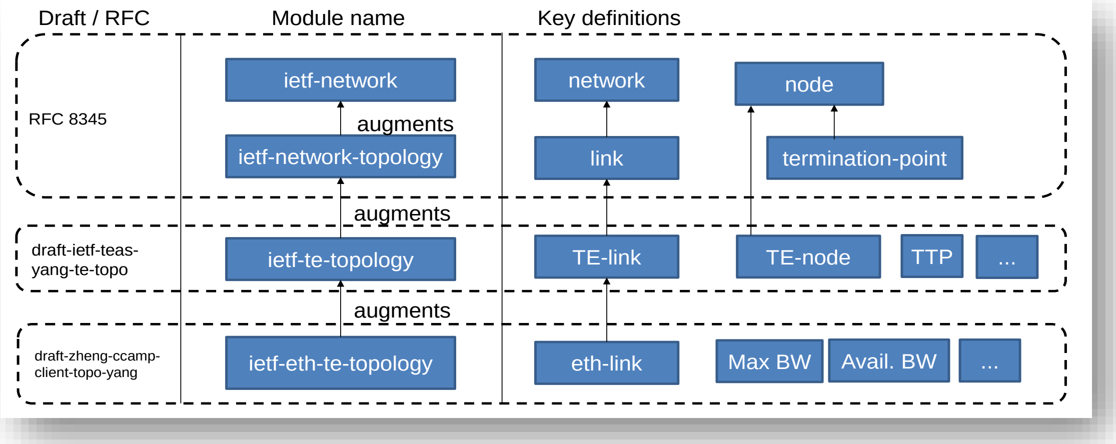


Figure 8 IETF Ethernet Topology Models

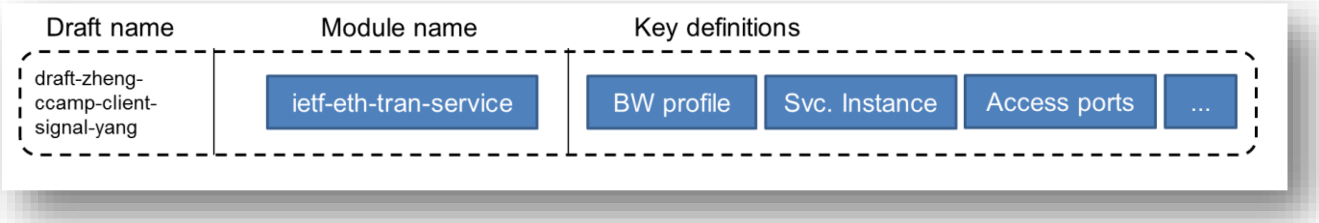


Figure 9 IETF Ethernet Service Model

5.4 Reference Topology Models

5.4.1 Multi-domain Physical Topology

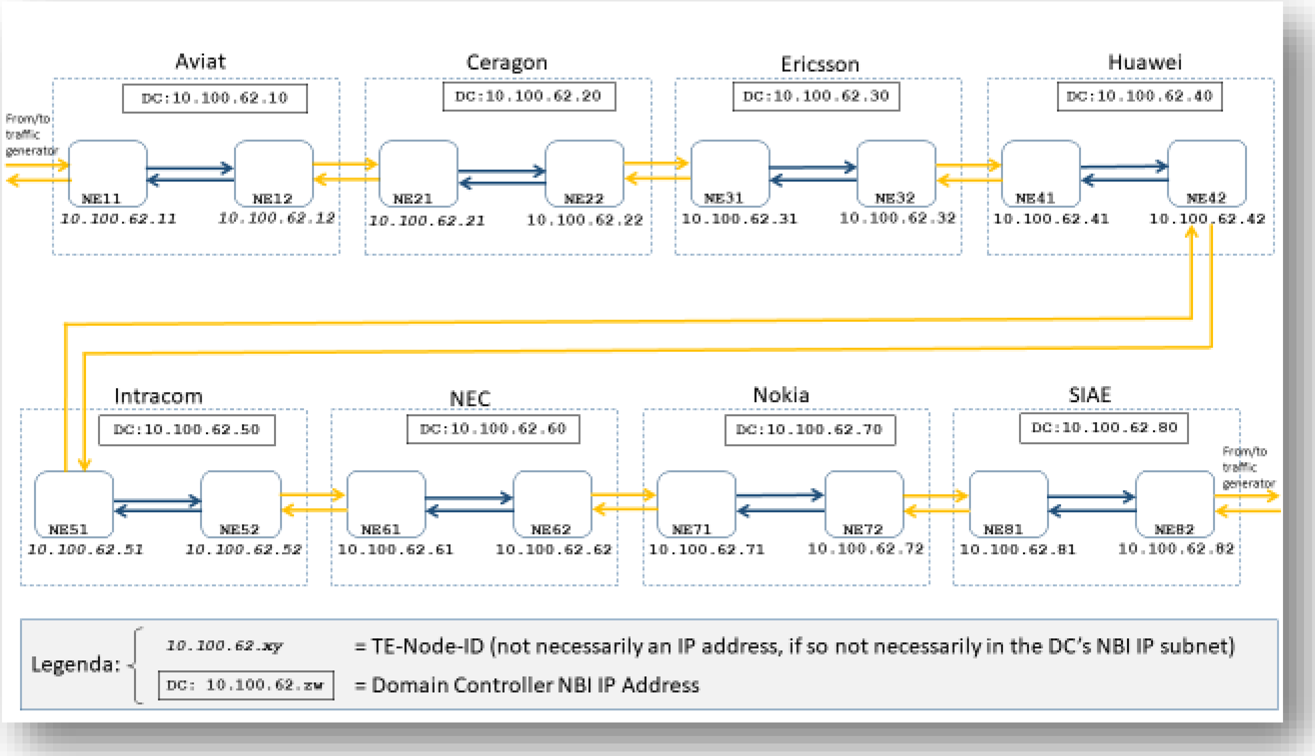


Figure 10 Multi-domain Physical Topology

5.4.2 Single Domain Topology Exposed on NBI

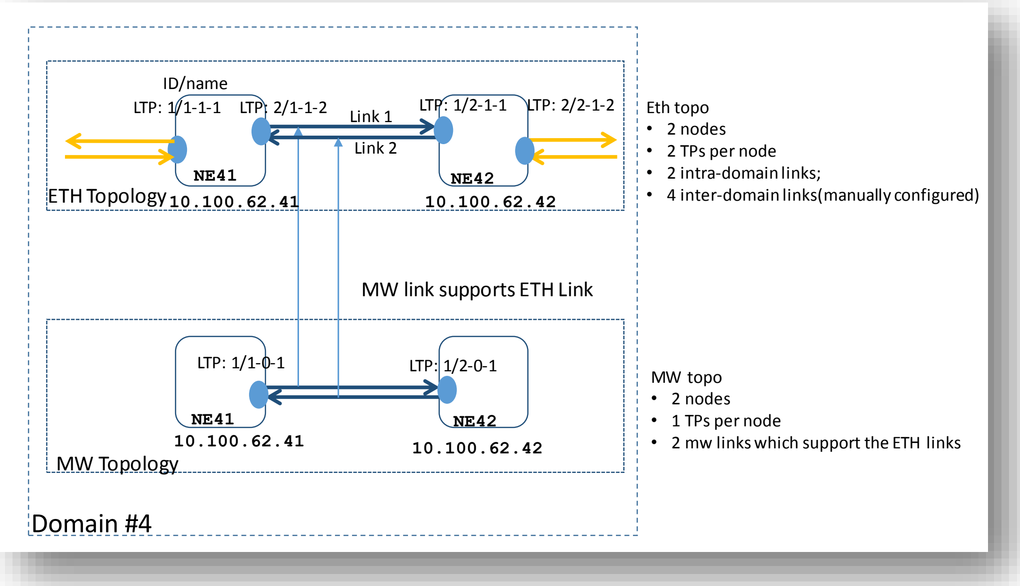


Figure 10 Single Domain Topology Exposed on NBI

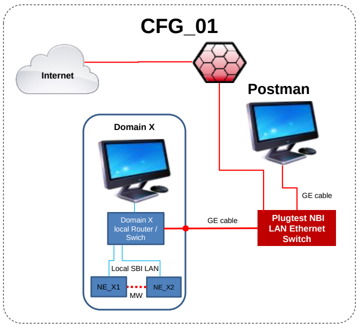
Note 1: This picture describes Huawei's implementation, for other Participants' implementation-specific variants, please refer to Annex B.

Note 2: The inter-domain links' information is not requested to be published across the NBI for this Plugtest.

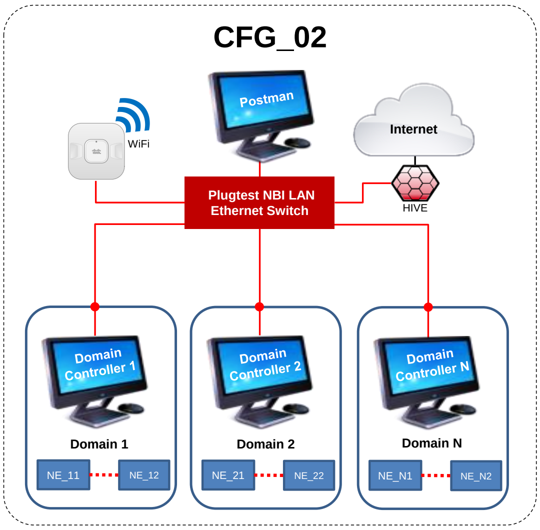
6 Configurations

The configurations applicable to the Basic Tests are:

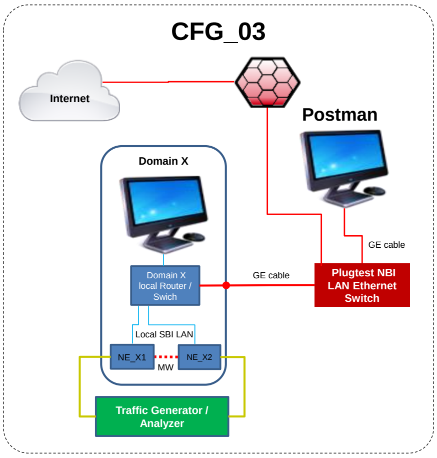
- CFG_01
  - A single Domain Controller is connected to the Postman system. This is necessary to test and troubleshoot each single DC without interactions with other systems.



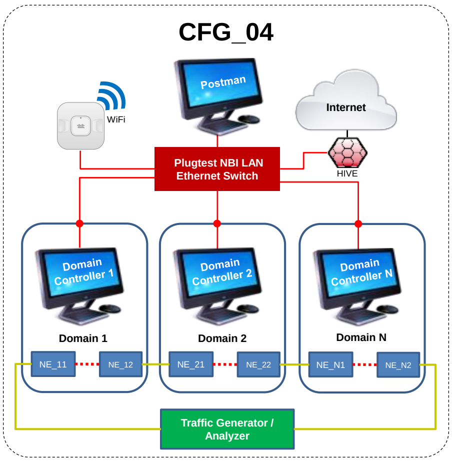
- CFG_02
  - All DCs are connected to the NBI LAN and the Postman system at the same time. This allows to perform the inter-domain interoperability tests.



- CFG_03
  - A single Domain Controller is connected to the Postman system. This is necessary to test and troubleshoot each single DC without interactions with other systems.
  - The traffic generator/analyzer is connected to the MW link under test to verify on the single DC that all use cases are tested corrected individually.


CFG_03
Internet
Postman
Domain X
Domain X local Router / Swich
GE cable
GE cable
Plugtest NBI LAN Ethernet Switch
Local SBI LAN
NE_X1
MW
NE_X2
Traffic Generator / Analyzer

- CFG_04
  - All DCs are connected to the NBI LAN and the Postman system at the same time. This allows to perform the inter-domain interoperability tests.
