Technical Report

TR-435

NETCONF Requirements for Access Nodes and Broadband Access Abstraction

Issue 1

Issue Date: December 2020

Notice

The Broadband Forum is a non-profit corporation organized to create guidelines for broadband network system development and deployment. This Technical Report is a draft, is subject to change, and has not been approved by members of the Forum. This Technical Report is owned and copyrighted by the Broadband Forum, and portions of this Technical Report may be owned and/or copyrighted by Broadband Forum members. This Technical Report is only available to Broadband Forum Members and Observers.

Intellectual Property

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of this Technical Report if it were to be adopted as a Technical Report, and to provide supporting documentation.

Terms of Use

1. License

Broadband Forum hereby grants you the right, without charge, on a perpetual, non-exclusive and worldwide basis, to utilize the Technical Report for the purpose of developing, making, having made, using, marketing, importing, offering to sell or license, and selling or licensing, and to otherwise distribute, products complying with the Technical Report, in all cases subject to the conditions set forth in this notice and any relevant patent and other intellectual property rights of third parties (which may include members of Broadband Forum). This license grant does not include the right to sublicense, modify or create derivative works based upon the Technical Report except to the extent this Technical Report includes text implementable in computer code, in which case your right under this License to create and modify derivative works is limited to modifying and creating derivative works of such code. For the avoidance of doubt, except as qualified by the preceding sentence, products implementing this Technical Report are not deemed to be derivative works of the Technical Report.

2. NO WARRANTIES

THIS Technical Report IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NONINFRINGEMENT AND ANY IMPLIED WARRANTIES ARE EXPRESSLY DISCLAIMED. ANY USE OF THIS Technical Report SHALL BE MADE ENTIRELY AT THE USER'S OR IMPLEMENTER'S OWN RISK, AND NEITHER THE BROADBAND FORUM, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY USER, IMPLEMENTER, OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER, DIRECTLY OR INDIRECTLY, ARISING FROM THE USE OF THIS Technical Report, INCLUDING BUT NOT LIMITED TO, ANY CONSEQUENTIAL, SPECIAL, PUNITIVE, INCIDENTAL, AND INDIRECT DAMAGES.

3. THIRD PARTY RIGHTS

Without limiting the generality of Section 2 above, BROADBAND FORUM ASSUMES NO RESPONSIBILITY TO COMPILE, CONFIRM, UPDATE OR MAKE PUBLIC ANY THIRD PARTY ASSERTIONS OF PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS THAT MIGHT NOW OR IN THE FUTURE BE INFRINGED BY AN IMPLEMENTATION OF THE Technical Report IN ITS CURRENT, OR IN ANY FUTURE FORM. IF ANY SUCH RIGHTS ARE DESCRIBED ON THE Technical Report, BROADBAND FORUM TAKES NO POSITION AS TO THE VALIDITY OR INVALIDITY OF SUCH ASSERTIONS, OR THAT ALL SUCH ASSERTIONS THAT HAVE OR MAY BE MADE ARE SO LISTED.

All copies of this Technical Report (or any portion hereof) must include the notices, legends, and other provisions set forth on this page.

Issue History

Issue Number	Approval Date	Publication Date	Issue Editor	Changes
1	17 December 2020	17 December 2020	Ken Kerpez, ASSIA Mauro Tilocca, TIM	Original

Comments or questions about this Broadband Forum Technical Report should be directed to info@broadband-forum.org.

Editors:	Ken Kerpez, ASSIA Mauro Tilocca, TIM
Work Area Director(s):	George Dobrowski Bruno Cornaglia, Vodafone
Project Stream Leader(s):	Yves Hertoghs, VMware Ning Zong, Huawei

Table of Contents

Table of Figures

Table of Tables

Executive Summary

A goal of Software-Defined Networking is to enable services providers to quickly respond to changing business requirements by defining a centralized architecture where the device management interfaces and data models are well defined and augmentable. It is equally important to standardize the interfaces used to manage these devices, defining the requirements necessary to ensure interoperability within a Software Defined Access Network (SD-AN).

New service requirements can then be introduced in a faster/agile and more error-free fashion. Standard Development Organizations like IETF, BBF, IEEE, and MEF have developed and validated YANG Data Models focused on defining the device and service data models to be applied in the northbound interface of devices.

This study examines NETCONF features and functions applicable to Access Nodes, the Broadband Access Abstraction layer and SDN Management and Control systems.

1 Purpose and Scope

1.1 Purpose

The purpose of this Technical Report is to define the NETCONF requirements applicable to the NETCONF/YANG Northbound interfaces (NBIs) of Access Nodes (AN), Broadband Access Abstraction (BAA) layer implementations and the Southbound interfaces (SBIs) of SDN Management and Control systems using standard YANG models defined by BBF (e.g., in TR-385 [6], TR-383 [4], TR-355 [2], WT-454 [41]) and relevant models defined by other bodies (e.g., IETF, IEEE).

This project relies on existing BBF access network related specifications that use or provide normative language for NETCONF usage, namely TR-413 [7], TR-301 [1] and WT-411 [40] with the aim to provide a unified and harmonized specification for implementers of NETCONF based solutions and Network Operators in their process of migration towards an SDN-based and automated management architecture.

Interoperability between ANs and SDN Management and Control systems, potentially mediated by a BAA layer, is critical to the success of managing and deploying access networks with new SDN/NFV paradigms, i.e., what is called Software Defined Access Network (SD-AN).

The adoption of NETCONF/YANG interfaces enables a number of key characteristics of the SD-AN.

More specifically the motivation for defining these NETCONF requirements is to address operators' requirements in order to:

- Foster a quick and effective ecosystem adoption of innovative NETCONF/YANG interfaces on managed network resources to enable automation,
- Improve agility (i.e., Time-to-Market, ease of service upgrade and scaling) for delivering services by supporting a well-defined set of operator use cases (and what NETCONF features/functions are needed for those use cases),
- Improve service manageability by supporting common management functions, where it is easier to troubleshoot and fix remotely,
- Reduce operational and capital expenses by defining common management models and use cases as well as automatable and less error-prone processes.

1.2 Scope

The scope of TR-435 is to define a common set of NETCONF Server and NETCONF Client requirements applicable, depending on their respective role, to ANs, as much as possible regardless of their type or technology, to BAA layers (when deployed as mediation elements between ANs and northbound elements) and to SDN Management and Control systems.

In this, the term AN is constrained to include Passive Optical Network (PON) Network Elements (NE), Distribution Point Units (DPU), Digital Subscriber Line Access Multiplexers (DSLAM), and Multi-Services Access Networks (MSAN).

This Technical Report is intended to be a list of requirements needed for NETCONF support of the different management use cases.

This specification is expected to be forward looking and possibly leave behind the tailoring of management features of ANs based on its technology, type, location and so on.

This is backed by the following:

- With SDN, ANs will be managed via a virtualized Management Entity (within the BAA or an SDN Management and Control) so there are less implementation constraints in the related feature set,

- For vendors, in an SDN-based production environment unifying the way ANs are managed allows to reuse the expertise, software and validation procedures,
- For Operators, the above allows unifying pre-deployment activities, network operation procedures, service creation, and service assurance workflows across different AN types and FTTx scenarios.

2 References and Terminology

2.1 Conventions

In this Technical Report, several words are used to signify the requirements of the specification. These words are always capitalized. More information can be found be in RFC 2119 [11].

MUST This word, or the term “REQUIRED”, means that the definition is an absolute requirement of the specification.

MUST NOT This phrase means that the definition is an absolute prohibition of the specification.

SHOULD This word, or the term “RECOMMENDED”, means that there could exist valid reasons in particular circumstances to ignore this item, but the full implications need to be understood and carefully weighed before choosing a different course.

SHOULD NOT This phrase, or the phrase "NOT RECOMMENDED" means that there could exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications need to be understood and the case carefully weighed before implementing any behavior described with this label.

MAY This word, or the term “OPTIONAL”, means that this item is one of an allowed set of alternatives. An implementation that does not include this option MUST be prepared to inter-operate with another implementation that does include the option.

2.2 References

2.2.1 Published References

The following references are of relevance to this Technical Report. At the time of publication, the editions indicated were valid. All references are subject to revision.

Users of this Technical Report are therefore strongly encouraged to investigate the possibility of applying the most recent edition of the listed references by checking at the web links reported below, especially for YANG DMs.

A list of currently valid BBF Technical Reports is published at www.broadband-forum.org/technical-reports. Users also may find useful BBF’s searchable Resources database for Technical Reports and other BBF deliverables.

The most recent edition of BBF’s published YANG Data Models can be found at YANG Projects.

Document	Title	Source	Year
[1] TR-301 Issue 2 Corrigendu	Architecture and Requirements for Fiber to the Distribution Point	BBF	2019

Document	Title	Source	Year
m 1			
[2] TR-355 Issue 1 Amendment 3	YANG Modules for FTTdp Management	BBF	2020
[3] TR-370 Issue 2	Fixed Access Network Sharing – Architecture and Nodal Requirements	BBF	2020
[4] TR-383 Issue 1 Amendment 3	Common YANG Modules for Access Networks	BBF	2020
[5] TR-384	Cloud Central Office (CloudCO) Reference Architectural Framework	BBF	2018
[6] TR-385 Issue 2	ITU-T PON YANG Modules	BBF	2020
[7] TR-413	SDN Management and Control Interfaces for CloudCO Network Functions	BBF	2018
[8] RFC 822	Standard For The Format Of ARPA Internet Text Messages	IETF	1983
[9] RFC 1035	Domain Names - Implementation And Specification	IETF	1987
[10] RFC 1122	Requirements for Internet Hosts -- Communication Layers	IETF	1989
[11] RFC 2119	Key words for use in RFCs to Indicate Requirement Levels	IETF	1997
[12] RFC 2132	DHCP Options and BOOTP Vendor Extensions	IETF	1997
[13] RFC 3396	Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)	IETF	2002
[14] RFC 3925	Vendor-Identifying Vendor Options for Dynamic Host Configuration Protocol version 4 (DHCPv4)	IETF	2004
[15] RFC 4361	Node-specific Client Identifiers for Dynamic Host Configuration Protocol Version Four (DHCPv4)	IETF	2006
[16] RFC 4741	NETCONF Configuration Protocol	IETF	2006
[17] RFC 5246	The Transport Layer Security (TLS) Protocol Version 1.2	IETF	2008
[18] RFC 5277	NETCONF Event Notifications	IETF	2008
[19] RFC 5280	Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile	IETF	2008
[20] RFC 6022	YANG Module for NETCONF Monitoring	IETF	2010
[21] RFC 6187	X.509v3 Certificates for Secure Shell Authentication	IETF	2011
[22] RFC 6241	Network Configuration Protocol (NETCONF)	IETF	2011
[23] RFC 6242	Using the NETCONF Protocol over Secure Shell (SSH)	IETF	2011
[24] RFC 6536	Network Configuration Protocol (NETCONF) Access Control Model	IETF	2012

Document	Title	Source	Year
[25] RFC 7589	Using the NETCONF Protocol over Transport Layer Security (TLS) with Mutual X.509 Authentication	IETF	2015
[26] RFC 7895	YANG Module Library	IETF	2016
[27] RFC 8071	NETCONF Call Home and RESTCONF Call Home	IETF	2017
[28] RFC 8341	Network Configuration Access Control Model	IETF	2018
[29] RFC 8415	Dynamic Host Configuration Protocol for IPv6 (DHCPv6)	IETF	2018
[30] RFC 8525	YANG Library	IETF	2019
[31] G.984.x	Gigabit-capable passive optical networks (GPON)	ITU-T	
	G.984.1 – General characteristics		2008
	G.984.2 – Physical Media Dependent (PMD) layer specification		2019
	G.984.3 – Transmission convergence layer specification		2014
	G.984.4 – ONT management and control interface specification		2008
	G.984.5 – Enhancement band		2014
	G.984.6 – Reach extension		2008
	G.984.7 – Long reach		2010
[32] G.987.x	10-Gigabit-capable passive optical networks (X-GPON)	ITU-T	
	G.987 – Definitions, abbreviations and acronyms		2012
	G.987.1 – General requirements		2016
	G.987.2 – Physical Media Dependent (PMD) layer specification		2016
	G.987.3 – Transmission convergence layer specification		2014
	G.987.4 – Reach extension		2012
[33] G.988	ONU management and control interface (OMCI) specification	ITU-T	2017
[34] G.989.x	40-Gigabit-capable passive optical networks (NG-PON2)	ITU-T	
	G.989 – Definitions, abbreviations and acronyms G.989.1 – General requirements		2015
	G.989.2 – Physical Media Dependent (PMD) layer specification		2013
	G.989.3 – Transmission convergence layer specification		2019
[35] G.9807.x	G.9807.1 – 10-Gigabit-capable symmetric passive optical network (XGS-PON)	ITU-T	2016
	G.9807.2 – Reach extension		2017
[36] G.988	ONU Management and Control Interface Specification (OMCI)	ITU-T	2010

Document	Title	Source	Year
[37] G.989	40-Gigabit-capable passive optical networks (NG-PON2) G.989 - Definitions, abbreviations and acronyms G.989.1 – General requirements G.989.2 – Physical media dependent (PMD) layer specification G.989.3 – Transmission convergence layer specification	ITU-T	2015 2013 2019 2015
[38] G.9807	G.9807.1 – 10 Gigabit-capable passive optical networks (XG(S)-PON) G.9807.2 – Reach extension	ITU-T	2016 2107
[39] X.509	Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks	ITU-T	2019

2.2.2 Draft References

The reference documents listed in this section are applicable to this Technical Report but are currently under development within the respective body and are expected to be released in the future. Users of this Technical Report are advised to consult the source body for current status of the referenced documents or their successors.

