O-RAN.WG4.MP.0-v06.00

Technical Specification

O-RAN Alliance Working Group 4

Management Plane Specification

Copyright © 2021 by O-RAN ALLIANCE e.V.
By using, accessing or downloading any part of this O-RAN specification document, including by copying, saving, distributing, displaying or preparing derivatives of, you agree to be and are bound to the terms of the O-RAN Adopter License Agreement contained in the Annex ZZZ of this specification. All other rights reserved.

Buschkauler Weg 27, 53347 Alfter, Germany
Register of Associations, Bonn VR 11238
VAT ID DE321720189

Revision History

Date	Revision	Description
2019.03.11	01.00	First published version based on import of xRAN M-Plane
2019.07.03	02.00	Bug fixes and correction to v01.00 Addition of new functionality, including: - Beam tilting - Antenna calibration - CU plane monitoring - Trace - 3GPP MV PnP support - QSFP
2020.04.17	03.00	Bug fixes and correction to v02.00 - NACM table - Clarifications on CU plane monitoring - Clarification of allowed sync state transitions - Corrections on overall Start-Up operation Addition of new functionality, including: - Shared cell - Dying Gasp - PM Counters - Config Notification - Hybrid Health Warning - Dynamic Spectrum Sharing - Grouping of eAxC-IDs - Energy, Power and Environmental statistics

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Date	Revision	Description
2020.08.10	04.00	Bug fixes and correction to v03.00: - Removing reference to Component eAxC references - Correcting YANG references for Non-Delay Managed Traffic - Correcting YANG references for enhanced U-Plane markings - Correcting YANG references in tables B.3 and C.3 - Clarify that validation of configuration is based on criteria which includes definitions in this specification - Clarification of supervision Addition of new functionality, including: - Enabling static configuration of PRACH or SRS - Supporting flexible TDD pattern configuration - To allow for different delay management parameters for C and U-plane - New sync capabilities for reporting estimated time and frequency errors - New capability to define compression on an endpoint basis - New optional feature – configurable full-scale offset - New optional feature - eAxC specific gain correction - New optional feature - TX gain reference level control

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

2020.12.10	05.00	Bug fixes and correction to v04.00: - Clarify operation of default acccount for certificate access - Clarify operation of supervision in lock state - Clarify PRACH patterns - Fixing copy/paste errors in the S-plane PTP status definitions - Corrected omissions from optional feature table - Clarify center bandwidth parameter - Replace previous NMS terms with SMO - Corrections to C/U plane monitoring for FHM Addition of new functionality, including: - New NACM permissions for SMO and hybrid O-DU - New optional feature for performing pnfRegistration - New optional feature for configured YANG subscriptions sent over JSON/REST - Updating mandatory cipher to AES128-CTR - Bandwidth management to avoid over-subscription of O-RU resources - Shared cell with selective Tx/Rx using Beam ID - Cascaded FHM Operation - New capability to support co-ordinated (self) antenna calibration
2021.03.22	06.00	Bug fixes and correction to v05.00: - Clarify operation of non-persistent M-Plane - Clarify operation of Software Management - Clarify operation of VLAN-IDs for C- and U-Plane - Clarify eaxc-id assignment - Clarify connectivity checks operation - Clarify procedures for deleting configuration - Clarify plug and play certificate aspects Addition of new functionality, including: - Optional support of NETCONF/TLS - Supporting IPv6 only O-RUs

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Contents

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Chapter 1 Introductory Material

1.1 Scope

This Technical Specification has been produced by the O-RAN.org.

The contents of the present document are subject to continuing work within O-RAN WG4 and may change following formal O-RAN approval. Should the O-RAN.org modify the contents of the present document, it will be re-released by O-RAN Alliance with an identifying change of release date and an increase in version number as follows:

Release x.y.z

where:

x the first digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc. (the initial approved document will have x=01).

y the second digit is incremented when editorial only changes have been incorporated in the document.

z the third digit included only in working versions of the document indicating incremental changes during the editing process.

The present document specifies the management plane protocols used over the fronthaul interface linking the O-RU (O-RAN Radio Unit) with other management plane entities, that may include the O-DU (O-RAN Distributed Unit), the O-RAN defined Service Management and Orchestration (SMO) functionality as well as other generic Network Management Systems (NMS).

1.2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document in Release 15.