  - The MW links are connected to each other at the user traffic level (GE ports of the tributary cards of the radio Nes) in a linear chain (one link per domain), according to a fixed sequence and numbering scheme as per test topology specification.
  - The traffic generator/analyzer is connected to the first and the last MW links under test, to verify the inter-domain creation and deletion of the L2 data service as per test specification



7 Test Summary

7.1 Single Domain Network and Service Discovery (SNSD) Tests

7.1.1 Applicable configurations

The configurations applicable to the Basic Tests are:

- CFG_01

7.1.2 List of objectives

Table 2: NSD Test Objectives

Test ID	Objective
TD_SDN_SNSD_01	Issue a request via Postman to one individual domain controller, to check it's reachability and basic functionality.
TD_SDN_SNSD_02	The microwave topology information are requested from a single DC. The received information is compared to a template and checked for compliance.
TD_SDN_SNSD_03	The Ethernet topology information are requested from a single DC. The received information is compared to a template and checked for compliance.

7.2 Multi-Domain Network and Service Discovery (MNSD) Tests

7.2.1 Applicable configurations

The configurations applicable to the Basic Tests are:

- CFG_02

7.2.2 List of objectives

Table 3: NSD Test Objectives

Test ID	Objective
TD_SDN_MNSD_01	Issue a request via Postman to all domain controllers, to check the overall viability of the test network.
TD_SDN_MNSD_02	The microwave topology information are requested from all DCs connected to the NBI LAN at the same time for the same information as TD_SDN_SNSD_02. This allows to check that the connectivity to all DC is fully functional. A comparison of the answers received may be performed to check consistency and compliance.
TD_SDN_MNSD_03	The Ethernet topology information are requested from all DCs connected to the NBI LAN at the same time for the same information as TD_SDN_SNSD_03. This allows to check that the connectivity to all DC is fully functional. A comparison of the answers received may be performed to check consistency and compliance.