Informative Appendix II cross-references draft-ietf-netconf-netconf-client-server [42] to specifiy requirements for the features specified in that IETF draft.

Document	Title	Source	Year
[40] WT-411	Definition of interfaces between Cloud CO Functional Modules	BBF	TBD
[41] WT-454	YANG Modules for Access Network Map & Equipment Inventory	BBF	TBD
[42] draft-ietf-netconf-netconf-client-server-18	NETCONF Client and Server Models	IETF	TBD

2.3 Definitions

The following terminology is used throughout this Technical Report.

NETCONF	The Network Configuration Protocol (NETCONF) is a network management protocol developed and standardized by the IETF.

NETCONF Server/Client Roles — NETCONF uses a simple RPC-based mechanism to facilitate communication between a NETCONF client and a server. The client can be a script or application typically running as part of a network manager. The server is typically a network device. A NETCONF client or server can initiate the NETCONF session (please refer to draft-NETCONF Client and Server Models for descriptions on when either option is available).

virtual AN Management Function — In a Software Defined Access Network (SDAN) architecture, this function represents the virtualization of the Management Entity and related functionalities traditionally implemented within the Access Node.

YANG Data Model — YANG is an abstract data modeling language used to model configuration and state data manipulated by the Network Configuration Protocol (NETCONF).

2.4 Abbreviations

This Technical Report uses the following abbreviations:

AN	Access Node
DS	Data Store
EMS	Element Management System
IP	Internet Protocol
OLT	Optical Line Terminal
ONT	Optical Network Terminal
ONU	Optical Network Unit
PKI/CA	Public Key Infrastructure/Certification Authority
RPC	Remote Procedure Call
SDAN	Software Defined Access Network
SSH	Secure Shell
TLS	Transport Layer Security
TR	Technical Report
vANMF	virtual Access Node Management Function
WA	Work Area
YANG	Yet Another Next Generation

3 Technical Report Impact

3.1 Energy Efficiency

TR-435 has no impact on energy efficiency.

3.2 Security

TR-435 has no impact on security. NETCONF by definition is a secure management protocol. TR-435 does not add additional security requirements beyond those already contained in IETF standards.

3.3 Privacy

TR-435 has no impact on privacy.

4 NETCONF Protocol

4.1 Introduction

RFC 4741 [16] introduced NETCONF protocol for managing, retrieving and manipulating network device configuration data. RFC 4741 [16] has been obsoleted by RFC 6241 [22]. RFC 6241 [22] does advocate that one of the key aspects of NETCONF is its ability to closely mirror functionality native to the device (e.g., command line interface) and thus reducing implementation costs and complexity when new features are added. While it is possible that a vendors NETCONF and command line interfaces are structured similarly, that is viewed as an implementation choice that a vendor may or may have not be taken advantage of.

Table 5 through Table 10 list requirements applicable to the Access Network NETCONF server or client, with each row representing a unique requirement. Each row (requirement) includes the following (listed by table column):

1. The requirement number.
2. The requirement details such as the NETCONF operation covered by the requirement.
3. Section number and Title from source document.
4. Requirement applicability, per section 2.1, to the NETCONF Server. An entry of “N/A” indicates the requirement is not applicable to the NETCONF Server.
5. Requirement applicability, per section 2.1, to the NETCONF Client. An entry of “N/A” indicates the requirement is not applicable to the NETCONF Client.
6. Use case and informative notes.

Certain NETCONF features are applicable only if other main features are supported. In these cases of functional dependencies, the corresponding requirement applicability cell refers to a note that cross-references to the relevant main feature(s).

A NETCONF server could be implemented on an Access Node (AN) (e.g., OLT, DPU, DSLAM, MSAN) and on a BAA layer’s NBI.

A NETCONF client could be implemented on a BAA layer’s SBI, an SDN Manager and Controller or an Element Management System (EMS).

In an SDN-based and automated management architecture, multiple NETCONF Clients can concurrently connect to the same NETCONF Server’s Data Store (DS). This requires that Data Store remains consistent and avoids conflicts when multiple NETCONF Clients read and update it.

For example, the following mechanisms can be employed to ensure consistency and avoid conflicts between NETCONF Clients:

- Using the NETCONF protocol’s NETCONF Server's usage built-in transactional capabilities (i.e., lock/unlock RPCs)
- Moving conflict avoidance and consistency checks to an arbitrating NETCONF Client responsible for directly interacting with the NETCONF Server. This NETCONF Client exposes the network resources to multiple “clients” and applies brokerage and reconciliation to the client’s requests.

These concepts are described further in:

- TR-384 [5], section 6.1.1.1 about multi-tenancy and reconciliation of service requests and section 6.3 about conflict or inconsistency avoidance via an appropriate reconciliation entity
- TR-370 [3], section 5.3 about resource brokerage and reconcile of requests from Tenants.

4.1.1 Use Cases

In the section Use Cases are described via interactions between a NETCONF Client (NC Client) and a NETCONF Server (NC Server).

In the following tables, a NC Client is generically mentioned while a generic AN is referred to as the device hosting the NC Server.

The majority of the Use Cases applies also to the case of a BAA Layer hosting a NC Server on its NBI. Those not applicable to the BAA Layer's NC Server are explicitly indicated.

All the use cases were grouped in the following three functional categories:

- Connectivity
- Network and Service Delivery
- Network and Service Assurance

The purpose of these use cases is two-fold, first to review which of NETCONF standards and capabilities are required for use case implementation, secondly to identify the necessary data models.

The following NC capabilities are concerned:

- urn:ietf:params:xml:ns:netconf:base:1.0
- urn:ietf:params:netconf:base:1.1
- urn:ietf:params:netconf:capability:writable-running:1.0
- urn:ietf:params:netconf:capability:candidate:1.0
- urn:ietf:params:netconf:capability:confirmed-commit:1.1
- urn:ietf:params:netconf:capability:rollback-on-error:1.0
- urn:ietf:params:netconf:capability:validate:1.1
- urn:ietf:params:netconf:capability:startup:1.0
- urn:ietf:params:netconf:capability:xpath:1.0
- urn:ietf:params:netconf:capability:url:1.0
- urn:ietf:params:netconf:capability:partial-lock:1.0
- urn:ietf:params:netconf:capability:notification:1.0
- urn:ietf:params:netconf:capability:interleave:1.0
- urn:ietf:params:netconf:capability:with-defaults:1.0
- urn:ietf:params:netconf:capability:yang-library:1.0
- urn:ietf:params:netconf:capability:yang-library:1.1
- urn:ietf:params:netconf:capability:time:1.0

Regarding YANG data models and features, BBF, IETF and IEEE data models are concerned as specified in TR-413 [7]. In the following it is assumed that the data models support the standard parameters and features.

Table 1 Connectivity Use Cases

	CONNECTIVITY

	CONNECTIVITY
UC Number	**Use Case**
UC-1	NC Client - AN session establishment
UC-2	AN-NC Client session establishment Note: not applicable to BAA Layer’s NC Server
UC-3	NC notifications
UC-4	NC Server Monitoring
UC-5	NC Client management of NC accounts in the AN

Table 2 Network & Service Delivery Use Cases

	NETWORK & SERVICE DELIVERY
UC-6	Configuration error management
UC-7	Multi-domain orchestrated service provision
UC-8	Provisioning synchronization with lock
UC-9	Operation scheduling
UC-10	Defaults discovery and control
UC-11	Profile creation in the AN
UC-12	AN resource discovery and spontaneous reports
UC-13	Resets of AN components and interfaces
UC-14	Administrative state management
UC-15	AN discovery
UC-16	VLAN service provisioning
UC-17	Software management and upgrade

Table 3 Network & Service Assurance Use Cases

	NETWORK & SERVICE ASSURANCE
UC-18	Diagnostics data
UC-19	Network Termination device Dying Gasp
UC-20	Performance Monitoring
UC-21	Connectivity Fault Management
UC-22	IPFIX bulk data collection
UC-23	Alarms and events notification management

4.1.1.1 UC-1 NC Client - AN Session Establishment

This use case concerns NC capabilities related to the NC Client initiating an NC session over SSH-2. The related requirements address NC, SSH session establishment, security settings session monitoring including keep-alive mechanism and capabilities like NETCONF base.

4.1.1.2 UC-2 AN-NC Client Session Establishment

This use case concerns NC capabilities related to the AN initiating an NC session using NETCONF Call Home. The related requirements address NC and TLS session establishment, security settings, DHCP message format and session monitoring including keep-alive mechanism. The possibility to reuse the solution defined in TR-301 to be studied. The requirements should also address provisioning of the initial network connectivity like setting up a management VLAN.

4.1.1.3 UC-3 NC Notifications

This use case concerns NC capabilities related to event and alarm reporting by the AN, including notification and interleave capabilities.

4.1.1.4 UC-4 NC Server Monitoring

This use case concerns NC capabilities (like yang-library) operations (like get-schema) and YANG data models related to NC server monitoring including RFC 6022 [20], RFC 7895 [26] and RFC 8525 [30].

4.1.1.5 UC-5 NC Client Management of NC Accounts in the AN

This use case concerns the configuration of NC user accounts in the AN. The related requirements address support of RFC 8341 [28] and other applicable specifications.

4.1.1.6 UC-6 Configuration Error Management

This use case concerns the capabilities for verification if the configuration to be deployed in the AN is correct and the means to roll back the invalid configuration automatically or upon NC client decision. The related requirements address the capabilities: candidate, confirmed-commit, rollback-on-error and validate.

4.1.1.7 UC-7 Multi-domain Orchestrated Service Provisioning

This use case concerns the situation where the AN is one of the NC-capable devices involved in the end-to-end service chain. The provisioning is done as a transaction spread over the AN and the other involved devices.

To support this use case the Candidate feature of NETCONF protocol is mandatory. This feature consists in the ability to set up a complete network configuration on multiple ANs, to verify it and then apply with a single commit command. In case of failure on a single element, a rollback option is available.

A hierarchical SDN architecture for the access (and any SDN domain), allows to deploy management and control functionalities flexibly across the elements of the hierarchy.

As a good practice for efficient resource storage allocation, the Candidate Datastore associated to a NC Server has to be supported on at least one layer of the SDN hierarchy hosting a NC Server, i.e., on either the AN or the BAA layer or the Access SDN M&C element. The advised SDN layer is where the Operator decides to place the so called “source of truth” for the access network.

The NC Client sequence of operations on the Server is:

- Lock running DS
- Clear candidate DS
- Edit candidate DS
- Validate candidate DS
- Commit
- Confirm Commit
- Unlock running DS

The related requirements address the capabilities: candidate, confirmed-commit, rollback-on-error and validate.

4.1.1.8 UC-8 Provisioning Synchronization with Lock

This use case concerns locking access to the whole data store or part of it to prevent configuration inconsistencies if more than one NC client attempts to change AN configuration at the same time.

The related requirements address the NC operation lock and the partial-lock capabilities.

4.1.1.9 UC-9 Operation Scheduling

This use case concerns operation scheduling in the AN. The related requirements address the time capability.

4.1.1.10 UC-10 Defaults Discovery and Control

This use case concerns discovery of how the default settings are processed by the AN. The related requirements concern the with-defaults capability.

4.1.1.11 UC-11 Profile Creation in the AN

This use case concerns CRUD operations and YANG data models to manage profiles like L2 QoS, ACL, DHCP, PPoE, IGMP, PON QoS, etc.

The related requirements concern the capabilities like writable-running, candidate:1.0, confirmed-commit:1.1, rollback-on-error:1.0, validate, startup, with-defaults.

4.1.1.12 UC-12 AN Resource Discovery and Spontaneous Reports

This use case concerns discovery of AN hardware configuration (e.g., boards, optical, transceivers/links, etc.) and its interfaces as well as spontaneous reporting by AN of hardware or interface changes including their state and usage.

The concerned operations include:

- Listing and getting detailed information on boards installed in the AN
- Listing and getting detailed information on installed PON ports for OLTs
- Listing and getting detailed information on installed Ethernet ports (network and line side)
- Listing and getting detailed information of provisioned/not provisioned ONUs (including the Serial Number)

The related requirements concern the subtree filtering feature, Xpath capability.

4.1.1.13 UC-13 Resets of AN Components and Interfaces

This use case concerns the reset action on the whole AN, its components (e.g., boards, PON ports, network/uplink ports, etc.), ONUs and interfaces.

The related requirements concern the writable-running, candidate, startup, subtree filtering feature, Xpath capability.

Note: this could be used in case of disaster recovery or hot swap replacement.

4.1.1.14 UC-14 Administrative State Management

This use case concerns the change of the administrative state of the AN, its components (e.g., boards, ports), ONUs and related interfaces.

The related requirements concern the subtree filtering feature, Xpath capability.