[1] 3GPP TR 21.905: “Vocabulary for 3GPP Specifications”

[2] ORAN-WG4.CUS.0-v06 “Control, User and Synchronization Plane Specification”, O-RAN Alliance, Working Group 4

[3] RFC 6241, “Network Configuration Protocol (NETCONF)”, IETF, June 2011

[4] RFC 7950, “The YANG 1.1 Data Modeling Language”, IETF, August 2016

[5] RFC 6242, “Using the NETCONF Protocol over Secure Shell (SSH)”, IETF, June 2011

[6] RFC 4252, “The Secure Shell (SSH) Authentication Protocol”, IETF, January 2006

[7] RFC 4253, “The Secure Shell (SSH) Transport Layer Protocol”, IETF, January 2006

[8] RFC 2132, “DHCP Options and BOOTP Vendor Extensions”, IETF, March 1997

[9] RFC 3925, “Vendor-Identifying Vendor Options for Dynamic Host Configuration Protocol version 4 (DHCPv4)”, IETF, October 2004

[10] RFC 2131, “Dynamic Host Configuration Protocol”, IETF, March 1997

[11] RFC 4862, “IPv6 Stateless Address Autoconfiguration”, IETF, September 2007

[12] RFC 3315, “Dynamic Host Configuration Protocol for IPv6 (DHCPv6)”, IETF, July 2003

[13] RFC 3736, “Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6”, IETF, April 2004

[14] RFC 8572, “Secure Zero Touch Provisioning (SZTP)”, IETF, April 2019

[15] RFC 8071, “NETCONF Call Home and RESTCONF Call Home”, IETF, February 2017

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

[16] SFF-8472v11, “Diagnostic Monitoring Interface for Optical Transceivers”, SFF Committee, September 2010

[17] IEEE 802.1ag, “IEEE Standard for Local and Metropolitan Area Networks Virtual Bridged Local Area Networks Amendment 5: Connectivity Fault Management”, IEEE, 2007

[18] RFC 862, “Echo Protocol”, IETF, May 1983

[19] MEF.38, “Service OAM Fault management YANG Modules”, Metro Ethernet Forum, April 2012

[20] RFC 7895, “YANG Model Library”, IETF, June 2016

[21] RFC 5277, “NETCONF Event Notifications”, IETF, July 2008

[22] G.8275.1, “Precision time protocol telecom profile for phase/time synchronization with full timing support from the network”, ITU, June 2016

[23] G.810, “Definitions and terminology for synchronization networks”, ITU, August 1996

[24] 1588v2-2008, “IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems”, IEEE, 2008

[25] Y.1731, “Operation, administration and maintenance (OAM) functions and mechanisms for Ethernet based networks”, ITU, August 2015

[26] AISG 2.0, “Control interface for antenna line devices”, Antenna Interface Standards Group, June 2006

[27] 3GPP 25.462, “UTRAN Iuant interface: Signalling transport”, 3GPP

[28] 3GPP 25.466, “UTRAN Iuant interface: Application part”, 3GPP

[29] 3GPP 25.463, “UTRAN Iuant interface: Remote Electrical Tilting (RET) antennas Application Part (RETAP) signalling”, 3GPP

[30] ITU X.733, “Information Technology – Open Systems Interconnection - System Management: Alarm Reporting Function”, 1992

[31] RFC 6187, “X.509v3 Certificates for Secure Shell Authentication”, IETF, March 2011

[32] 3GPP TS 36.213, “Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures”, 3GPP, V13.6.0 (2017-06)

[33] RFC 4361, Node-specific Client Identifiers for Dynamic Host Configuration Protocol Version Four (DHCPv4), IETF, February 2006

[34] SFF-8636v2.9.3, “Specification for Management Interface for Cabled Environment”, SFF Committee, April 2019

[35] RFC 6470, “Network Configuration Protocol (NETCONF) Base Notifications”, IETF, February 2012

[36] VES Event Listener 7.2 https://docs.onap.org/projects/onap-vnfrqts-requirements/en/latest/Chapter8/ves_7_2/ves_event_listener_7_2.html

[37] RFC 8639, “Subscription to YANG Notifications”, IETF, September 2019

[38] RFC 7951, “JSON Encoding of Data Modeled with YANG”, IETF, August 2016

[39] RFC 5246, "The Transport Layer Security (TLS) Protocol Version 1.2", IETF, August 2008

[40] RFC 6125, “Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS) “, IETF, March 2011

[41] RFC 7589, “Using the NETCONF Protocol over Transport Layer Security (TLS) with Mutual X.509 Authentication”, IETF, June 2015[41] RFC 7540, “Hypertext Transfer Protocol Version 2 (HTTP/2)”, IETF, May 2015

[42] RFC 8446, “The Transport Layer Security (TLS) Protocol Version 1.3”, IETF, August 2018

[43] RFC 7030, “Enrollment over Secure Transport”, IETF, October 2013

[44] RFC 4210: "Internet X.509 Public Key Infrastructure Certificate Management Protocol".