7.3 Single-Domain L2 service provisioning (SSP) Tests

7.3.1 Applicable configurations

The configurations applicable to the Basic Tests are:

- CFG_03

7.3.2 List of objectives

Table 4: ISP Test Objectives

Test ID	Objective
TD_SDN_SSP_01	Create the specified L2 data service over a single domain. The traffic generator/analyzer confirms that data start flowing.
TD_SDN_SSP_02	The Ethernet service information is requested from the single DC under test in TD_SDN_SSP_01. The received information is checked to correctly list the newly created service.
TD_SDN_SSP_03	Delete the specified L2 data service over a single domain. The traffic generator/analyzer confirms that data stops flowing.
TD_SDN_SSP_04	The Ethernet service information is requested from the single DC under test in TD_SDN_SSP_03. The received information is checked to correctly not list the newly deleted service anymore.

7.4 Inter-domain L2 service provisioning (ISP) Tests

7.4.1 Applicable configurations

The configurations applicable to the Basic Tests are:

- CFG_04

7.4.2 List of objectives

Table 5: ISP Test Objectives

Test ID	Objective
TD_SDN_ISP_01	Create the specified L2 data service over all available domains. The traffic generator/analyzer confirms that data start flowing.
TD_SDN_ISP_02	The Ethernet service information is requested from all the DCs under test in TD_SDN_ISP_01. The received information is checked to correctly list the newly created service.
TD_SDN_ISP_03	Delete the specified L2 data service over all available domains. The traffic generator/analyzer confirms that data stops flowing.
TD_SDN_ISP_04	The Ethernet service information is requested from all the DCs under test in TD_SDN_ISP_03. The received information is checked to correctly not list the newly deleted service anymore.

7.5 Initialization Procedure

7.5.1 Applicable configurations

Not relevant.

7.5.2 List of objectives

Table 6: Initialization Test Objectives

Test ID	Objective
TD_SDN_INIT	Create and initialize the TD_SDN_SETTINGS object in the runtime environment of Postman. This needs to be done only once at the beginning of a test session with Postman.

8 Test Descriptions

8.1 Network and Service Discovery Test Descriptions

<table>
<tr><th colspan="4">Interoperability Test Description</th></tr>
<tr><td><b>Identifier</b></td><td colspan="3">TD_SDN_SNSD_01</td></tr>
<tr><td><b>Test Objective</b></td><td colspan="3">Issue a request via Postman to one individual domain controller, to check it's reachability and basic functionality.</td></tr>
<tr><td><b>Configuration</b></td><td colspan="3">CFG_01</td></tr>
<tr><td><b>References</b></td><td colspan="3"></td></tr>
<tr><td><b>Applicability</b></td><td colspan="3">MW_8040, MW_8345</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td><b>Pre-test conditions</b></td><td colspan="3">• Postman has been correctly initialized earlier, by executing TD_SDN_INIT
• The Domain Controller instance is up and running normally
• All the devices are upgraded to correct versions
• All basic configurations are completed (e.g., NE_id, OSPF, PCEP, etc.)</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td><b>Test Sequence</b></td><td><b>Step</b></td><td><b>Type</b></td><td><b>Description</b></td></tr>
<tr><td></td><td>1</td><td><b><i>Request</i></b></td><td><i>Send GET request via Postman to one individual domain controller by executing Collection TD_SDN_SNSD_01 (see <b>Error! Reference source not found.</b>)</i></td></tr>
<tr><td></td><td>2</td><td><b><i>Validation</i></b></td><td><i>Check the response body of the above request and confirm if the Restconf server is serviceable. See Annex D.</i></td></tr>
<tr><td></td><td>3</td><td><b><i>Validation</i></b></td><td><i>The response body of the request should contain a list of all YANG modules and submodules used by the Restconf server along with information about name and revision for each module.</i></td></tr>
<tr><td></td><td>4</td><td><b><i>Validation</i></b></td><td><i>The response body of each query should contain the specified YANG module along with its name and revision. See Annex D.</i></td></tr>
</table>

<table>
<tr><th colspan="4">Interoperability Test Description</th></tr>
<tr><td><b>Identifier</b></td><td colspan="3">TD_ SDN_MNSD_01</td></tr>
<tr><td><b>Test Objective</b></td><td colspan="3">Issue a request via Postman to all domain controllers, to check the overall viability of the test network.</td></tr>
<tr><td><b>Configuration</b></td><td colspan="3">CFG_02</td></tr>
<tr><td><b>References</b></td><td colspan="3"></td></tr>
<tr><td><b>Applicability</b></td><td colspan="3">MW_8040, MW_8345</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td><b>Pre-test conditions</b></td><td colspan="3">• Postman has been correctly initialized earlier, by executing TD_SDN_INIT
• All the Domain Controller instances are up and running normally
• All the devices are upgraded to correct versions
• All basic configurations are completed (e.g., NE_id, OSPF, PCEP, etc.)</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td><b>Test Sequence</b></td><td><b>Step</b></td><td><b>Type</b></td><td><b>Description</b></td></tr>
<tr><td></td><td>1</td><td><b><i>Request</i></b></td><td><i>Send GET request via Postman to all domain controllers by executing Collection TD_SDN_MNSD_01 (see </i>Error! Reference ource not found.<i>)</i></td></tr>
<tr><td></td><td>2</td><td><b><i>Validation</i></b></td><td><i>Check the response body of each request and confirm if all the Restconf servers are serviceable. See Annex D.</i></td></tr>
<tr><td></td><td>3</td><td><b><i>Validation</i></b></td><td><i>The response body of the request should contain a list of all YANG modules and submodules used by the Restconf server along with information about name and revision for each module.</i></td></tr>
<tr><td></td><td>4</td><td><b><i>Validation</i></b></td><td><i>The response body of each query should contain the specified YANG module along with its name and revision. See Annex D.</i></td></tr>
</table>

Interoperability Test Description			
Identifier	TD_ SDN_SNSD_02		
Test Objective	The microwave topology information are requested from a single DC. The received information is compared to a template and checked for compliance.		
Configuration	CFG_01		
References			
Applicability	MW_8040, MW_8345, MW_TETOPO, MW_MWTOPO		
Pre-test conditions	• Postman has been correctly initialized earlier, by executing TD_SDN_INIT • The Domain Controller instance is up and running normally • All the devices are upgraded to correct versions • All basic configurations are completed (e.g., NE_id, OSPF, PCEP, etc.) • The Restconf server is serviceable.		
Test Sequence	**Step**	**Type**	**Description**
	1	***Request***	*Send GET request via Postman to one individual domain controller by executing Collection TD_SDN_SNSD_02 (see Error! eference source not found.)*
	2	***Validation***	*The response body should contain information about the microwave topology in JSON format as specified in Annex D.*