4.1.1.15 UC-15 ONU Discovery

This use case concerns the scenario when OLT reports to the NC Client the discovery of a new ONUs (either rogue or not).

The related requirements concern the capabilities like notification, interleave.

4.1.1.16 UC-16 VLAN Service Provisioning

This use case concerns the VLAN provisioning, based on AN available resource retrieval, per the following connectivity models:

- 1:N unicast VLAN with PPPoE
- 1:N unicast VLAN with DHCP
- 1:1 unicast VLAN with PPPoE
- 1:1 unicast VLAN with DHCP
- 1:N multicast VLAN with IGMP
- TR-156 - Transparent LAN Service (for business)

The related requirements concern the subtree filtering feature, Xpath capability.

4.1.1.17 UC-17 Software Management and Upgrade

This use case incudes:

- Retrieval of software release information of AN components and Network Termination devices
- Software release upgrades of AN components and, for OLTs, Network Termination devices and rollback upon error

The related requirements concern the subtree filtering feature, Xpath capability.

4.1.1.18 UC-18 Diagnostics Data

This use case concerns retrieval of diagnostics data (e.g., status, operational state, CPU, memory usage, temperature measurements, power levels) on the AN components (e.g., boards, ports, etc.), ONUs and the interfaces.

The related requirements concern the subtree filtering feature, Xpath capability.

4.1.1.19 UC-19 Network Termination Device Dying Gasp

AN reports to the NC Client a Dying Gasp (start and clear events) state of the Network Termination device.

The related requirements concern the capabilities like notification, interleave, feature.

4.1.1.20 UC-20 Performance Monitoring

This use case concerns starting, stopping, resetting and retrieving PM counters and statistics on Ethernet interfaces (e.g., associated to uplink ports, LT ports, NT ports, etc.) and physical layer interfaces on both the NT and LT sides.

The related requirements concern the subtree filtering feature, Xpath capability.

4.1.1.21 UC-21 Connectivity Fault Management

The use case concerns Connectivity Fault Management.

The related requirements concern the notification, subtree filtering feature, Xpath capability.

4.1.1.22 UC-22 IPFIX Bulk Data Collection

This use case concerns configuration of IPFIX export process and bulk statistics collection on AN hardware components, Network Termination devices and interfaces (e.g., traffic passed and rejected).

The related requirements concern the notification, subtree filtering feature, Xpath capability.

4.1.1.23 UC-23 Alarms and Events Notification Management

This use case includes:

- Configuration and assignment to AN and Network Termination device resources of alarm profiles (including assigned severity)

- Configuration and assignment to AN and Network Termination device resources of events notification profiles. In particular those associated to Threshold Crossing (e.g., to anticipate traffic congestion conditions)
- Subscription to alarms and events reported by the AN
- Spontaneous reporting of alarms and events per the assigned profiles
- Retrieval of the list of active alarms or a list of historical alarms

The related requirements concern the subtree filtering feature, Xpath, notification, interleave capabilities.

4.2 Infrastructure Requirements

The following convention applies to Table 4 - NETCONF Infrastructure about fulfilling the requirements indicated in the column “R-x” of each row:

- To fulfill the requirement associated to each row, the NETCONF server MUST support, for the functional area indicated in the column “Area”, the value indicated in column “NETCONF server”. If the corresponding value is “N/A” then that requirement is not applicable.
- To fulfill the requirement associated to each row, the NETCONF client MUST support, for the functional area indicated in the column “Area”, the value indicated in column “NETCONF client”. If the corresponding value is “N/A” then that requirement is not applicable.

Table 4 - NETCONF Infrastructure

R-x	Area	NETCONF		Additional comments
		server	client	
[R-1]	Minimal NETCONF version	NETCONF 1.1	NETCONF 1.1	
[R-2]	Minimal YANG version	YANG 1.1	YANG 1.1	
[R-3]	Number of simultaneous NETCONF sessions (note 1)	Multiple	N/A	ANs may need to support multiple sessions for different operational needs.
[R-4]	Number of management IP address (if NETCONF client supports redundancy)	1	N/A	
[R-5]	RFC 7895 [26] support	Yes	Yes	BBF has adopted YANG 1.1. This is considered a mandatory library.
[R-6]	RFC 6022 [20] support	Yes	Yes	For interoperability testing, the key assumption is that the Manager does not know NETCONF server implementation. That's why RFC 6022 [20] is needed so that the Manager application can discover what operations are supported on the AN

R-x	Area	NETCONF		Additional comments
		server	client	
				side. It also simplifies analysis of interoperability issues.
For information	IPv4 versus IPv6			The use of IPv4 versus IPv6 varies per operator and therefore left in this table for informative purposes only. The reader should refer to technology specific TRs for management of IP address type.

Note:

The maximum number of concurrent sessions may be limited by the total number of management sessions (CLI, NETCONF, etc.) and depends on specific deployments.

4.3 Datastores

A NETCONF configuration datastore is defined as the complete set of configuration data that is required to get a device from its initial default state into a desired operational state. The NETCONF configuration datastore does not include state data or action commands. RFC 6241 [22] defines 4 types of NETCONF configuration datastores (see Table 5).

The running configuration datastore holds the complete configuration currently active on the network device. Only one configuration datastore of this type exists on the device, and it is always present.

Additional configuration datastores may be implemented on the NETCONF Server to allow support of certain NETCONF capabilities.

Such configuration datastores are available only on devices that advertise the those capabilities.

Table 5 - NETCONF Datastore Capabilities

R-x	NETCONF Datastore Capabilities	RFC 6241 [22]	NETCONF		Use case / Notes
			server	client	
[R-7]	**Candidate**	8.3. Candidate Configuration Capability	note 1	N/A	UC-7 (note 2)
[R-8]	**Running**	8.2. Writable-Running Capability	MUST (note 3)	N/A	UC-1 (note 2)

R-x	NETCONF Datastore Capabilities	RFC 6241 [22]	NETCONF		Use case / Notes
			server	client	
[R-9]	**Startup**	8.7. Distinct Startup Capability	MAY	N/A	
[R-10]	**URL**	8.8. URL Capability	MAY	N/A	note 4

Notes:

1 In a SDN Management and Control hierarchy, the Candidate feature MUST be supported on at least one layer of the SDN hierarchy hosting a NC Server. The advised SDN layer is where the Operator decides to place the so called “source of truth” for the access network.

2 If a NETCONF Server does not support the Writable Running Configuration feature, the mandatory Running Data Store is read-only which requires the support of the Candidate Data Store on that NETCONF Server to <commit> changes on the Running Data Store.

3

If the Candidate and Writable Running features are supported by a NETCONF Server, any NETCONF Client, that establishes a NETCONF session with that NETCONF Server, MUST ensure that requests for Data Store (DS) changes preserve the consistency of the DS and avoid conflicts due to concurrent access of NETCONF Clients.
Section 4.1 briefly touches upon DS consistency and conflict avoidance among multiple clients.

4 Examples of URL usage include save configuration to external file, restore configuration from external files. This is an optional feature.

Figure 1 summarizes the NETCONF operations that can be applied to different NETCONF datastores.

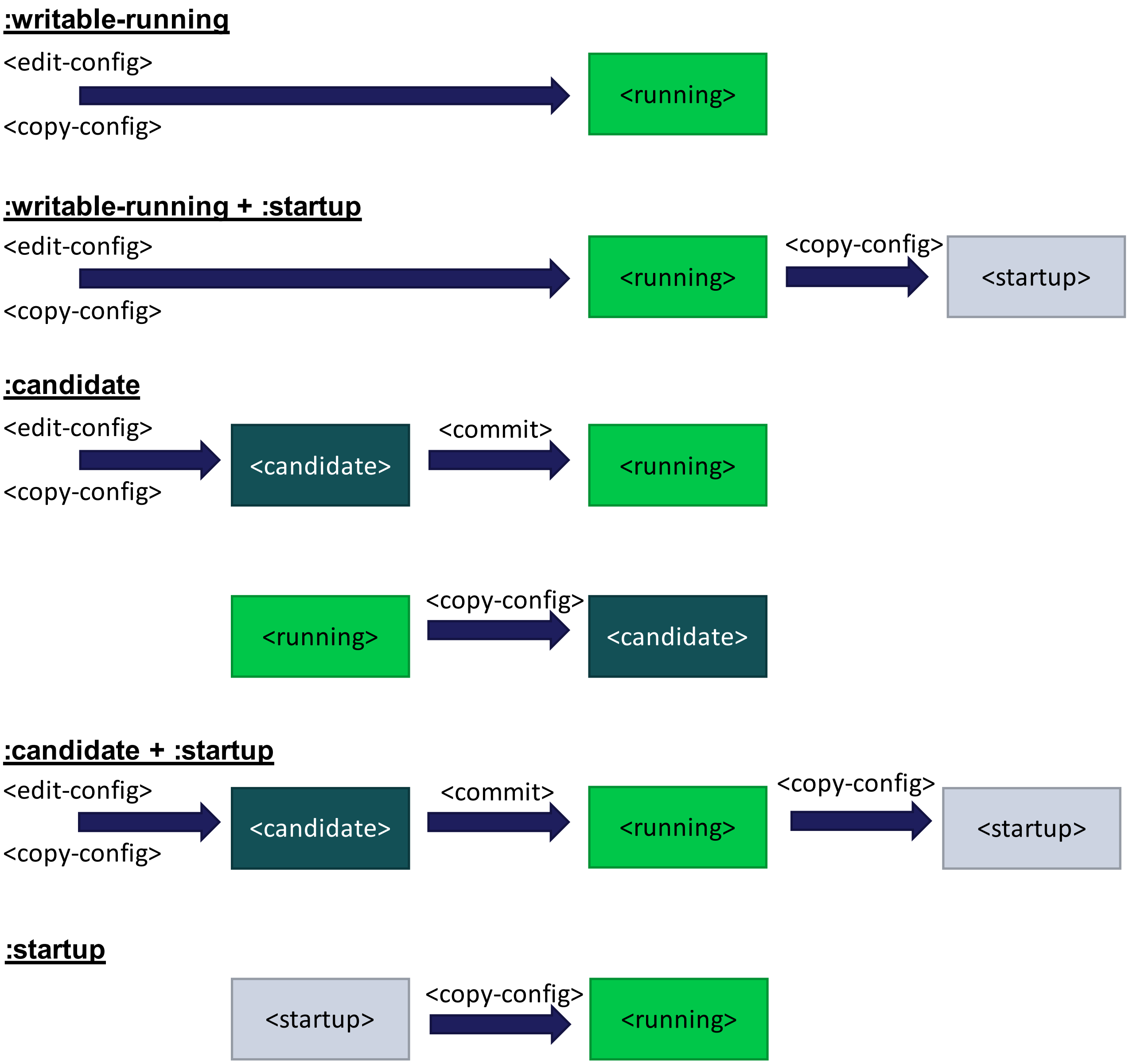


Figure 1 - NETCONF Data Store Operations

To apply changes to the <running> datastore, it is recommended to do a <copy-config> to <candidate>, then to <running>.

Copying <running> to <candidate> may be useful for a NC client to set <candidate> to an initial configuration rather than starting from blank. Copying <startup> to <running> may be useful for NC client to restore configuration to the locally saved one.

4.4 General NETCONF Conformance Requirements

This section contains conformance requirements generally applicable to the NETCONF features supported by a NETCONF Server or by a NETCONF Client.

[R-11] A NETCONF Server MUST generate responses and process requests for all applicable NETCONF features it supports in conformance with the relevant standard reference from section 4.5 to 4.10.

[R-12] A NETCONF Client MUST generate requests and process responses for all applicable NETCONF features it supports in conformance with the relevant standard reference from section 4.5 to 4.10.

The actual requirements about support of specific NETCONF features are specified for NC Server and Client implementations in the remaining sections of this chapter 4. The NETCONF standard references relevant for each requirement are reported as well.

4.5 NETCONF RPC Messages

NETCONF uses an RPC-based communication model where NETCONF peers use <rpc> and <rpc-reply> elements to provide transport-protocol-independent framing of NETCONF requests and responses. This section defines whether that functionality should be mandatory or not (and under what circumstances an optional NETCONF feature is mandatory on an AN).

Table 6 - NETCONF RPC Handling

R-x	RPC messages, attributes and values	RFC 6241 [22]	NETCONF		Use case / Notes
			server	client	
[R-13]	RPC messages	4. RPC Model, 4.1. <rpc> Element, Appendix B	MUST	MUST	UC-1
[R-14]	RPC replies	4.2. <rpc-reply> Element	MUST	MUST	UC-1
[R-15]	RPC error reporting	4.3. <rpc-error> Element, Appendix A	MUST	MUST	UC-1
[R-16]	RPC multiple error reporting	4.3. <rpc-error> Element	MAY	MUST	UC-1

R-x	RPC messages, attributes and values	RFC 6241 [22]	NETCONF		Use case / Notes
			server	client	
[R-17]	RPC error types Error Report data	4.3. <rpc-error> Element	MUST	MUST	UC-1
[R-18]	RPC error tag	4.3. <rpc-error> Element, Appendix A. NETCONF Error List	MUST	MUST	UC-1
[R-19]	RPC error severity	4.3. <rpc-error> Element	MUST	MUST	UC-1 (note 1)
[R-20]	RPC error-app-tag	4.3. <rpc-error> Element	MAY	MAY	
[R-21]	RPC error path	4.3. <rpc-error> Element	MUST	MUST	UC-1
[R-22]	RPC error info	4.3. <rpc-error> Element, Appendix A. NETCONF Error	MUST	MUST	UC-1
[R-23]	RPC error info to carry extended/implementation specific debugging information	4.3. <rpc-error> Element	MAY	MAY	UC-2 (note 2)

R-x	RPC messages, attributes and values	RFC 6241 [22]	NETCONF		Use case / Notes
			server	client	
[R-24]	<ok>	4.4. <ok> Element	MUST	MUST	UC-1
[R-25]	Pipelining	4.5. Pipelining	MUST	MUST	

Notes:

1 Support for 'warning' is not required as RFC 6241 [22] does not define any <error-tag> values which use it, rather it is reserved for future use.