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

[45] Transport Layer Security (TLS) Parameters. https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml, Internet Assigned Numbers Authority, (IANA), January 27, 2021.

[46] 3GPP TS 33.210, "Network Domain Security (NDS); IP network layer security", Release 16, 3GPP, July 2020.

[47] RFC 5289, "TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM)", IETF, August 2008.

[48] RFC 5288, "AES Galois Counter Mode (GCM) Cipher Suites for TLS", August 2008.

[49] RFC 7540, "Hypertext Transfer Protocol Version 2 (HTTP/2", May 2015

1.3 Definitions and Abbreviations

1.3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [1].

Antenna Line: connection between O-RU and antenna

C-Plane: Control Plane: refers specifically to real-time control between O-DU and O-RU, and should not be confused with the UE's control plane

Cascade mode: Mode of Shared cell which is realized by several O-RUs cascaded in chain topology

DL: DownLink: data flow towards the radiating antenna (generally on the LLS interface)

eAxC: extended Antenna-Carrier: a data flow for a single antenna (or spatial stream) for a single carrier in a single sector.

Event-Collector: A REST server to which an O-RU supporting NON-PERSISTENT-NETCONF feature can send a JSON notification

FHM mode: Mode of Shared cell which is realized by FHM and several O-RUs in star topology.

LLS: Lower Layer Split: logical interface between O-DU and O-RU when using a lower layer (intra-PHY based) functional split.

LLS-U: Lower Layer Split User-plane: logical interface between O-DU and O-RU when using a lower layer functional split.

LLS-C: Lower Layer Split Control-plane: logical interface between O-DU and O-RU when using a lower layer functional split.

LLS-S: Lower Layer Split Synchronization-plane: logical interface between O-DU and O-RU when using a lower layer functional split.

High-PHY: those portions of the PHY processing on the O-DU side of the fronthaul interface, including FEC encode/decode, scrambling, and modulation/demodulation.

Low-PHY: those portions of the PHY processing on the O-RU side of the fronthaul interface, including FFT/iFFT, digital beamforming, and PRACH extraction and filtering.

M-Plane: Management Plane: refers to non-real-time management operations between the O-DU and the O-RU

North-node: the O-DU or a connected O-RU closer to the O-DU for the O-RU, e.g., the cascade O-RU#1 connected to O-RU#2 is north-node for O-RU#2, when O-DU, O-RU#1 and O-RU#2 are in cascade chain topology. The O-DU in star topology connected to an FHM is north-node for the FHM.

NMS: A Network Management System dedicated to O-RU operations

Port: End of a transport link – in most cases this is an optical port

Port Number: A number which identifies a port (see Port). In case of SFP/SFP+ port, port number value is 0 to N-1 where N is number of ports in the device. Numbers 0 to N-1 are assigned to ports in order following order of labels on the device (labels for ports are not necessarily numbers starting from zero)

O-DU: O-RAN Distributed Unit: a logical node hosting PDCP/RLC/MAC/High-PHY layers based on a lower layer functional split.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

O-RU: O-RAN Radio Unit: a logical node hosting Low-PHY layer and RF processing based on a lower layer functional split. This is similar to 3GPP's "TRP" or "RRH" but more specific in including the Low-PHY layer (FFT/iFFT, PRACH extraction).

O-RU Controller: A network function that is permitted to control the configuration of an O-RU. Examples of O-RU controllers include, an O-DU, a classical NMS, an O-RAN Service Management and Orchestration function, or other network automation platforms.

S-Plane: Synchronization Plane: refers to traffic between the O-RU or O-DU to a synchronization controller which is generally an IEEE-1588 Grand Master (however, Grand Master functionality may be embedded in the O-DU).

Shared cell: The operation for the same cell by several O-RUs.

Shared cell network: the network for several cascade O-RUs in a chain topology or the network for one FHM and several O-RUs in a star topology.

South-node: a connected O-RU far from O-DU for the O-RU, e.g., the cascade O-RU#2 connected to O-RU#1 is south-node for O-RU#1, when O-DU, O-RU#1 and O-RU#2 are in cascade chain topology. The O-RU in star topology connected to an FHM is south-node for the FHM.