Interoperability Test Description			
Identifier	TD_ SDN_SNSD_03		
Test Objective	The Ethernet topology information are requested from a single DC. The received information is compared to a template and checked for compliance.		
Configuration	CFG_01		
References			
Applicability	MW_8040, MW_8345, MW_ETHSVC		
Pre-test conditions	• Postman has been correctly initialized earlier, by executing TD_SDN_INIT • The Domain Controller instance is up and running normally • All the devices are upgraded to correct versions • All basic configurations are completed (e.g., NE_id, OSPF, PCEP, etc.) • The Restconf server is serviceable.		
Test Sequence	**Step**	**Type**	**Description**
	1	***Request***	*Send GET request via Postman to one individual domain controller by executing Collection TD_SDN_SNSD_03 (see **Error! Reference source not found.**)*
	2	***Validation***	*The response body should contain information about the Ethernet topology in JSON format as specified in Annex D.*

Interoperability Test Description			
Identifier	TD_ SDN_MNSD_02		
Test Objective	The microwave topology information are requested from all DCs connected to the NBI LAN at the same time for the same information as TD_SDN_SNSD_02. This allows to check that the connectivity to all DC is fully functional. A comparison of the answers received may be performed to check consistency and compliance.		
Configuration	CFG_02		
References			
Applicability	MW_8040, MW_8345, MW_TETOPO, MW_MWTOPO, MW_ETHSVC		
Pre-test conditions	• Postman has been correctly initialized earlier, by executing TD_SDN_INIT • All the Domain Controller instances are up and running normally • All the devices are upgraded to correct versions • All basic configurations are completed (e.g., NE_id, OSPF, PCEP, etc.) • All Restconf servers are serviceable.		
Test Sequence	**Step**	**Type**	**Description**
	1	***Request***	*Send GET request via Postman to all domain controllers by executing Collection TD_SDN_MNSD_02 (see Error! Reference ource not found.)*
	2	***Validation***	*The response body of each request should contain information about the microwave topology in JSON format as specified in Annex D.*

Interoperability Test Description			
Identifier	TD_ SDN_MNSD_03		
Test Objective	The Ethernet topology information are requested from all DCs connected to the NBI LAN at the same time for the same information as TD_SDN_SNSD_03. This allows to check that the connectivity to all DC is fully functional. A comparison of the answers received may be performed to check consistency and compliance.		
Configuration	CFG_02		
References			
Applicability	MW_8040, MW_8345, MW_ETHSVC		
Pre-test conditions	• Postman has been correctly initialized earlier, by executing TD_SDN_INIT • All the Domain Controller instances are up and running normally • All the devices are upgraded to correct versions • All basic configurations are completed (e.g., NE_id, OSPF, PCEP, etc.) • All Restconf servers are serviceable.		
Test Sequence	**Step**	**Type**	**Description**
	1	***Request***	*Send GET request via Postman to all domain controllers by executing Collection TD_SDN_MNSD_03 (see Error! Reference ource not found.)*
	2	***Validation***	*The response body of each request should contain information about the Ethernet topology in JSON format as specified in Annex D.*

8.2 Service Provisioning Test Descriptions

<table>
<tr><th colspan="4">Interoperability Test Description</th></tr>
<tr><td><b>Identifier</b></td><td colspan="3">TD_ SDN_SSP_01</td></tr>
<tr><td><b>Test Objective</b></td><td colspan="3">Create the specified L2 data service over a single domain. The traffic generator/analyzer confirms that data start flowing.</td></tr>
<tr><td><b>Configuration</b></td><td colspan="3">CFG_03</td></tr>
<tr><td><b>References</b></td><td colspan="3"></td></tr>
<tr><td><b>Applicability</b></td><td colspan="3">MW_8040, MW_8345, MW_TETOPO, MW_MWTOPO, MW_ETHSVC</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td><b>Pre-test conditions</b></td><td colspan="3">• Postman has been correctly initialized earlier, by executing TD_SDN_INIT
• The Domain Controller instance is up and running normally
• All the devices are upgraded to correct versions
• All basic configurations are completed (e.g., NE_id, OSPF, PCEP, etc.)
• The Restconf server is serviceable.</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td><b>Test Sequence</b></td><td><b>Step</b></td><td><b>Type</b></td><td><b>Description</b></td></tr>
<tr><td></td><td>1</td><td><b><i>Request</i></b></td><td><i>Send POST request via Postman to one individual domain controller by executing Collection TD_ SDN_SSP_01. (see <b>Error! eference source not found.</b>)</i></td></tr>
<tr><td></td><td>2</td><td><b><i>Validation</i></b></td><td><i>Check the traffic generator/analyzer if the data start flowing properly.</i></td></tr>
</table>

<table>
<tr><th colspan="4">Interoperability Test Description</th></tr>
<tr><td><b>Identifier</b></td><td colspan="3">TD_ SDN_SSP_02</td></tr>
<tr><td><b>Test Objective</b></td><td colspan="3">The Ethernet service information is requested from the single DC under test in TD_SDN_SSP_01. The received information is checked to correctly list the newly created service.</td></tr>
<tr><td><b>Configuration</b></td><td colspan="3">CFG_03</td></tr>
<tr><td><b>References</b></td><td colspan="3"></td></tr>
<tr><td><b>Applicability</b></td><td colspan="3">MW_8040, MW_8345, MW_TETOPO, MW_MWTOPO, MW_ETHSVC</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td><b>Pre-test conditions</b></td><td colspan="3">• Postman has been correctly initialized earlier, by executing TD_SDN_INIT
• The Domain Controller instance is up and running normally
• All the devices are upgraded to correct versions
• All basic configurations are completed (e.g., NE_id, OSPF, PCEP, etc.)
• The Restconf server is serviceable.</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td><b>Test Sequence</b></td><td><b>Step</b></td><td><b>Type</b></td><td><b>Description</b></td></tr>
<tr><td></td><td>1</td><td><b><i>Request</i></b></td><td><i>Send GET request via Postman to one individual domain controller by executing Collection TD_ SDN_SSP_02 (see Error! eference source not found.)</i></td></tr>
<tr><td></td><td>2</td><td><b><i>Validation</i></b></td><td><i>The response body should contain information about the created L2 service in JSON format referring to Annex D.<b>Error! Reference ource not found.</b></i></td></tr>
</table>