2 Error-info is permitted to be used to carry extended/implementation specific debugging information. Error-info can be used to provide additional details about the error cause.

4.6 NETCONF Filtering

NETCONF supports filtering to narrow down the search of configuration elements.

Table 7 - NETCONF Filtering Options

R-x	Filtering	RFC 6241 [22]	NETCONF		Use case / Notes
			server	client	
[R-26]	Support subtree filtering based on below elements. • Namespace Selection • Attribute Match Expressions • Containment Nodes • Selection Nodes • Content Match Node	6.1. Overview, 6.2. Subtree Filter Components	MUST	MUST	UC-1
[R-27]	Namespace filtering	6.2.1. Namespace Selection	MUST	MUST	UC-1

R-x	Filtering	RFC 6241 [22]	NETCONF		Use case / Notes
			server	client	
[R-28]	Attribute Match Expression filtering	6.2.2. Attribute Match Expressions	MUST	MUST	UC-1
[R-29]	Containment Nodes	6.2.3. Containment Nodes	MUST	MUST	UC-1
[R-30]	Selection Nodes	6.2.4. Selection Nodes	MUST	MUST	UC-1
[R-31]	Content Match names	6.2.5. Content Match Nodes	MUST	MUST	UC-1
[R-32]	Xpath filtering	8.9. XPath Capability	MAY	MAY	UC-1

4.7 NETCONF Operations

Table 8 - NETCONF Operations

R-x	Operation	RFC 6241 [22]	NETCONF		Use case / Notes
			server	client	
[R-33]	<get-config>	7.1. <get-config>	MUST	MUST	UC-1

[R-34]	XPath modifications to Existing <get-config> and <get> Operations	6. Subtree Filtering”, 7.1. <get-config> 8.9.5 Modifications to Existing Operations	note 1	note 1	UC-1
[R-35]	subtree filtering	7.1. <get-config>	MUST	MUST	UC-1
[R-36]	<source> parameter set to <startup/> datastore	7.1. <get-config>	note 2	MUST	UC-3
[R-37]	<source> parameter set to <running/> datastore	7.1. <get-config>	MUST	MUST	UC-1
[R-38]	<source> parameter set to <candidate/> datastore	7.1. <get-config>	note 3	MUST	UC-7
[R-39]	<source> parameter set to URL	7.1. <get-config>	note 4	MUST	UC-5
[R-40]	<copy-config>	7.3. <copy-config>	note 5	MUST	UC-3
[R-41]	<source> parameter set to <startup/> datastore	7.3. <copy-config>	note 2	MUST	UC-3

[R-42]	<source> parameter set to <running/> datastore	7.3. <copy-config>	MUST	MUST	UC-1
[R-43]	<source> parameter set to <candidate/> datastore	7.3. <copy-config>	note 3	MUST	UC-7
[R-44]	<source> parameter set to URL	7.3. <copy-config>	note 4	MUST	UC-5
[R-45]	<target> parameter set to <startup/> datastore	7.3. <copy-config>	note 2	MUST	UC-3
[R-46]	<target> parameter set to <running/> datastore	7.3. <copy-config>	MUST (note 6)	MUST (note 6)	UC-1
[R-47]	<target> parameter set to <candidate/> datastore	7.3. <copy-config>	note 3	MUST	UC-7
[R-48]	<target> parameter set to URL	7.3. <copy-config>	note 5	MUST	UC-5
[R-49]	<edit-config>	7.2. <edit-config>	MUST	MUST	UC-1
[R-50]	<target> parameter set to <running/> datastore	7.2. <edit-config>	MUST (note 6)	MUST (note 6)	UC-1

[R-51]	<target> parameter set to <candidate/> datastore	7.2. <edit-config>	note 3	MUST	UC-7
[R-52]	<target> parameter set to URL	7.2. <edit-config>	note 5	MUST	UC-5
[R-53]	<default-operation> parameter set to “merge”, “replace” or “none”	7.2. <edit-config>	MAY	MUST	
[R-54]	<test-option> parameter set to “test-then-set”, “set”, “test-only”	7.2. <edit-config>	MUST	MUST	UC-6
[R-55]	<error-option> parameter set to “stop-on-error” or “continue-on-error”	7.2. <edit-config>	MAY	MUST	UC-1
[R-56]	<error-option> parameter set to “rollback-on-error”	7.2. <edit-config>	MUST	MUST	UC-1, UC-6
[R-57]	“operation” attribute set to “merge”, “replace”, “create”, “delete”, or “remove”	7.2. <edit-config>	MUST	MUST	UC-1
[R-58]	<delete-config>	7.4. <delete-config>	MUST	MUST	UC-1 (note 7)
[R-59]	<target> parameter set to <startup/> datastore	7.4. <delete-config>	note 2	MUST	UC-3

[R-60]	<target> parameter set to <candidate/> datastore	7.4. <delete-config>	note 3	MUST	UC-7
[R-61]	<target> parameter set to URL	7.4. <delete-config>	note 5	MUST	UC-5
[R-62]	<lock>	7.5. <lock>	MUST	MUST	UC-1, UC-8
[R-63]	<target> parameter set to <running/> datastore	7.5. <lock>	MUST (note 6)	MUST (note 6)	UC-1, UC-8
[R-64]	<target> parameter set to <startup/> datastore	7.5. <lock>	note 2	MUST	UC-1, UC-8
[R-65]	<target> parameter set to <candidate/> datastore	7.5. <lock>	note 3	MUST	UC-7, UC-8
[R-66]	<unlock>	7.6. <unlock>	MUST	MUST	UC-1, UC-8
[R-67]	<target> parameter set to <running/> datastore	7.6. <unlock>	MUST (note 6)	MUST (note 6)	UC-1, UC-8
[R-68]	<target> parameter set to <startup/> datastore	7.6. <unlock>	note 2	MUST	UC-1, UC-8
[R-69]	<target> parameter set to <candidate/> datastore	7.6. <unlock>	note 3	MUST	UC-7, UC-8
[R-70]	<get>	7.7. <get>	MUST	MUST	UC-1
[R-71]	XPath filtering	7.7. <get>	note 1	MUST	UC-1, UC-9, UC-13

[R-72]	subtree filtering	7.7. <get>	MUST	MUST	UC-1, UC-13
[R-73]	<close-session>	7.8. <close-session>	MUST	MUST	UC-1
[R-74]	<kill-session>	7.9. <kill-session>	MUST	MUST	UC-1
[R-75]	<commit>	8.3. Candidate Configuration Capability 8.3.4.1. <commit>	note 3	MUST	UC-7
[R-76]	<confirmed>	8.4. Confirmed Commit 8.4.5.1 <commit> Capability	note 3	MUST	UC-7
[R-77]	<confirm-timeout>	8.4.5. Modifications to Existing Operations 8.4.5.1 <commit>	note 3	MUST	UC-7
[R-78]	<persist>	8.4.5. Modifications to Existing Operations 8.4.5.1 <commit>	note 3	MUST	UC-7

[R-79]	<persist-id>	8.4.5. Modifications to Existing Operations 8.4.5.1 <commit>	note 3	MUST	UC-7
[R-80]	<discard-changes>	8.3.4.2. <discard-changes>	note 3	MUST	UC-6, UC-7
[R-81]	<cancel-commit>	8.4.4.1. <cancel-commit>	note 3	MUST	UC-6, UC-7
[R-82]	<validate>	8.6.4.1. <validate>	MUST	MUST	UC-6
[R-83]	<source> parameter set to <running/> datastore	8.6. Validate Capability	MUST	MUST	UC-6
[R-84]	<source> parameter set to <startup/> datastore	8.6. Validate Capability	note 2	MUST	UC-6
[R-85]	<source> parameter set to <candidate/> datastore	8.6. Validate Capability	note 3	MUST	UC-6, UC-7
[R-86]	<source> parameter set to URL	8.6. Validate Capability	note 4	MUST	UC-6

Notes:

1 This NETCONF Operation MUST be supported if Xpath filtering feature is supported, see [R-32] otherwise it is N/A.

2 This feature MUST be supported if the Startup Data Store is supported, see [R-9], otherwise it is N/A.

3 This feature MUST be supported if the Candidate Data Store is supported, see [R-7], otherwise it is N/A.

4 Mandatory if the URL feature is supported, see [R-10], otherwise it is N/A.

5 This feature MUST be supported if either the Startup Data Store is supported, see [R-9], or the Candidate Data Store is supported, see [R-7], otherwise it is N/A.

6 If the Writable Running feature is supported, it MUST be possible to use it as target.

7 The delete-config is a method to delete a datastore (running datastore is never allowed to be deleted). It may be desirable in an emergency situation to:

a) replace a startup datastore with a default startup datastore (with factory settings) by deleting the current startup datastore and copying a startup config containing factory settings to its running config. This would be a last-ditch troubleshooting scenario where you are trying to get services up and need to force a device to factory default.

b) copy a backed-up config to the running config in the case of a catastrophe that requires equipment replacement. The replaced equipment would need its startup DS deleted (and replaced with the backed up datastore for the equipment that is being replaced). Privileged users can be configured to limit who can delete a data store (see RFC 6536 [24] NACM Model deny-delete-config rule).

4.8 NETCONF Capabilities Exchange

Table 9 - NETCONF Capabilities

R-x	Capabilities Exchange	RFC 5277 [18], RFC 6241 [22]	NETCONF		Use case / Notes
			server	client	
[R-87]	Exchange capabilities as part of hello message exchange	RFC 6241 8.1. Capabilities Exchange	MUST	MUST	UC-1
[R-88]	Exchange of Notification capability	RFC 5277 3.1.1 Notification Capability	MUST	MUST	UC-1, UC-3
[R-89]	Exchange of Base NETCONF capability	RFC 6241 8.1. Capabilities Exchange	MUST	MUST	UC-1

[R-90]	Session ID inclusion in transmitted <hello> message	RFC 6241 8.1. Capabilities Exchange	MUST	MUST NOT	UC-1, UC-3
[R-91]	NETCONF session termination by: • NETCONF server upon receipt of <hello> message with the <session-id> element • NETCONF client upon receipt of <hello> message without the <session-id> element	RFC 6241 8.1. Capabilities Exchange	MUST	MUST	UC-1, UC-3
[R-92]	Exchange of Interleave capability	RFC5277 6. Interleave Capability	MAY	MAY	UC-1, UC-3

4.9 NETCONF Notifications

Table 10 - NETCONF Notifications

R-x	Notifications	RFC 5277	NETCONF		Use case / Notes
			server	client	
[R-93]	Generating/processing event notifications	2.1. Subscribing to Receive Event Notifications, 2.1.1. <create-subscription>	MUST	MUST	UC-3, UC-27
[R-94]	Defining vendor specific streams/subscriptions (above default NETCONF stream)	N/A	MAY	MAY	UC-3
[R-95]	Notification Replay Capability	3.3. Notification Replay	MAY	MAY	UC-3, UC-27 (note 12)

R-x	Notifications	RFC 5277	NETCONF		Use case / Notes
			server	client	
[R-96]	Interleave Capability	6. Interleave Capability	MAY	MAY	UC- 3 (note 2)

Notes:

1 While this functionality could be used for alarms/events synchronization, it has limitations in that it can supply a limited number of notifications and for this reason is marked optional.

2 This capability helps scalability by reducing the total number of NETCONF sessions required by a given human operator or management application.

4.10 NETCONF Subscriptions

Table 11 - NETCONF Subscriptions

R-x	Notifications	RFC 5277 [18]	NETCONF		Use case / Notes
			server	client	
[R-97]	Stream	2.1.1. <create-subscription>, 4. XML Schema for Event Notifications	MAY	N/A	UC-3
[R-98]	Filter	2.1.1. <create-subscription>, 4. XML Schema for Event Notifications	MAY	N/A	UC-3

R-x	Notifications	RFC 5277 [18]	NETCONF		Use case / Notes
			server	client	
[R-99]	Start Time	2.1.1. <create-subscription>, 4. XML Schema for Event Notifications	MAY	N/A	UC-3
[R-100]	Stop Time	2.1.1. <create-subscription>, 4. XML Schema for Event Notifications	MAY	N/A	UC-3

5 NETCONF Client and Server Configuration

5.1 Call Home and Zero-Touch Provisioning

The commissioning of an Access Node (AN) like an OLT is a complex sequence of tasks that Network Service Providers aim at automating. In legacy systems it was achievable up to a certain level using proprietary EMS and a collection of proprietary/home-made management tools. However, the Call-Home mechanism specified in RFC 8071 [27] provides a standard method to achieve a much higher level of automation for installing a new AN in the access network.