Spatial stream: the data flow on the DL associated with precoded data (may be same as layers or different if there is expansion in the precoding), and on UL associated with the number of outputs from the digital beamforming (sometimes called "beams").

SSM: Synchronization Status Message: part of ITU G.781 and G.8264 standards.

TRX: Refers to the specific processing chain in an O-RU associated with D/A or A/D converters. Due to digital beamforming the number of TRXs may exceed the number of spatial streams, and due to analog beamforming the number of TRXs may be lower than the number of antenna elements.

U-Plane: User Plane: refers to IQ sample data transferred between O-DU and O-RU

UL: Up-Link: data flow away from the radiating antenna (generally on the LLS interface)

Virtual Connection: a connection between O-RU and O-RU controller. This connection is established by means of autodetection procedure and is supervised by supervision procedure.

1.3.2 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

ALD	Antenna Line Device
AVP	Average Power
BCN	BTS Clock Number
CRC	Cyclic Redundancy Check
CUS	Control/User/Synchronization
DHCP	Dynamic Host Configuration Protocol
DMTC	DRS Measurement Timing Configuration
DRS	Discovery Reference Signal
DSCP	Differentiated Services Code Point
FHM	Fronthaul Multiplexer
HDLC	High-Level Data Link Control
lls-M	Lower Layer Split Management plane
LAA	Licensed Assisted Access
LBM	Loop-Back Message
LBR	Loop Back Reply
LBT	Listen Before Talk

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

ME	Maintenance Entity
MEP	Maintenance association End Point
NAT	Network Address Translation
NDM	Non-Delay Managed
NETCONF	Network Configuration Protocol
O-DU	O-RAN Distributed Unit (see definitions section)
O-RU	O-RAN Radio Unit
OMA	Optical Modulation Amplitude
PDV	Packet Delay Variation
PNF	Physical Network Function
QoS	Quality of Service
RET	Remote Electrical Tilt
RPC	Remote Procedure Call
SFP	Small Form-factor Pluggable
sFTP	Secure File Transfer Protocol or SSH File Transfer Protocol
SLAAC	Stateless Address Auto Configuration
SMO	Service Management and Orchestration
SRS	Sounding Reference Signal
SSH	Secure Shell
TLS	Transport Layer Security
T-TSC	Telecom Time Subordinate Clock. This is what ITU-T standards refer to as a Telecom Time Slave Clock
VLAN	Virtual LAN
YANG	Yet Another Next Generation

1.4 Conventions

This management plane specification includes cross references to a set of associated YANG models. Text may reference particular YANG leafs, notifications and remote procedure calls (RPCs). In order to assist in readability, all cross references to YANG defined elements will keep the identical case format as defined in the corresponding YANG model, with the font-weight set to **bold**. This convention applies only to text and not to YANG elements embedded into figures.

If there is any conflict between the YANG models and the accompanying text description in this specification, the definition of the YANG models shall take precedence.

1.5 Topics for Future Specification Versions

Extensions to this version of the O-RAN WG4 Management Plane specification together with corrected errors will be included in the future versions of this document.

The following topics are to be considered for future versions of the specification:

1. Beam Id field interpretation for various types of beamforming
2. Redundancy and failover scenario
3. Shared cell support for IP-defined flows
4. Enhancements to better align with O-RAN Alliance O1 specification
5. The introduction of file transfer using FTPs

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

The revision statement in the YANG models will be used to describe future revisions to the models that are backwards compatible. Backwards incompatible changes will be addressed by incrementing the number used as part of the model name and namespace, effectively creating a new YANG model. The format of the namespace used in all O-RAN YANG models is "urn:o-ran:"<model-name>":"<model-number>, where the initial <model-number> used in a newly defined YANG model is "1.0". Where this document makes reference to models, irrespective of their backward compatibility, a generic <model-number> of "x.y" is used to enable reference to all versions of the namespace for a particular <model-name>.

The revision statement in all YANG models includes a reference statement used to cross-reference to the first version of this document where the corresponding description was introduced. For example, the reference in all revision statements for the initial O-RAN models include cross-reference to "ORAN-WG4.MP.0-v01.00".