Interoperability Test Description			
Identifier	TD_ SDN_SSP_03		
Test Objective	Delete the specified L2 data service over a single domain. The traffic generator/analyzer confirms that data stops flowing.		
Configuration	CFG_03		
References			
Applicability	MW_8040, MW_8345, MW_TETOPO, MW_MWTOPO, MW_ETHSVC		
Pre-test conditions	• Postman has been correctly initialized earlier, by executing TD_SDN_INIT • The Domain Controller instance is up and running normally • All the devices are upgraded to correct versions • All basic configurations are completed (e.g., NE_id, OSPF, PCEP, etc.) • The Restconf server is serviceable.		
Test Sequence	**Step**	**Type**	**Description**
	1	***Request***	*Send DELETE request via Postman to one individual domain controller by executing Collection TD_ SDN_SSP_03 (see* **Error! Reference source not found.***)*
	2	***Validation***	*Check the traffic generator/analyzer if the data stop flowing.*

Interoperability Test Description			
Identifier	TD_ SDN_SSP_04		
Test Objective	The Ethernet service information is requested from the single DC under test in TD_SDN_SSP_03. The received information is checked to correctly not list the newly deleted service anymore.		
Configuration	CFG_03		
References			
Applicability	MW_8040, MW_8345, MW_TETOPO, MW_MWTOPO, MW_ETHSVC		
Pre-test conditions	• Postman has been correctly initialized earlier, by executing TD_SDN_INIT • The Domain Controller instance is up and running normally • All the devices are upgraded to correct versions • All basic configurations are completed (e.g., NE_id, OSPF, PCEP, etc.) • The Restconf server is serviceable.		
Test Sequence	**Step**	**Type**	**Description**
	1	***Request***	*Send GET request via Postman to one individual domain controller by executing Collection TD_ SDN_SSP_04 (see* Error! eference source not found.*)*
	2	***Validation***	*The response body should no longer contain information about the L2 service deleted in TD_ SDN_SSP_03.*

<table>
<tr><th colspan="4">Interoperability Test Description</th></tr>
<tr><td><b>Identifier</b></td><td colspan="3">TD_ SDN_ISP_01</td></tr>
<tr><td><b>Test Objective</b></td><td colspan="3">Create the specified L2 data service over all available domains. The traffic generator/analyzer confirms that data start flowing.</td></tr>
<tr><td><b>Configuration</b></td><td colspan="3">CFG_04</td></tr>
<tr><td><b>References</b></td><td colspan="3"></td></tr>
<tr><td><b>Applicability</b></td><td colspan="3">MW_8040, MW_8345, MW_TETOPO, MW_MWTOPO, MW_ETHSVC</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td><b>Pre-test conditions</b></td><td colspan="3">• Postman has been correctly initialized earlier, by executing TD_SDN_INIT
• The Domain Controller instance is up and running normally
• All the devices are upgraded to correct versions
• All basic configurations are completed (e.g., NE_id, OSPF, PCEP, etc.)
• All Restconf servers are serviceable.
• If TD_SDN_SSP_01 has been run before TD_ SDN_ISP_01, all Domain Controllers and microwave units should be reset to the state they were before executing TD_SDN_SSP_01</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td rowspan="3"><b>Test Sequence</b></td><td><b>Step</b></td><td><b>Type</b></td><td><b>Description</b></td></tr>
<tr><td>1</td><td><b><i>Request</i></b></td><td><i>Send POST request via Postman to all domain controllers by executing Collection TD_ SDN_ISP_01 (see Error! Reference ource not found.)</i></td></tr>
<tr><td>2</td><td><b><i>Validation</i></b></td><td><i>Check the traffic generator/analyzer if the data start flowing properly.</i></td></tr>
</table>

<table>
<tr><th colspan="4">Interoperability Test Description</th></tr>
<tr><td><b>Identifier</b></td><td colspan="3">TD_ SDN_ISP_02</td></tr>
<tr><td><b>Test Objective</b></td><td colspan="3">The Ethernet service information is requested from all the DCs under test in TD_SDN_ISP_01. The received information is checked to correctly list the newly created service.</td></tr>
<tr><td><b>Configuration</b></td><td colspan="3">CFG_04</td></tr>
<tr><td><b>References</b></td><td colspan="3"></td></tr>
<tr><td><b>Applicability</b></td><td colspan="3">MW_8040, MW_8345, MW_TETOPO, MW_MWTOPO, MW_ETHSVC</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td><b>Pre-test conditions</b></td><td colspan="3">• Postman has been correctly initialized earlier, by executing TD_SDN_INIT
• All the Domain Controller instances are up and running normally
• All the devices are upgraded to correct versions
• All basic configurations are completed (e.g., NE_id, OSPF, PCEP, etc.)
• All Restconf servers are serviceable.
• All L2 data services are successfully created.</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td rowspan="3"><b>Test Sequence</b></td><td><b>Step</b></td><td><b>Type</b></td><td><b>Description</b></td></tr>
<tr><td>1</td><td><b><i>Request</i></b></td><td><i>Send GET request via Postman to all domain controllers by executing Collection TD_ SDN_ ISP_02 (see Error! Reference ource not found.)</i></td></tr>
<tr><td>2</td><td><b><i>Validation</i></b></td><td><i>The response body of each request should contain information about the created L2 service in JSON format referring to Annex D.</i></td></tr>
</table>

<table>
<tr><th colspan="4">Interoperability Test Description</th></tr>
<tr><td><b>Identifier</b></td><td colspan="3">TD_ SDN_ISP_03</td></tr>
<tr><td><b>Test Objective</b></td><td colspan="3">Delete the specified L2 data service over all available domains. The traffic generator/analyzer confirms that data stops flowing.</td></tr>
<tr><td><b>Configuration</b></td><td colspan="3">CFG_04</td></tr>
<tr><td><b>References</b></td><td colspan="3"></td></tr>
<tr><td><b>Applicability</b></td><td colspan="3">MW_8040, MW_8345, MW_TETOPO, MW_MWTOPO, MW_ETHSVC</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td><b>Pre-test conditions</b></td><td colspan="3">• Postman has been correctly initialized earlier, by executing TD_SDN_INIT
• The Domain Controller instance is up and running normally
• All the devices are upgraded to correct versions
• All basic configurations are completed (e.g., NE_id, OSPF, PCEP, etc.)
• All Restconf servers are serviceable.</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td rowspan="3"><b>Test Sequence</b></td><td><b>Step</b></td><td><b>Type</b></td><td><b>Description</b></td></tr>
<tr><td>1</td><td><b><i>Request</i></b></td><td><i>Send DELETE request via Postman to all domain controllers by executing Collection TD_ SDN_ ISP_03 (see Error! eference source not found.)</i></td></tr>
<tr><td>2</td><td><b><i>Validation</i></b></td><td><i>Check the traffic generator/analyzer if the data stop flowing.</i></td></tr>
</table>