The Call-Home mechanism can be used at the M_{inf} interface of the CloudCO, between the AN and the virtual Access Node Management Function (vANMF), where the vANMF is either a legacy NETCONF capable management system (e.g., an EMS with NETCONF interfaces), or, as highlighted in Figure 2, an Access SDN Management & Control system or a BAA Layer:

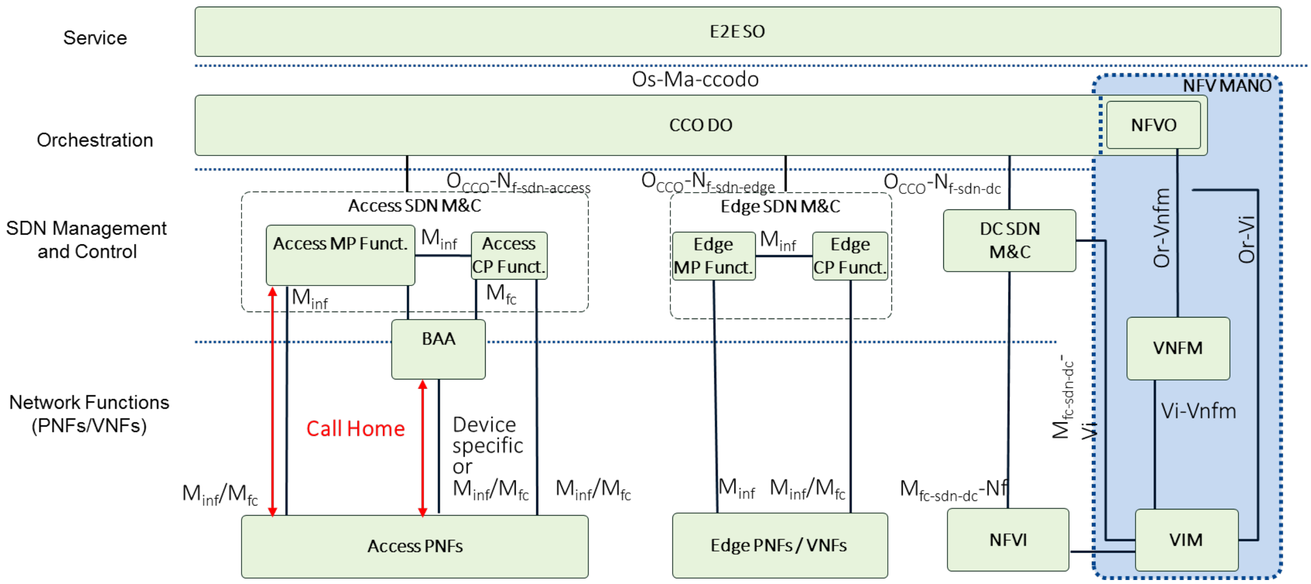


Figure 2 - Call-Home at the M_{inf} interface

In order to activate Call-Home mechanism, an IP connection first needs to be activated between the AN and the corresponding vANMF instance.

TR-301 [1] Section 16.5 describes how to establish connectivity between a DPU and its PMA. That procedure can be generalized and applied to establish connectivity between an AN and the corresponding vANMF within a management element of the SDN hierarchy.

The AN management architecture connects several ANs to a management system via the management network. In the context of this section, the vANMF can either be a NETCONF capable manager used, for example, for OLT commissioning using the Call-Home mechanism, or a full-featured SDAN controller featuring a BAA function and associated functionality.

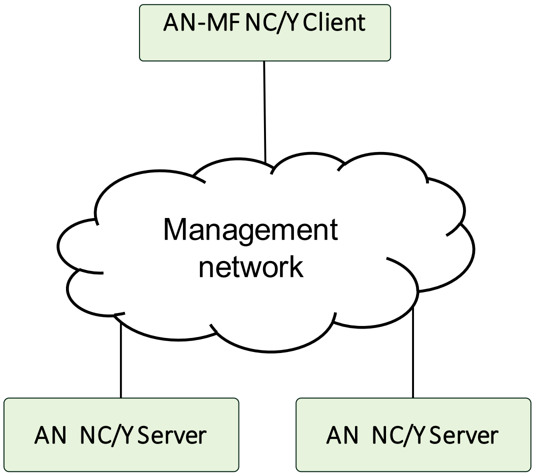


Figure 3 - AN Management Network

This section describes the overall management architecture and specifies a set of high-level functional requirements.

The vANMF uses NETCONF and YANG data models to manage the AN. The detailed YANG models are documented in separate Technical Reports.

[R-101] The AN and vANMF MUST support NETCONF.

[R-102] The AN and vANMF MUST support YANG.

NETCONF requires a persistent, reliable connection that supports authentication, data integrity, confidentiality, and replay protection. For the AN-vANMF connection, SSH or TLS are used to provide the functionality to meet the connection requirements of NETCONF as described in Section 5.

Permanent connectivity between the AN and vANMF is essential to the management of the AN. There is a need for a regular keep-alive between these 2 entities to know when there is management plane connectivity; it is not sufficient to rely on the Dying Gasp to deduce that management connectivity has been lost.

[R-103] The vANMF MUST periodically transmit Transmission Control Protocol (TCP) keep-alive messages, as defined in RFC 1122 [10], to the AN on the TCP connection established for call home when the connection is otherwise idle.

[R-104] The AN MUST respond to keep-alive messages from the vANMF on the TCP connection established for Call-Home.

[R-105] The vANMF SHOULD support configuration of the timing of TCP keep-alive messages and the threshold for detecting keep-alive failures.

[R-106] The AN MUST periodically transmit TCP keep-alive messages, as defined in RFC 1122 [10] , to the vANMF on the TCP connection established for Call-Home when the connection is otherwise idle.

[R-107] The vANMF MUST respond to keep-alive messages from the AN on the TCP connection established for Call-Home.

[R-108] In the event of a keep-alive failure, the AN MUST re-initiate the NETCONF Call-Home procedure as specified in the discovery process in Section 5.1.1.3, beginning with step 6b.

[R-109] If the AN fails to reconnect to the vANMF, it MUST restart the discovery process beginning with the AN Discovery message.

[R-110] The AN SHOULD support configuration of the timing of TCP keep-alive messages and the threshold for detecting keep-alive failures.

5.1.1 vANMF Discovery and NETCONF Session Establishment

When an AN is powered up, it needs to discover its vANMF to which the AN would associate. Both the installation process and vANMF discovery must be supported without requiring manual configuration of IP addresses or other parameters in the AN. vANMF discovery requires the following steps:

- Upon powering up, an AN creates a link-local IPv6 address on its management interface. The AN sends an vANMF Discover message formatted as a set of options within a DHCPv6 Information-request message to discover the address of the vANMF to which it should establish a connection. The AN may also send other IPv4 DHCP and/or DHCPv6 messages to discover vANMF Information and to establish a routable IPv4 or IPv6 address for the AN.
- A DHCP/DHCPv6 server responds to an vANMF Discover with an vANMF Offer, which embeds the required vANMF Information as sub-options within a DHCP or DHCPv6 message.
- The AN initiates a TCP connection to the vANMF as defined in NETCONF Call-Home and RESTCONF Call Home RFC 8071 [27]. The vANMF responds by initiating TLS to the AN. As part of the TLS session initiation, the AN and the vANMF authenticate each other.
- The vANMF initiates a NETCONF session over the established TLS session.

These steps are defined in detail in the following subsections.

5.1.1.1 vANMF Discovery Architecture

5.1.1.1.1 Use of DHCP

DHCP plays a central role in for AN-vANMF discovery, as the DHCP (or DHCPv6 or stateless DHCPv6) server provides the vANMF reachability information to the AN, that is the EMS' IP address or FQDN.

If the AN and vANMF are within the same Layer 2 broadcast domain, they can communicate using IPv6 link-local IP addresses and there is no need for the AN to establish a routable IP address. Otherwise, the AN establishes a routable IPv4 or IPv6 address (using SLAAC, DHCPv6 or DHCP) to communicate with the vANMF.

The vANMF Information sub-option (section 5.1.1.2) used to convey vANMF reachability information includes fields for Domain name, IPv4 and IPv6 addresses. While the address version used will typically match the DHCP version, there are valid exceptions. For instance, IPv6 link-local addressing may be used for AN-vANMF communication in a management network that uses IPv4 for other applications including DHCP.

5.1.1.1.2 AN Identification

The AN must identify itself to the DHCP server so to discover its vANMF. The DHCP Unique Identifier (DUID) defined in RFC 8415 [29] provides an identification code that is both globally unique and stable – no two ANs should have the same DUID, and an AN's DUID should not change over time.

RFC 8415 [29] defines three ways to construct the DUID. In the AN, DUID construction is limited to the DUID-EN option, with the vendor's IANA registered private enterprise number followed by a unique serial number assigned by the vendor. Unless specified by another TR, the DUID Unique Identifier is formatted as an ASCII string-based serial number. For example, TR-301 [1] does identify the formatting of the DUID and that specification's format of the DUID takes precedence over the formatting of the DUID's unique identifier specified in this Technical Report.

[R-111] The AN MUST contain a DHCP Unique Identifier (DUID).

[R-112] The DUID MUST be in DUID-EN format as defined in RFC 8415 [29]

[R-113] The DUID Unique Identifier MUST format the AN's serial number as an ASCII string unless specified differently in another TR (e.g., TR-301 [1] for DPUs).

5.1.1.1.3 Initiation of NETCONF connection and AN-vANMF Authentication

Before the AN and the vANMF can establish a NETCONF connection, they need to authenticate each other's identities. The mutual authentication takes place using X.509 [39] certificates, with each node tracing the other node's certificates to a trust anchor. To validate against a trust anchor, the AN contains one or more trusted certificates (e.g., pinned certificates, CA root certificates, intermediate certificates, or issuing certificates) that are either pre-loaded in the AN or provided to the AN by a secure means outside the scope of this specification.

The vendor's enterprise identifier and the AN serial number together form a globally unique identifier in the IDevID certificate's subject field. These two values are concatenated in the serialNumber attribute and separated by a hyphen. The serial number is formatted as a character string containing all characters of the serial number, including the vendor ID if any, as it would appear on a label. This format of the IDevID certificate's subject field use the format specified in this Technical Report unless specified by another TR. For example, TR-301 does identify the formatting of the IDevID certificate's subject field and that specification's format of the IDevID certificate's subject field takes precedence over the formatting of the IDevID certificate's subject field specified in this Technical Report.

[R-114] The AN MUST provide an IDevID certificate to the vANMF.

[R-115] The AN's certificate MUST be unique to the individual AN unit.

[R-116] The AN' serial number SHOULD be formatted as a character string in the IDevID subject field.

[R-117] The IDevID subject field MUST include the serialNumber attribute.

[R-118] The IDevID subject field's serialNumber attribute MUST include the following elements, concatenated in order:

- The vendor's IANA registered enterprise number, formatted as a decimal value string;
- Hyphen (ASCII 0x2D);
- The AN' serial number, formatted as a character string and including the vendor ID if present.

[R-119] The vANMF MUST ensure that the presented AN certificate has a valid chain of trust to a preconfigured trust anchor, and that the reference identifier from the certificate matches a preconfigured value before establishing a NETCONF connection.

[R-120] The AN MUST verify the integrity of the certificate presented by the vANMF, either by performing path certificate validation per RFC 7589 [25] and RFC 5280 [19], OR by another trusted mechanism such as matching the presented certificate fingerprint against the configured certificate fingerprint.

[R-121] The vANMF MUST verify the integrity of the certificate presented by the AN, either by performing path certificate validation per RFC 7589 [25] and RFC 5280 [19], OR by another trusted mechanism such as matching the presented certificate fingerprint against the configured certificate fingerprint.

RFC 7589 [25] specifies the use of NETCONF over TLS with mutual X.509 [39] authentication. The AN initiates the TCP connection to the vANMF using RFC 8071 [27] rather than waiting for the vANMF to initiate the connection.

5.1.1.2 Discovery Messages

The discovery process uses two message types, both of which are carried within DHCP or DHCPv6 messages. AN Discover is sent from the AN to initiate the process. vANMF Offer is sent from the DHCP/DHCPv6 server to provide information on one or more vANMF to the AN.

5.1.1.2.1 AN Discover message

The AN Discover message is sent by the AN to identify itself and to request the IP address of the vANMF with which it should connect. It can be formatted as any of several DHCP or DHCPv6 messages.

[R-122] The AN MUST send AN Discover formatted as a DHCPv6 Information-request message as part of the discovery process.

[R-123] The AN MAY send AN Discover formatted as a DHCPv6 Solicit, DHCPINFORM, and/or DHCPDISCOVER message as part of the discovery process.

The AN Discover message includes the options shown in Table 12 and described below.

Table 12 - DHCPv6 and DHCP options used for AN Discover message

Description	DHCPv6 option			DHCP option (IPv4)		
	#	Name	Ref	#	Name	Ref
DHCP Unit Identifier (DUID)	1	Client Identifier	RFC 8415 [29]	61	Client Identifier	RFC 4361 [15]
Requested parameters	6	Option Request	RFC 8415 [29]	55	Parameter Request List	RFC 2132 [12]
BBF-specific information	17	Vendor-Specific Information	RFC 8415 [29]	125	Vendor-Identifying Vendor-Specific Information	RFC 3925 [14]

DHCP Unique Identifier (DUID) is mandatory in all AN Discover messages. Note that in addition to the DUID, the AN Discover message can include a Registration ID as described below:

[R-124] The AN MUST provide a DUID as defined in RFC8415 [29] as option 1 in all DHCPv6 AN Discover messages.

[R-125] The AN MUST provide a DUID as defined in RFC4361 [15] as option 61 in all DHCP AN Discover messages.

The requested parameters option is used in DHCPv6 and DHCP client messages to identify to the DHCP server the network information sought by the client. For AN Discover, the requested information includes the vendor-specific information option. The AN may request other information in addition to this option.

[R-126] The AN MUST request option 17 (Vendor-specific Information) in option 6 of all DHCPv6 AN Discover messages.

[R-127] The AN MUST request option 125 (Vendor-Identified Vendor-specific information) in option 55 of all DHCP AN Discover messages.

The Vendor-specific Information option is used to provide BBF-specific information as identified by the BBF enterprise-number (3561) assigned by IANA. The formatting used for the Vendor-specific Information option for DHCPv6 and DHCP messages is described in Appendix I. For AN Discover, the option is included using the BBF enterprise-number value of 3561 and a SuboptionN-code of 194.

[R-128] The AN MUST include option 17 (Vendor-Specific Information) with Enterprise-number = 3561 and SuboptionN-code = 194 in all DHCPv6 AN Discover messages.

[R-129] The AN MUST include option 125 (Vendor-Identifying Vendor-Specific Information) with Enterprise-number = 3561 and SuboptionN-code = 194 in all DHCP AN Discover messages.

5.1.1.2.2 vANMF Offer message

vANMF Offer is sent by a DHCPv6 or DHCP server to the AN to provide reachability information for the vANMF with which the AN should connect. It can be formatted as DHCPv6 Reply, DHCPv6 Advertise, DHCPOFFER, or DHCPACK, depending on the type of server and the DHCP message type to which it is replying.

One or more vANMF Information sub-options are encapsulated within the Vendor-specific Information option (option 17) in DHCPv6 messages and within the Vendor-Identifying Vendor-specific Information option (option 125) in DHCP messages, with the BBF Enterprise-number value of 3561 used to identify the vendor. The formatting used for the options 17 and 125 is described in Appendix A. Each instance of vANMF Information is identified using the SuboptionN-code of 193.

[R-130] DHCPv6 server MUST include option 17 (Vendor-specific Information) with Enterprise-number = 3561 and at least one instance of SuboptionN-code = 193 (vANMF Information) in all vANMF Offer messages.

[R-131] A DHCP server MUST include option 125 (Vendor-Identifying Vendor-specific Information) with Enterprise-number = 3561 and at least one instance of SuboptionN-code = 193 in all vANMF Offer messages.

The TLVs used in the SuboptionN-data field for SuboptionN-code = 193 (vANMF Information) are shown in Table 16-3. Any Type value not listed in the table is reserved for future use.

Table 13 - TLVs for vANMF Information

Type	Length	Value	Description
16	4 octets	IPv4 address	IPv4 address of the vANMF being offered
17	16 octets	IPv6 address	IPv6 address of the vANMF being offered
18	1 octet	PCP	Priority Code Point value to be used by the AN for management traffic.

19	Variable	Domain name	The domain name of the vANMF being offered.

The TLVs in are described below.

- IPv4 address, IPv6 address and Domain name each provide the reachability information in different forms to allow the AN to establish a TCP connection with the vANMF . Only one of these TLVs is typically provided.
- PCP provides the Priority Code Point value to be used by the AN for all traffic sent to the vANMF.

[R-132] An vANMF Information suboption MUST include either the IPv4 address, the IPv6 address or Domain name of the associated vANMF.

[R-133] If an vANMF Information suboption includes the Domain name TLV, the AN MUST use the Domain name and ignore any IP address TLVs in the same vANMF Information suboption.

[R-134] If an vANMF Information suboption includes the Domain name TLV, the Value field for the TLV MUST be encoded as defined in section 3.1 of RFC 1035 [9].

[R-135] If an vANMF Information suboption includes the Domain name TLV, the AN MUST also acquire one or more DNS server addresses (via DHCP, DHCPv6, NDP, etc.).

[R-136] The AN MUST support DNS lookup over IPv4 and IPv6.

[R-137] If the AN needs to use a PCP value other than 0 for traffic sent to the vANMF, the associated vANMF Information sub-option MUST include the required PCP.

5.1.1.3 Discovery Process

The vANMF discovery process starts when the AN is powered-on. The discovery process takes place as follows:

1. The AN generates an IPv6 link-local address on its management uplink interface and performs Duplicate Address Detection. The management uplink interface can be either a dedicated physical interface (although unlikely), or a specific management VLAN on the uplink interface (common case) or the VLAN 0 (untagged frames) on the uplink interface.

2. Using its link-local address as source address, the AN sends an AN Discover message formatted as DHCPv6 Information-request.

   a. The AN can also send the AN Discover message formatted as DHCPDISCOVER and/or DHCHPv6 Solicit to solicit a routable address and/or to initiate discovery on IPv4 networks. It can also use SLAAC to generate a routable IPv6 address.

3. A DHCP or DHCPv6 server maps the DUID and/or registration ID encapsulated in the AN Discover message to one or more vANMF.

4. The DHCP/DHCPv6 server responds with an vANMF Offer message containing one vANMF Information suboption for each vANMF being offered. The vANMF Information suboptions are listed under the Vendor-Specific Information option in the order in which the AN should attempt to connect.

5. If the vANMF Information suboption(s) in the vANMF Offer contain Domain names, or if the vANMF address(es) are IPv4 or IPv6 routable addresses, the AN must complete the process of generating a corresponding routable address before proceeding to step 6. However, if the vANMF IP address(es)

contained in the vANMF Offer are IPv6 link-local addresses, the AN must proceed with step 6 using its own IPv6 link-local address.

6. The AN performs the following steps when establishing a NETCONF connection with an vANMF. In the steps below, the vANMF is the NETCONF/TLS Client and the AN is the NETCONF/TLS Server.

   a. If the current vANMF Information suboption contains a Domain name, the AN performs a DNS lookup to resolve the domain name to an IP address.
   b. The AN opens a TCP connection on port 4335 by default or on another configurable port number to the offered vANMF.
   c. The vANMF opens a TLS connection to the AN as defined in RFC 7589 [25].
   d. The AN and the vANMF each perform path certificate validation as per RFC 7589 [25] and RFC 5280 [19]. The AN ensures that the presented vANMF certificate has a valid chain of trust to a pre-configured trust anchor and derives the vANMF' NETCONF username from its certificate. The vANMF ensures that the presented AN certificate has a valid chain of trust to a pre-configured trust anchor and that the AN's reference identifier matches the identifier pre-configured in the vANMF. If either the vANMF or the AN cannot authenticate the other's identity, the connection is terminated.
   e. After mutual authentication has completed and the TLS connection established, the AN immediately starts the NETCONF server.
   f. Once each node has authenticated the other, the vANMF and the AN can begin to exchange NETCONF messages sent as TLS application data, as specified in RFC 7589 [25].
   g. If the AN cannot successfully establish a NETCONF connection with the vANMF , it closes the TLS and/or TCP connections and attempts to re-establish the connection with the vANMF.

[R-138] If the AN fails to connect to the offered vANMF, it MUST restart the discovery process beginning with the AN Discovery message.

5.1.2 Management VLAN

All discovery traffic and AN - vANMF traffic uses a predefined management VID in the range from 0 (P-tagged) to 4094 at the AN interface. If the network uses a different VLAN for management traffic to and from the vANMF, it is the responsibility of the Ethernet/IP aggregation equipment which the AN connects to perform tag operations that place the upstream and downstream AN – vANMF traffic on the correct VLANs. The Ethernet/IP aggregation equipment should perform any VLAN tag manipulations necessary to convert between the management VLAN used by the AN and the management VLAN used by the network.

[R-139] The AN MUST support use of a predefined VID in the range of 0 - 4094 for all discovery messages and all traffic sent to the vANMF.

[R-140] The AN MUST support use of a predefined VID p-bit value for all discovery messages and all traffic sent to the vANMF.

[R-141] If the vANMF Information contains a PCP TLV, the AN MUST use the associated PCP value for AN-vANMF traffic.

Notes:

1 This feature MUST be supported if tls-initiate is supported, see [R-163], otherwise it is N/A.

2 For SSH connections, it is mandatory to support at least one of the following NETCONF client identity authentication methods: password, certificate, public keys, hostbased client identity.

The option of no NETCONF client authentication MUST be supported as well but it should not be used for network deployments. It may be used for other applications like lab testing.

At minimum, Public Key encryption should be supported with NETCONF over SSH

3 If either tls-initiate, see [R-163] or tls-listen, see [R-189], are supported, it is mandatory to support at least one of the following NETCONF client identity authentication methods: certificate, raw public keys, PSKs. Otherwise this feature is N/A.

4 For SSH connections, it is mandatory to support at least one of the following NETCONF server authentication methods: SSH host keys, Certification Authority certificates, server certificates.

At minimum, Public Key encryption MUST be supported with NETCONF over SSH.

5 If either tls-initiate, see [R-163], or tls-listen, see [R-189], are supported, it is mandatory to support at least one of the following NETCONF server authentication methods: Certification Authority certificates, server certificates, raw public keys, PSKs. Otherwise this feature is N/A.

6 This feature MAY be supported if tls-initiate is supported, [R-163], otherwise it is N/A.

7 This feature MUST be supported if tls-listen is supported, see [R-189], otherwise it is N/A.

6 NETCONF Security

NETCONF is defined to be a secure protocol. Two options are defined in standards: NETCONF over SSH or NETCONF over TLS.

Table 14 is informative and compares the two NETCONF options and implementation requirements as described in relevant IETF RFCs. Certificate based security (SSH or TLS) requires a Public-Key Infrastructure Certification Authority (PKI/CA} to handle certificate validation and/or revocation.

This Technical Report assumes an AN must interoperate with a 3rd party open-source manager. (e.g., Open Daylight, OpenStack and others) and that Open-source managers comply with IETF NETCONF requirements.

IETF mandates support of NETCONF over SSH (however makes TLS optional).

This Technical Report recommends following IETF NETCONF requirements so as to promote higher chance of interoperability with managers that are IETF NETCONF compliant.

Table 14 - NETCONF over SSH and NETCONF over TLS Comparison

	NETCONF over SSH	NETCONF over TLS
Testability of encrypted NETCONF messages	Please refer to Section 5 of this Technical Report	Please refer to Section 5 of this Technical Report
IETF Default Protocol for NETCONF	Indicated as mandatory in RFC 6241 [22]	Indicated as optional in RFC 6241 [22]
Keep-alive of session support	Yes	Yes
Availability of open source	Yes	Yes
Support of Pinned Keys or certificates	Yes (Pinned certificates) when RFC 6187 [21] SSH with certificates implemented Yes (Pinned Host Keys)	Yes (Pinned certificates)
Infrastructure	PKI/CA infrastructure when using RFC 6187 [21] SSH with certificates a	PKI/CA infrastructure for validation and/or revocation required by X.509 [39] which is used for TLS
3rd party AAA authentication	Yes	No

* At the time of publication, these Open SDN controllers did not support NETCONF over TLS.

Table 15 and Table 16 summarize NETCONF security requirements for NETCONF over SSH and TLS.

Table 15 - NETCONF over SSH Security Options

R-x	Area	Disposition	Notes

R-x	Area	Disposition	Notes
[R-142]	SSHv2 support	MUST	note 1
	Methods supported		
[R-143]	Public Key Based	MUST	note 1
[R-144]	Host based method support	MAY	
[R-145]]	Password method support	MAY	
[R-146]	Pluggable Authentication Module support (with challenge-response authentication)	MAY	note 2
[R-147]	Pluggable Authentication Module support (with password and challenge-response authentication) needed	MAY	note 2
[R-148]	RFC 6187 [21] X.509 Certificates	MAY	

Notes:

1 Minimum requirement

2 Pluggable Authentication Module is a generic framework for AAA.

Table 16 - NETCONF over TLS Features

R-x	Area	Disposition	Notes
[R-149]	TLS1.2 support	note 1	note 2

Notes:

1 TLS1.2 MUST be supported when TLS is implemented

2 Minimum requirement

Table 17 describes certificate requirements (for NETCONF over TLS and NETCONF over SSH (when implementing certificates).