<table>
<tr><th colspan="4">Interoperability Test Description</th></tr>
<tr><td><b>Identifier</b></td><td colspan="3">TD_ SDN_ISP_04</td></tr>
<tr><td><b>Test Objective</b></td><td colspan="3">The Ethernet service information is requested from all the DCs under test in TD_SDN_ISP_03. The received information is checked to correctly not list the newly deleted service anymore.</td></tr>
<tr><td><b>Configuration</b></td><td colspan="3">CFG_04</td></tr>
<tr><td><b>References</b></td><td colspan="3"></td></tr>
<tr><td><b>Applicability</b></td><td colspan="3">MW_8040, MW_8345, MW_TETOPO, MW_MWTOPO, MW_ETHSVC</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td><b>Pre-test conditions</b></td><td colspan="3">• Postman has been correctly initialized earlier, by executing TD_SDN_INIT
• All the Domain Controller instances are up and running normally
• All the devices are upgraded to correct versions
• All basic configurations are completed (e.g., NE_id, OSPF, PCEP, etc.)
• All Restconf servers are serviceable.
• All L2 data services are successfully created.</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td rowspan="3"><b>Test Sequence</b></td><td><b>Step</b></td><td><b>Type</b></td><td><b>Description</b></td></tr>
<tr><td>1</td><td><b><i>Request</i></b></td><td><i>Send GET request via Postman to all domain controllers by executing Collection TD_ SDN_ ISP_04 (see Error! Reference ource not found.)</i></td></tr>
<tr><td>2</td><td><b><i>Validation</i></b></td><td><i>The response body should no longer contain information about the L2 service deleted in TD_ SDN_ISP_03.</i></td></tr>
</table>

8.3 Initialization Test Descriptions

Interoperability Test Description			
Identifier	TD_SDN_INIT		
Test Objective	Create and initialize the TD_SDN_SETTINGS object in the runtime environment of Postman. This needs to be done only once at the beginning of a test session with Postman. In case implementation-specific parameters contained in the TD_SDN_INIT source code have been changed, it should be run again, followed by TD_SDN_SNSD_02 and TD_SDN_SNSD_03 (or TD_SDN_MNSD_02 and TD_SDN_MNSD_03, depending on the case).		
Configuration	Not relevant		
References			
Applicability	Not relevant		
Pre-test conditions	• Postman is running		
Test Sequence	**Step**	**Type**	**Description**
	1	***Request***	*Launch the TD_SDN_INIT script from Postman's GUI*

Annex A Interoperability Feature Statement

A.1 Entities

Table 4: Entities

Item	Which entity do you support?	Status	Support
1	MW SDN Domain Controller	Available	Mandatory

A.2 MW Domain Controller Features

Table 5: MW Domain Controller Features

Item	Feature	ID	Ref	Status	Support
1	DC supports Restconf (RFC 8040) on the NBI	MW_8040	[6]	Available	Mandatory
2	DC supports the I2RS DM (RFC 8345) on the NBI as per Annex C	MW_8345	[1]	Available	Mandatory
3	DC supports the TE Topology DM (draft-ietf-teas-yang-te-topo) on the as per Annex C	MW_TETOPO	[2]	Available	Mandatory
4	DC supports the MW Topology DM (draft-ye-ccamp-mw-topo-yang) on the as per Annex C	MW_MWTOPO	[3]	Available	Mandatory
5	DC supports the Ethernet Topology DM (draft-zheng-ccamp-client-topo-yang) on the NBI as per Annex C	MW_ETHTOPO	[4]	Available	Mandatory
6	DC supports the Ethernet Service DM (draft-zheng-ccamp- client-signal-yang) on the NBI as per Annex C	MW_ETHSVC	[5]	Available	Mandatory

Note: for the precise reference to the model drafts to be used, please refer to Annex C.1.

Annex B Domain-Specific Information

In this Annex each Vendor can list any specific implementation-dependent details, which may be necessary to correctly implement the test procedures.

*It is not intended to be a list of allowed non-compliances, full compliance to the feature list described in "*A.2 MW Domain Controller Features*" is mandatory.*

B.1 Ceragon Networks

	Where	Type definition	Ceragon
LTP name	ietf-te-topology:te/ietf-te-topology:name	string	String describing interface type, slot number, port number (eg: "Radio: Slot 2, Port 2")
Tp-id	ietf-network-topology:tp-id	URI	String value of te-tp-id
Te-tp-id	ietf-te-topology:te-tp-id	uint32 or inet:ip-address	Uint32, internal identifier of interface (eg: 268451970)
Service name	etht-svc-name	string	String, user defined
access-ltp-id	access-ltp-id	uint32 or inet:ip-address	Uint32, same as "te-tp-id" above

This domain requires manual deletion of the Bandwidth Profile between TD_SDN_SSP_01 and TD_SDN_ISP_01.

B.2 Ericsson

	Where	Type definition	Ericsson
LTP name	ietf-te-topology:te/ietf-te-topology:name	string	String, user defined
Tp-id	ietf-network-topology:tp-id	URI	"lan:1/5/1"
Te-tp-id	ietf-te-topology:te-tp-id	uint32 or inet:ip-address	Uint32, sequential numbering starting from 1
Service name	etht-svc-name	string	String, user defined
access-ltp-id	access-ltp-id	uint32 or inet:ip-address	Integer, same as "te-tp-id" above

B.3 Huawei Technologies

	Where	Type definition	Huawei
LTP name	ietf-te-topology:te/ietf-te-topology:name	string	“IP addr-id” “100.10.1.23-1”
Tp-id	ietf-network-topology:tp-id	URI	Numerical String
Te-tp-id	ietf-te-topology:te-tp-id	uint32 or inet:ip-address	Uint32
Service name	etht-svc-name	string	string
access-ltp-id	access-ltp-id	uint32 or inet:ip-address	Uint32
topology-id	te-types:te-topology-id	string	“44” is used for microwave topology “45” is used for Ethernet topology

B.4 Intracom Telecom

	Where	Type definition	Intracom
LTP name	ietf-te-topology:te/ietf-te-topology:name	string	String describing the Port Type plus an optional suffix containing a space and the Port Number
Tp-id	ietf-network-topology:tp-id	URI	String describing the Port Type plus an optional suffix containing an underscore and the Port Number
Te-tp-id	ietf-te-topology:te-tp-id	uint32 or inet:ip-address	An integer revealing the Port Number
Service name	etht-svc-name	string	String of practically unlimited size
access-ltp-id	access-ltp-id	uint32 or inet:ip-address	The same integer as te-tp-id

B.5 NEC

	Where	Type definition	NEC
LTP name	ietf-te-topology:te/ietf-te-topology:name	string	String describing card type / port number (e.g. “GbE-A / 9 / 3“)