Table 17 - Certificate Requirements

R-x	Certificate Validation	Disposition	Notes
	Certificate path validation		
[R-150]	Client/Server use X.509 [39] certificate path validation per RFC 5280 [19] to verify integrity of the certificate presented by the peer	note 1	
[R-151]	Naming Chaining	MAY	
[R-152]	Key Chaining	MAY	
[R-153]	Certification Revocation	MAY	
	Derivation of NETCONF server username (maptype options)		
[R-154]	username is derived from auxiliary data	MAY	
[R-155]	username derived from subjectAltName's rfc822Name field	MAY	

[R-156]	username derived from subjectAltName's dNSName field	MAY	
[R-157]	username derived from subjectAltName's iPAddressfield	MAY	
[R-158]	username derived from matching subjectAltName rfc822Name, dNSname, IPAddress (san-any)	MAY	
[R-159]	username derived from CommonName	SHOULD NOT	
	Certificate Installation		
[R-160]	Method used to secure the installed certificate (TPM or Other method to obfuscate certificate)	note 2	note 3

Notes:

1. This certificate path validation MUST be supported when TLS is implemented or SSH with certificates is implemented
2. This certificate installation method MUST be supported, as a minimum, when certificate-based security is used
3. draft-netconf-keystore recommends non-operating system level data store

6.1 Level of Privacy

NETCONF is a secure protocol that can support different security goals.

Table 18examines different security goals along with possible options that can used to accomplish them. At minimum, all NETCONF sessions need to be secured and encrypted (privacy should be supported).

Table 18 - NETCONF Privacy

NETCONF Security Goal Priorities	Use cases
Privacy (encryption only)	At minimum, all NETCONF sessions must be encrypted. Encryption can be accomplished with Host Keys (NETCONF over SSH). Certificate based encryption (SSH or TLS) can be accomplished using self-signed certificates on NETCONF client/server.
Authentication of NETCONF client (e.g.,source-organization)	This use case is optional. This use case can be accomplished via certificated based authentication (SSH or TLS). The AN authenticates the NETCONF client certificate (e.g., source/organization, etc.).
Mutual Authentication (NETCONF server/client)	This use case is optional and can be implemented either using NETCONF over TLS (path validation of CA authority) or SSH (see draft-ietf-netconf-ssh-client-server). This use case can be accomplished via certificate based authentication where both the AN and NETCONF client authenticate each other’s certificates.
3rd party authentication	This use case is optional and available with NETCONF over SSH. This use case uses Backend Authentication, Authorization, and Accounting (AAA) servers to authenticate users on a NETCONF session.

6.2 Ciphersuites

Ciphersuites are expected to vary per operators. The security requirements may vary accordingly.

7 User Security

RFC 8341 [28] defines a standard mechanism to restrict NETCONF protocol access for particular users to a preconfigured subset of all available NETCONF protocol operations and content. Roles are modeled as a set of one or more "rules" in the NETCONF Access Control Model YANG Container along with default behavior for Read, Write and Exec actions.

8 NETCONF Testing

There are multiple options for testing NETCONF:

1. Use 3rd party tools capable of establishing a NETCONF over an SSH/TLS session with an AN and retrieving AN configuration and state data as well as modifying the AN configuration via NETCONF. This is considered the simplest solution to use for compliance purposes (however insufficient to capture interoperability issues).
2. The AN logging decrypted NETCONF message exchange (outgoing/incoming messages).
3. The use of SSH/TLS proxies to perform a "man-in-the-middle" spoof and capture the NETCONF message exchanges between NETCONF client and server.

Option 3 is viewed the more reliable method to use (captures the full exchange of NETCONF messages). BBF has adopted this 3rd method for interop testing.

Appendix I.DHCPv6 and DHCP Vendor Specific Option Formatting

This Appendix describes how BBF-specific information fields used for AN and vANMF discovery are mapped to a DHCPv6 or a DHCP message.

I.1 DHCPv6 Option 17 Formatting

The AN uses Vendor-specific Information option 17 to provide AN Discover information regarding its certifications and supported features over DHCPv6 for discovery. The DHCPv6 server uses the same option to provide one or more AN Offers, including certification, supported features, and other data, to the AN. The format for the Vendor-specific Information option is shown below and described in RFC 8415 [29]. Multiple instances of this option can be included in a DHCPv6 message with each instance containing a unique enterprise-number.