	Where	Type definition	NEC
Tp-id	ietf-network-topology:tp-id	URI	String describing card type / port number (e.g. “GbE-A / 9 / 3“)
Te-tp-id	ietf-te-topology:te-tp-id	uint32 or inet:ip-address	Uint32, sequential numbering starting from 1
Service name	etht-svc-name	string	String, user defined (Maximum: 32)
access-ltp-id	access-ltp-id	uint32 or inet:ip-address	Integer, same as “te-tp-id” above

B.6 Nokia

	Where	Type definition	Nokia
LTP name	ietf-te-topology:te/ietf-te-topology:name	string	“SlotId-PortId”
Tp-id	ietf-network-topology:tp-id	URI	Numerical String of the Tp Id
Te-tp-id	ietf-te-topology:te-tp-id	uint32 or inet:ip-address	Integer
Service name	etht-svc-name	string	String (Maximum Length allowed is 128 character)
access-ltp-id	access-ltp-id	uint32 or inet:ip-address	Integer

B.7 SIAE Microelettronica

	Where	Type definition	SIAE
LTP name	ietf-te-topology:te/ietf-te-topology:name	string	String describing LTP (i.e. Lan1, Lan2, ..., Radio-1,...)
Tp-id	ietf-network-topology:tp-id	URI	Numerical String of the Tp Id
Te-tp-id	ietf-te-topology:te-tp-id	uint32 or inet:ip-address	Integer
Service name	etht-svc-name	string	string (max.100 character)

	Where	Type definition	SIAE
access-ltp-id	access-ltp-id	uint32 or inet:ip-address	Integer

Annex C IETF Data Model Selection

The subset of IETF Data models, and the subset of parameters thereof, to be used in this Plugtest are specified here.

For convenience, the relevant information is listed here too.

C.1 IETF Data Models Version

The IETF data models are defined in different IETF documents (Internet-Drafts or RFCs) which contain one or more YANG modules each as listed below:

- ietf-microwave-topology@2018-10-22.yang (draft-ye-ccamp-mw-topo-yang-02)
- ietf-eth-te-topology@2018-03-01.yang (draft-zheng-ccamp-client-topo-yang-03)
- ietf-eth-tran-service@2018-08-30.yang (draft-zheng-ccamp-client-signal-yang-02)
- ietf-network@2018-02-26.yang (RFC8345)
- ietf-network-topology@2018-02-26.yang (RFC8345)
- ietf-te-topology@2018-06-15.yang (draft-ietf-teas-yang-te-topo-18) – RFC Queue
- ietf-te-types@2018-06-12.yang (draft-ietf-teas-yang-te-15)
- ietf-eth-tran-types@2018-08-30.yang (draft-zheng-ccamp-client-signal-yang-02)
- ietf-yang-types@2013-07-15.yang (RFC6991)
- ietf-inet-types@2013-07-15.yang (RFC6991)
- ietf-routing-types@2017-12-04.yang (RFC8294)
- ietf-yang-schema-mount@2018-10-16.yang (draft-ietf-netmod-schema-mount-12) – RFC Queue (See Note below)

A reference copy of these files can be found here.

NOTE – The ietf-yang-schema-mount module is needed just to compile the ietf-microwave-topology but it is not to be implemented for the Plugtest (so any version can be used as long as it compiles). The latest version is reported here.

Different YANG modules have different levels of maturity in the standardization process:

- YANG modules are officially released once published in an RFC
- YANG modules defined by Internet-Drafts in RFC Editors' Queue are stable
- YANG modules defined by WG Internet-Drafts (with draft names starting with "draft-ietf") are quite stable
- YANG modules defined by individual Internet-Drafts (with draft names not starting with "draft-ietf") are just individual proposals to IETF and subject to changes during IETF development process

C.2 Tree Diagrams

C.2.1 Microwave Topology Sub-tree

```
+--rw ietf-network:networks
  +--rw ietf-network:network* [network-id]
    +--rw ietf-network:network-id         network-id
    +--rw ietf-network:network-types
    |  +--rw ietf-te-topology:te-topology!
    |  |  +--rw ietf-microwave-topology:mw-topology!
    +--rw ietf-te-topology:provider-id?      te-types:te-global-id
    +--rw ietf-te-topology:client-id?        te-types:te-global-id
    +--rw ietf-te-topology:te-topology-id?   te-types:te-topology-id
    +--rw ietf-te-topology:te!
    | +--rw ietf-te-topology:name?                   string
    +--rw ietf-network:node* [node-id]
    | +--rw ietf-network:node-id           node-id
    | +--rw ietf-te-topology:te-node-id?   te-types:te-node-id
    | +--rw ietf-te-topology:te!
    | | +--rw ietf-te-topology:te-node-attributes
    | | | +--rw ietf-te-topology:name?                  string
    | | +--ro ietf-te-topology:oper-status?               te-types:te-oper-status
    | +--rw ietf-network-topology:termination-point* [tp-id]
    |    +--rw ietf-network-topology:tp-id                         tp-id
    |    +--rw ietf-te-topology:te-tp-id?   te-types:te-tp-id
    |    +--rw ietf-te-topology:te!
    |       +--rw ietf-te-topology:admin-status?
    |       |       te-types:te-admin- status
    |       +--rw ietf-te-topology:name?                          string
    |       +--ro ietf-te-topology:oper-status?
    |       |       te-types:te-oper-status
    +--rw ietf-network-topology:link* [link-id]
      +--rw ietf-network-topology:link-id           link-id
      +--rw ietf-network-topology:source
      | +--rw ietf-network-topology:source-node?   -> ../../../nw:node/node-id
      | +--rw ietf-network-topology:source-tp?     leafref
      +--rw ietf-network-topology:destination
      | +--rw ietf-network-topology:dest-node?   -> ../../../nw:node/node-id
      | +--rw ietf-network-topology:dest-tp?     leafref
      +--rw ietf-te-topology:te-link-attributes
      | +--rw ietf-te-topology:admin-status?
      | |       te-types:te-admin-status
      | +--rw ietf-microwave-topology:mw-link-frequency?          uint32
      | +--rw ietf-microwave-topology:mw-link-channel-separation? uint32
      | +--ro ietf-microwave-topology:mw-link-nominal-bandwidth?  rt-types:bandwidth-ieee-float32
      | +--ro ietf-microwave-topology:mw-link-current-bandwidth?  rt-types:bandwidth-ieee-float32
      | +--rw ietf-microwave-topology:mw-unreserved-bandwidth    rt-types:bandwidth-ieee-float32
      | +--ro ietf-microwave-topology:mw-link-availability*  [availability]
      |    +--ro ietf-microwave-topology:mw-link-availability     rt-types:percentage
      |    +--ro ietf-microwave-topology:mw-link-bandwidth        rt-types:bandwidth-ieee-float32
      +--ro ietf-te-topology:oper-status?                te-types:te-oper-status
```