Table 19 - DHCPv6 Option 17 fields

```
                     1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

Option-code = 17	Option-len	
Enterprise-number = 0x0DE9		
Suboption1-code	Suboption1-len	
Suboption1-data …		
Suboption2-code	Suboption2-len	Optional
Suboption2-data …		
…		

Field	Length	Description
Option-code	2 octets	Vendor-specific Information (17)
Option-len	2 octets	Total length of all following option data in octets. This value is exclusive of the option-code and option-len octets.
Enterprise-number	4 octets	The vendor's 32-bit Enterprise Number as registered with IANA. The Broadband Forum value is 3561 (0x0DE9).
SuboptionN-code	2 octets	For AN – vANMF discovery, the type of discovery message. 194: AN Discover, sent by the AN to request discovery information. Only one instance of this suboption type may occur within a DHCPv6 message. 193: vANMF Information, sent by a DHCPv6 server to provide AN reachability information to a AN. One instance of this suboption type may occur for each vANMF being offered.
SuboptionN-len	2 octets	The length of the SuboptionN-data field in octets.
SuboptionN-data	Variable	The SuboptionN-data field contains information specific to the SuboptionN-code field. See Table 20 for formatting.

The suboptionN-data fields are formatted as a series of Type/Length/Values (TLVs). The TLVs have the format shown below.

Table 20 - Format of BBF-specific TLV fields for DHCPv6 Option 17

1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1	
Type1	Length1
Value1 …	
Type2	Length2
Value2 …	

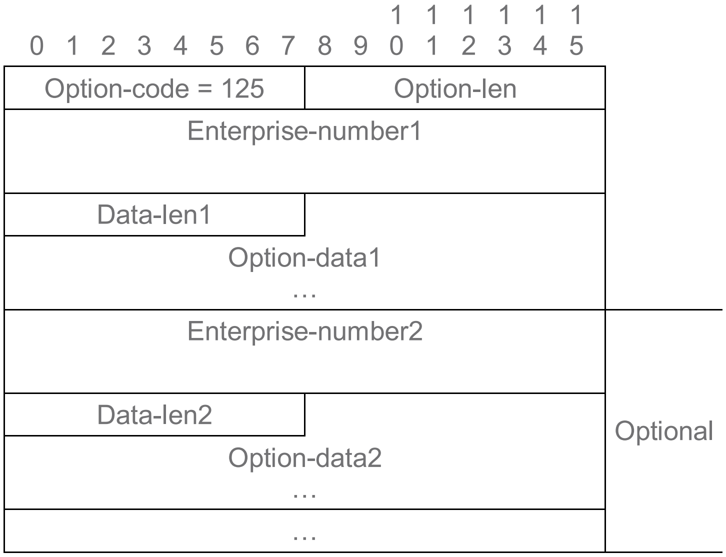
Field	Length	Description

TypeN	2 octets	The TypeN field identifies the type of data contained in the ValueN field.
LengthN	2 octets	The LengthN field contains the length of the ValueN field in octets.
ValueN	Variable	The Value field is zero or more octets and contains information specific to the Type. The format and length of the Value field are determined by the Type and Length fields.

I.2 DHCP (IPv4) Option Formatting

The AN uses Vendor-Identifying Vendor-specific Information option 125 to provide AN Discover information regarding its certifications and supported features over DHCP for discovery. The DHCP server uses the same option to provide one or more vANMF Offers, including certification, supported features, and other data, to the AN. The format for the Vendor-Identifying Vendor-specific option is shown below and described in RFC 3925 [14]. The option may also be divided into a series of smaller options as defined in RFC 3396 [13].

Table 21 - DHCP Option 125 fields



Field	Length	Description
Option-code	1 octet	Vendor-Identifying Vendor-specific Information (125)
Option-len	1 octet	Total length of all following option data in octets. This value is exclusive of the option-code and option-len octets.
Enterprise-numberN	4 octets	The vendor's 32-bit Enterprise Number as registered with IANA. The Broadband Forum value is 3561 (0x0DE9).
Data-lenN	1 octet	Length of Option-dataN field in octets
Option-dataN	Variable	Vendor-specific option data. When Enterprise-numberN = 3561 and used for DPU-PMA discovery, see Table A-4 for formatting.

When used with the Broadband Forum enterprise-number for AN – vANMF discovery, the option-data field for option 125 is formatted as one or more suboptions as shown in Table 22.

Table 22 - BBF-specific Option-data field

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td></td></tr>
<tr><td colspan="8">Suboption1-code</td><td colspan="8">Suboption1-len</td><td></td></tr>
<tr><td colspan="16">Suboption1-data
…</td><td></td></tr>
<tr><td colspan="8">Suboption2-code</td><td colspan="8">Suboption2-len</td><td rowspan="3">Optional</td></tr>
<tr><td colspan="16">Suboption2-data
…</td></tr>
<tr><td colspan="16">…</td></tr>
</table>

Field	Length	Description
SuboptionN-code	1 octet	For AN – vANMF discovery, the type of discovery message. 194: AN Discover, sent by the AN to request discovery information. Only one instance of this suboption type may occur within a DHCP message. 193: vANMF Information, sent by a DHCP server to provide AM reachability information to a AN. One instance of this suboption type may occur for each vANMF being offered.
SuboptionN-len	1 octet	The length of the SuboptionN-data field in octets.
SuboptionN-data	Variable	The SuboptionN-data field contains information specific to the SuboptionN-code field. See Table 23 for formatting.

When used with the Broadband Forum enterprise-number for AN – vANMF discovery, the SuboptionN-data field is formatted as a series of Type/Length/Values (TLVs). The TLVs have the format shown in Table 23. This formatting is similar to that for the TLV fields for Options 17 as defined in section I.1, except that the Type and Length fields are only one octet in length.

Table 23 - Format of BBF-specific TLV fields for DHCP Option 125

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td colspan="8">Type1</td><td colspan="8">Length1</td></tr>
<tr><td colspan="16">Value1
…</td></tr>
<tr><td colspan="8">Type2</td><td colspan="8">Length2</td></tr>
<tr><td colspan="16">Value2
…</td></tr>
</table>

Field	Length	Description
TypeN	1 octet	The TypeN field identifies the type of data contained in the ValueN field.
LengthN	1 octet	The LengthN field contains the length of the ValueN field in octets.
ValueN	Variable	The Value field is zero or more octets and contains information specific to the Type. The format and length of the Value field are determined by the Type and Length fields.

Appendix II. Draft NETCONF Client and Server Connection Features

In the scope of this Technical Report, this appendix specifies the NETCONF Client and Server connection features and requirements per draft-ietf-netconf-netconf-client-server [42] but are defined as in-progress work by the owning standards bodies. The expectation is that as the work is finalized within IETF, the finalized RFC will be re-assessed for use within this Technical Report.

Table 24 summarizes the applicable NETCONF Client connection features and attributes defined in draft-ietf-netconf-netconf-client-server [42].

Table 24 - NETCONF Client Connection Features and Attributes

R-x	[Feature] OR container / leaf	Description	Comment	NETCONF Client	
				SSH connection	TLS connection
[R-161]	[Feature] Initiate	Enables the NETCONF client to initiate TCP connections		MUST	MAY
[R-162]	[Feature] ssh-initiate	The NETCONF client supports initiating SSH connections to NETCONF servers		MUST	N/A
[R-163]	[Feature] tls-initiate	The NETCONF client supports initiating TLS connections to NETCONF servers		N/A	MAY
[R-164]	netconf-client / initiate / netconf-server / endpoints	List of NETCONF servers the NETCONF client is to maintain simultaneous connections with.	An AN may need to support multiple endpoints. Plausible examples: An operator sets up a client for fault monitoring, another for statistics and a client for the EMS.	MUST	note 1
[R-165]	endpoints / endpoint/ name			MUST	note 1
[R-166]	endpoints / endpoint/ tcp-client-parameters			MUST	note 1
[R-167]	client-identity / username	NETCONF username	This attribute is used for determining privileges (see RFC 6242 [23] section 3).	MUST	N/A

R-x	[Feature] OR container / leaf	Description	Comment	NETCONF Client	
				SSH connection	TLS connection
			It is not applicable to certificate based configuration (cert-maps are contained in Table 25)		
[R-168]	client-identity / auth-type / password	Password based NETCONF client identity authentication		note 2	N/A
[R-169]	client-identity / auth-type /public-key	Public keys based NETCONF client identity authentication		note 2	N/A
[R-170]	client-identity / auth-type / hostbased	Hostbased client identity authentication		note 2	N/A
[R-171]	client-identity / auth-type / none	No NETCONF client identity authentication		note 2	N/A
[R-172]	client-identity / auth-type / certificate	Certificate based NETCONF client identity authentication		note 2	note 3
[R-173]	client-identity / auth-type / raw-public-key	Raw public keys based NETCONF client identity authentication		N/A	note 3
[R-174]	client-identity / auth-type / psk	Pre-Shared Key (PSK) based NETCONF client identity authentication		N/A	note 3
[R-175]	server-authentication	Defines the methods to authenticate the NETCONF server.		note 4	note 5
[R-176]	server-authentication / ssh-host-keys	Use of SSH host keys to authenticate the NETCONF server		note 4	N/A
[R-177]	server-authentication /	Use of Certification Authority certificates to authenticate the NETCONF server		note 4	note 5

R-x	[Feature] OR container / leaf	Description	Comment	NETCONF Client	
				SSH connection	TLS connection
	ca-certs				
[R-178]	server-authentication / server-certs	Use of server certificates to authenticate the NETCONF server		note 4	note 5
[R-179]	server-authentication / raw-public-key	Use of raw public keys to authenticate the NETCONF server		note 4	note 5
[R-180]	server-authentication / psks	Use of Pre-Shared Keys (PSKs) to authenticate the NETCONF server		note 4	note 5
[R-181]	transport-params	Configures transport characteristics		MUST	N/A
[R-182]	keep-alive	Configures session keep alive characteristics		MUST	note 1
[R-183]	hello-params	Configures the TLS version and ciphersuites	Defines the TLS version and ciphersuites.	N/A	note 1
[R-184]	connection-type / persistent-connection	Persistent connections parameters	The ability to configure whether or not the session is closed due to an inactivity timer and the keep-alive policy for keeping a session up.	MAY	note 6
[R-185]	connection-type / periodic-connection	Periodic connections parameters	The ability to configure the characteristics of that connection (idle and reconnect timers)	MAY	note 6
[R-186]	reconnect-strategy	Parameters to configure how a NETCONF client reconnects to a NETCONF server	This configuration permits a prioritization of how to reconnect when multiple endpoints are configured: which device to start with and number of attempts.	MAY	note 6

R-x	[Feature] OR container / leaf	Description	Comment	NETCONF Client	
				SSH connection	TLS connection
[R-187]	[Feature] listen	Enables NETCONF client to accept call-home connections	Indicates that the NETCONF client supports opening a port to accept NETCONF server call home connections using at least one transport (e.g., SSH, TLS, etc.)	MUST	MAY
[R-188]	[Feature] ssh-listen	The NETCONF client supports opening a port to listen for incoming NETCONF server call-home SSH connections.		MUST	N/A
[R-189]	[Feature] tls-listen	The NETCONF client supports opening a port to listen for incoming NETCONF server call-home TLS connections.		N/A	MAY
[R-190]	netconf-client / listen / idle-timeout	Idle timeout before the NETCONF session is closed	Defaults to 1 hour Option to not close the NETCONF session is also available.	MUST	note 7
[R-191]	netconf-client / listen / endpoint	List of endpoints to listen for NETCONF connections		MUST	note 7
[R-192]	netconf-client / listen / endpoint/ name			MUST	note 7
[R-193]	endpoints / tcp-server-parameters			MUST	note 7
[R-194]	client-identity / username	NETCONF username	This attribute is used for determining privileges (see RFC 6242 [23] section 3).	MUST	N/A

R-x	[Feature] OR container / leaf	Description	Comment	NETCONF Client	
				SSH connection	TLS connection
			It is not applicable to certificate based configuration (cert-maps are contained in Table 25)		
[R-195]	client-identity / auth-type / password	Password based NETCONF client identity authentication		note 2	N/A
[R-196]	client-identity / auth-type /public-key	Public keys based NETCONF client identity authentication		note 2	N/A
[R-197]	client-identity / auth-type / hostbased	Hostbased client identity authentication		note 2	N/A
[R-198]	client-identity / auth-type / none	No NETCONF client identity authentication		note 2	N/A
[R-199]	client-identity / auth-type / certificate	Certificate based NETCONF client identity authentication		note 2	note 3
[R-200]	client-identity / auth-type / raw-public-key	Raw public keys based NETCONF client identity authentication		N/A	note 3
[R-201]	client-identity / auth-type / psk	Pre-Shared Key (PSK) based NETCONF client identity authentication		N/A	note 3
[R-202]	server-authentication	Defines the methods to authenticate the NETCONF server.		note 4	note 5
[R-203]	server-authentication /	Use of SSH host keys to authenticate the NETCONF server		note 4	N/A

R-x	[Feature] OR container / leaf	Description	Comment	NETCONF Client	
				SSH connection	TLS connection
	ssh-host-keys				
[R-204]	server-authentication / ca-certs	Use of Certification Authority certificates to authenticate the NETCONF server		note 4	note 5
[R-205]	server-authentication / server-certs	Use of server certificates to authenticate the NETCONF server		note 4	note 5
[R-206]	server-authentication / raw-public-key	Use of raw public keys to authenticate the NETCONF server		note 4	note 5
[R-207]	server-authentication / psks	Use of Pre-Shared Keys (PSKs) to authenticate the NETCONF server		note 4	note 5
[R-208]	transport-params	Configures transport characteristics		MUST	N/A
[R-209]	keep-alive	Configures session keep alive characteristics		MUST	note 1
[R-210]	hello-params	Configures the TLS version and ciphersuites	Defines the TLS version and ciphersuites.	N/A	note 1

1.

Notes

1 This feature MUST be supported if tls-initiate is supported, see [R-163] otherwise it is N/A.

2 For SSH connections, it is mandatory to support at least one of the following NETCONF client identity authentication methods: password, certificate, public keys, hostbased client identity.

The option of no NETCONF client authentication MUST be supported as well but it should not be used for network deployments. It may be used for other applications like lab testing.

At minimum, Public Key encryption should be supported with NETCONF over SSH

3 If either tls-initiate, see [R-163] or tls-listen, see [R-189] are supported, it is mandatory to support at least one of the following NETCONF client identity authentication methods: certificate, raw public keys, PSKs. Otherwise this feature is N/A.

4 For SSH connections, it is mandatory to support at least one of the following NETCONF server authentication methods: SSH host keys, Certification Authority certificates, server certificates.

At minimum, Public Key encryption MUST be supported with NETCONF over SSH.

5 If either tls-initiate, see [R-163] or tls-listen, see [R-189], are supported, it is mandatory to support at least one of the following NETCONF server authentication methods: Certification Authority certificates, server certificates, raw public keys, PSKs. Otherwise this feature is N/A.

6 This feature MAY be supported if tls-initiate is supported, see [R-163] otherwise it is N/A.

7 This feature MUST be supported if tls-listen is supported, see [R-189] otherwise it is N/A.

Note: Table 24 does not report the so called "netconf-client-app-grouping" which allows unified configuration of NETCONF Client attribute that are common to "initiate" and "listen" features and to SSH and TLS types of connections

Table 25 summarizes the applicable AN NETCONF server configuration attributes defined in draft-ietf-netconf-netconf-client-server [42].

Table 25 - NETCONF Server Connection Features and Attributes

R-x	[Feature] OR container / leaf	Description	Comment	NETCONF Server	
				SSH connection	TLS connection
[R-211]	[Feature] call-home	Enables NETCONF server to initiate TCP connections		MUST	MAY
[R-212]	[Feature] ssh-call-home	The NETCONF server client supports initiating SSH connections to NETCONF clients		MUST	N/A
[R-213]	[Feature] tls-call-home	The NETCONF server supports initiating TLS connections to NETCONF clients		N/A	MAY
[R-214]	netconf-server / call-home / netconf-client / endpoints	List of NETCONF clients the NETCONF server is to maintain simultaneous call-home connections with		MUST	note 1
[R-215]	endpoints / endpoint/ name			MUST	note 1

R-x	[Feature] OR container / leaf	Description	Comment	NETCONF Server	
				SSH connection	TLS connection
[R-216]	endpoints / endpoint/ tcp-client-parameters			MUST	note 1
[R-217]	server-identity / username			MUST	N/A
[R-218]	server-identity / host-key-type / certificate			note 2	N/A
[R-219]	server-identity / host-key-type / public-key			note 2	N/A
[R-220]	server-identity / auth-type / certificate			N/A	note 3
[R-221]	server-identity / auth-type / raw-private-key			N/A	note 3
[R-222]	server-identity / auth-type /psk			N/A	note 3
[R-223]	client-authentication			note 4	note 1
[R-224]	client-authentication / supported-authentication-methods / public-key			note 4	N/A
[R-225]	client-authentication / supported-authentication-methods /			note 4	N/A

R-x	[Feature] OR container / leaf	Description	Comment	NETCONF Server	
				SSH connection	TLS connection
	passsword				
[R-226]	client-authentication / supported-authentication-methods / hostbased			note 4	N/A
[R-227]	client-authentication / supported-authentication-methods / none			note 4	N/A
[R-228]	client-authentication / users / user / public-keys			note 4	N/A
[R-229]	client-authentication / users / user / password			note 4	N/A
[R-230]	client-authentication / users / user / hostbased			note 4	N/A
[R-231]	client-authentication / users / user / none			note 4	N/A
[R-232]	client-authentication / ca-certs			MUST	note 5
[R-233]	client-authentication / client-certs			MUST	note 5

R-x	[Feature] OR container / leaf	Description	Comment	NETCONF Server	
				SSH connection	TLS connection
[R-234]	client-authentication / raw-public-key			N/A	note 5
[R-235]	client-authentication / psks			N/A	note 5
[R-236]	transport-params			MUST	N/A
[R-237]	keep-alive			MUST	note 1
[R-238]	netconf-server-parameters			MUST	note 1
[R-239]	hello-params			N/A	note 1
[R-240]	connection-type / persistent-connection			MAY	note 6
[R-241]	connection-type / periodic-connection			MAY	note 6
[R-242]	reconnect-strategy			MAY	note 6
[R-243]	[Feature] listen	Enables NETCONF server to accept connections	Indicates that the NETCONF client supports opening a port to accept NETCONF client connections using at least one transport (e.g., SSH, TLS, etc.)	MUST	MAY

R-x	[Feature] OR container / leaf	Description	Comment	NETCONF Server	
				SSH connection	TLS connection
[R-244]	[Feature] ssh-listen	The NETCONF server supports opening a port to accept NETCONF over SSH client connections.		MUST	N/A
[R-245]	[Feature] tls-listen	The NETCONF server supports opening a port to accept NETCONF over TLS client connections.		N/A	MAY
[R-246]	netconf-client / listen / idle-timeout	Idle timeout before the NETCONF session is closed	Defaults to 1 hour Option to not close the NETCONF session is also available.	MUST	note 7
[R-247]	netconf-server / listen / endpoint	List of endpoints to listen for NETCONF connections		MUST	note 7
[R-248]	netconf-server / listen / endpoint/ name			MUST	note 7
[R-249]	endpoints / tcp-server-parameters			MUST	note 7
[R-250]	server-identity / host-key-type / public-key			note 2	N/A
[R-251]	server-identity / host-key-type / certificate			note 2	N/A
[R-252]	server-identity / host-key-type /public-key			note 2	N/A
[R-253]	server-identity / auth-type / certificate			N/A	note 3
[R-254]	server-identity / auth-type / raw-			N/A	note 3

R-x	[Feature] OR container / leaf	Description	Comment	NETCONF Server	
				SSH connection	TLS connection
	private-key				
[R-255]	server-identity / auth-type /psk			N/A	note 3
[R-256]	client-authentication			note 4	note 8
[R-257]	client-authentication / supported-authentication-methods / public-key			note 4	N/A
[R-258]	client-authentication / supported-authentication-methods / passsword			note 4	N/A
[R-259]	client-authentication / supported-authentication-methods / hostbased			note 4	N/A
[R-260]	client-authentication / supported-authentication-methods / none			note 4	N/A
[R-261]	client-authentication / users / user / public-keys			note 4	N/A
[R-262]	client-authentication / users / user / password			note 4	N/A

R-x	[Feature] OR container / leaf	Description	Comment	NETCONF Server	
				SSH connection	TLS connection
[R-263]	client-authentication / users / user / hostbased			note 4	N/A
[R-264]	client-authentication / users / user / none			note 4	N/A
[R-265]	client-authentication / ca-certs			MUST	note 8
[R-266]	client-authentication / client-certs			MUST	note 8
[R-267]	transport-params			MUST	N/A
[R-268]	hello-params		Defines the TLS version and ciphersuites.	N/A	note 7
[R-269]	keep-alive			MUST	note 7
[R-270]	netconf-server-parameters			MUST	note 7

Notes:

1 This feature MUST be supported if tls-call-home is supported, see [R-213], otherwise it is N/A.

2 For SSH connections, it is mandatory to support at least one of the following NETCONF server identity host key type: certificate, public key, hostbased.

At minimum, Public Key encryption should be supported with NETCONF over SSH

3 If either tls-call-home, see [R-213], or tls-listen, see [R-245], are supported, it is mandatory to support at least one of the following NETCONF server identity authentication methods: certificate, raw private keys, PSKs. Otherwise this feature is N/A.

4 For SSH connections, it is mandatory to support at least one of the following NETCONF client authentication methods: public keys, password, hostbased.

At minimum, public key encryption MUST be supported with NETCONF over SSH.

The option of no NETCONF client authentication MUST be supported as well but it should not be used for network deployments. It may be used for other applications like lab testing.

[5] If tls-call-home, see [R-213], is supported, it is mandatory to support at least one of the following NETCONF client authentication methods: Certification Authority certificates, client certificates, raw public keys, PSKs. Otherwise this feature is N/A.

[6] This feature MAY be supported if tls-call-home is supported, see [R-213], otherwise it is N/A.

[7] This feature MUST be supported if tls-listen is supported, see [R-245], otherwise it is N/A.

[8] If tls-listen, [R-245], is supported, it is mandatory to support at least one of the following NETCONF client authentication methods: Certification Authority certificates, client certificates. Otherwise this feature is N/A.

Note: Table 25 does not report the so called “netconf-client-app-grouping” which allows unified configuration of NETCONF Server attribute that are common to “Call Home” and “listen” features and to SSH and TLS types of connections

End of Broadband Forum Technical Report TR-435