C.2.2 Ethernet Topology Sub-tree

```
+--rw ietf-network:networks
  +--rw ietf-network:network* [network-id]
    +--rw ietf-network:network-id           network-id
    +--rw ietf-network:network-types
    |  +--rw ietf-te-topology:te-topology!
    |  |  +--rw ietf-eth-te-topology:eth-tran-topology!
    +--rw ietf-te-topology:provider-id?      te-types:te-global-id
    +--rw ietf-te-topology:client-id?        te-types:te-global-id
    +--rw ietf-te-topology:te-topology-id?   te-types:te-topology-id
    +--rw ietf-te-topology:te!
    | +--rw ietf-te-topology:name?                   string
    +--rw ietf-network:node* [node-id]
    | +--rw ietf-network:node-id           node-id
    | +--rw ietf-te-topology:te-node-id?   te-types:te-node-id
    | +--rw ietf-te-topology:te!
  | | +--rw ietf-te-topology:te-node-attributes
 | | | +--rw ietf-te-topology:name?                 string
  | | +--ro ietf-te-topology:oper-status?              te-types:te-oper-status
  | +--rw ietf-network-topology:termination-point* [tp-id]
  |    +--rw ietf-network-topology:tp-id                       tp-id
  |    +--rw ietf-te-topology:te-tp-id?   te-types:te-tp-id
  |    +--rw ietf-te-topology:te!
  |    | +--rw ietf-te-topology:admin-status?
  |    | |      te-types:te-admin- status
  |    | +--rw ietf-te-topology:name?                          string
  |    | +--ro ietf-te-topology:oper-status?
  |    | |      te-types:te-oper-status
  |    +--rw ietf-eth-te-topology:svc!
  |       +--rw ietf-eth-te-topology:client-facing?              boolean
  +--rw ietf-network-topology:link* [link-id]
  | +--rw ietf-network-topology:link-id           link-id
  | +--rw ietf-network-topology:source
  | | +--rw ietf-network-topology:source-node?   -> ../../../nw:node/node-id
  | | +--rw ietf-network-topology:source-tp?     leafref
  | +--rw ietf-network-topology:destination
  | | +--rw ietf-network-topology:dest-node?   -> ../../../nw:node/node-id
  | | +--rw ietf-network-topology:dest-tp?     leafref
  | +--rw ietf-te-topology:te-link-attributes
  | | +--rw ietf-te-topology:underlay {te-topology-hierarchy}?
  | | | +--rw ietf-te-topology:enabled?                  boolean
  | | | +--rw ietf-te-topology:primary-path
  | | | | +--rw ietf-te-topology:network-ref?    ->
/nw:networks/network/network-id
  | | | | +--rw ietf-te-topology:path-element* [path-element-id]
  | | | |    +--rw ietf-te-topology:path-element-id    uint32

  | | | |    +--rw ietf-te-topology:index?            uint32
  | | | |    +--rw (ietf-te-topology:type)?
  | | | |       +--:(ietf-te-topology:num-unnum-hop )
  | | | |       | +--rw ietf-te-topology:num-unnum-hop
  | | | |       |    +--rw ietf-te-topology:node-id?      te-types:te-node-id
  | | | |       |    +--rw ietf-te-topology:link-tp-id?   te-types:te-tp-id
  | | | |       |    +--rw ietf-te-topology:hop-type?     te-hop-type
  | | +--rw ietf-te-topology:admin-status?
  | | |      te-types:te-admin-status
  | | +--rw ietf-eth-te-topology:max-bandwidth?         uint64
  | | +--rw ietf-eth-te-topology:available-bandwidth?    uint64
  | +--ro ietf-te-topology:oper-status?              te-types:te-oper-status
```

C.2.3 Ethernet Service Sub-tree

```
module: ietf-eth-tran-service
   +--rw etht-svc
      +--rw globals
      | +--rw etht-svc-bandwidth-profiles* [bandwidth-profile-name]
      |    +--rw bandwidth-profile-name    string
      |    +--rw bandwidth-profile-type?   etht-types:bandwidth-profile-type
      |    +--rw CIR?                      uint64
      |    +--rw EIR?                      uint64
      |    +--rw color-aware?              boolean
      |    +--rw coupling-flag?            boolean
      +--rw etht-svc-instances* [etht-svc-name]
         +--rw etht-svc-name            string
         +--rw etht-svc-type?           etht-types:service-type
         +--rw access-provider-id?      te-types:te-global-id
         +--rw access-client-id?        te-types:te-global-id
         +--rw access-topology-id?      te-types:te-topology-id
```

```
+--rw etht-svc-access-ports* [access-port-id]
|  +--rw access-port-id                   uint16
|  +--rw access-node-id?                  te-types:te-node-id
|  +--rw access-ltp-id?                   te-types:te-tp-id
|  +--rw service-classification-type?     identityref
|  +--rw (service-classification)?
|  |  +--:(vlan-classification)
|  |     +--rw outer-tag!
|  |     |  +--rw tag-type?     etht-types:eth-tag-classify
|  |     |  +--rw (individual-bundling-vlan)?
|  |     |     +--:(individual-vlan)
|  |     |     |  +--rw vlan-value?   etht-types:vlanid
|  +--rw (direction)?
|  |  +--:(symmetrical)
|  |  |  +--rw ingress-egress-bandwidth-profile-name?   string
+--rw admin-status?           identityref
+--ro state
   +--ro operational-state?        identityref
    +--ro provisioning-state?        Identityref
```

Annex D JSON Code

D.1 Microwave Topology

Please refer to this file.

D.2 Ethernet Topology

Please refer to this file.

D.3 Ethernet Service

Please refer to this file.

Annex E Postman

E.1 Postman Collection Structure

Postman can perform an automated test sequence, thanks to its capability to execute scripts.

Postman allows to group requests into Collections, so that it is possible to execute multiple requests with one command. It also allows to add a **pre-request** script (optional), which is executed before the collection runs, and a **test script** (optional), that is executed after the collection runs.

E.2 Postman Collections

The Collections defined for this Plugtest are available at https://forge.etsi.org/gitlab/sdn/mwt/mwt-plu-postman-collections/tree/intracom-postman/collections.

An initialization script to be run once at the beginning of a test session, to prepare the Postman execution environment, is provided at https://forge.etsi.org/gitlab/sdn/mwt/mwt-plu-postman-collections/tree/intracom-postman/collections/mwt-plu-1/TD_SDN_INIT.

A tutorial about how to use the Collections is available at https://forge.etsi.org/gitlab/sdn/mwt/mwt-plu-postman-collections/blob/intracom-postman/collections/mwt-plu-1/Guide.pdf.