The revision statement of the YANG models also includes a description which is used to track the versioning of the YANG model. All revision statement descriptions will begin with "version "<a>"."<b>"."<c>, where <a>, <b> and <c> are used to reflect the version of the YANG model, where

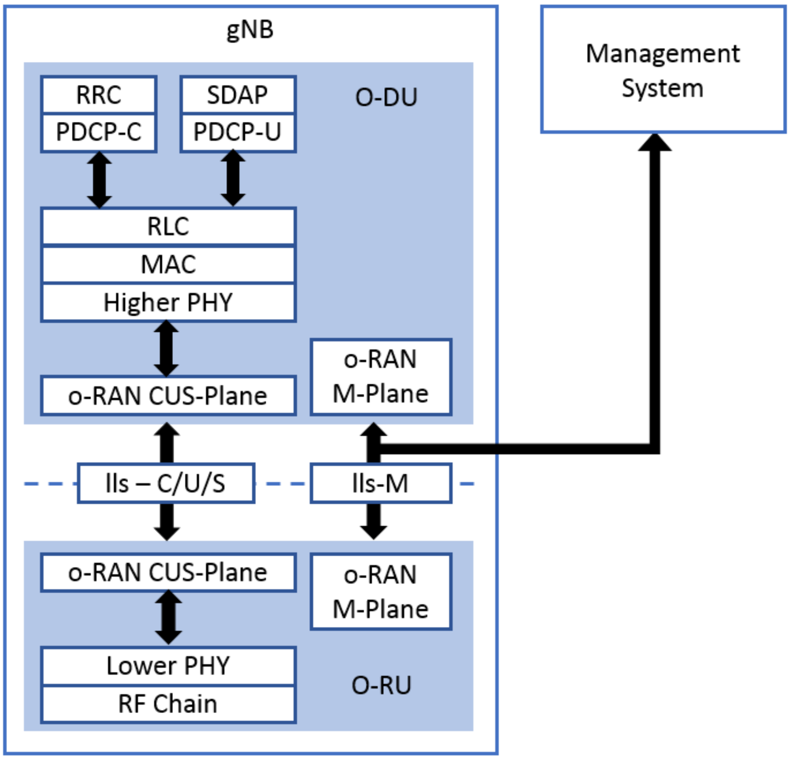
- <a> corresponds to the first digit of the O-RAN WG4 management plane specification version where the corresponding description was first introduced, corresponding to <x> in sub-section 1.1
- <b> is incremented when errors in the YANG model have been corrected
- <c> is incremented only in working versions of the YANG model indicating incremental changes during the editing process

Chapter 2 High Level Description

2.1 Top level functional description, terminology, including hybrid, hierarchical

2.1.1 Architecture for O-RAN WG4 Fronthaul functional split

This O-RAN FH specification addresses the lower layer functional split as depicted in Figure 1. Refer to the O-RAN CUS plane specification [2] for more details on the split architecture. The Lower-Layer Split M-plane (LLS-M) facilitates the initialization, configuration and management of the O-RU to support the stated functional split.



Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Figure 1 – O-RAN WG4 FH functional split

2.1.2 M-Plane architecture model

A NETCONF/YANG based M-Plane is used for supporting the management features including "start up" installation, software management, configuration management, performance management, fault management and file management towards the O-RU. The M-Plane supports two architectural models:

1. **Hierarchical model**. As shown on the left side Figure 1, the O-RU is managed entirely by one or more O-DU(s) using a NETCONF based M-Plane interface. When the O-RU is managed by multiple O-DUs, it is typically for enabling O-DU and/or transport connectivity redundancy capabilities. Refer to Chapter 3 for more details.

2. **Hybrid model**. As shown on the right side of Figure 1, the hybrid architecture enables one or more direct logical interface(s) between management system(s) and O-RU in addition to a logical interface between O-DU and the O-RU. It should be noted that the NETCONF clients connecting to the O-RU may be of different classes (e.g. O-DU and SMO). For example, functions like O-RU software management, performance management, configuration management and fault management can be managed directly by the management system(s).

In the hybrid model, the O-RU has end to end IP layer connectivity with the SMO. From a physical network point of view, this connectivity could be via the O-DU, where the O-DU is acting as an IP/Ethernet packet forwarder, forwards the packets between O-RU and the SMO. Direct logical communication between an O-RU and SMO can be enabled via O-RUs being assigned routable IPs or local private IPs resolved by a NAT function in the network (or implemented at the O-DU). Refer to chapter 3 for details how O-RU acquires the IP address of O-DU and SMO for the M-plane communication.

As described in Chapter 3, there is no explicit signaling to indicate that an O-RU is operating in a hierarchical or hybrid configuration. All NETCONF servers supporting this M-Plane specification shall support multiple NETCONF sessions, and hence all complaint O-RUs shall be able to support both hierarchical and hybrid deployment.

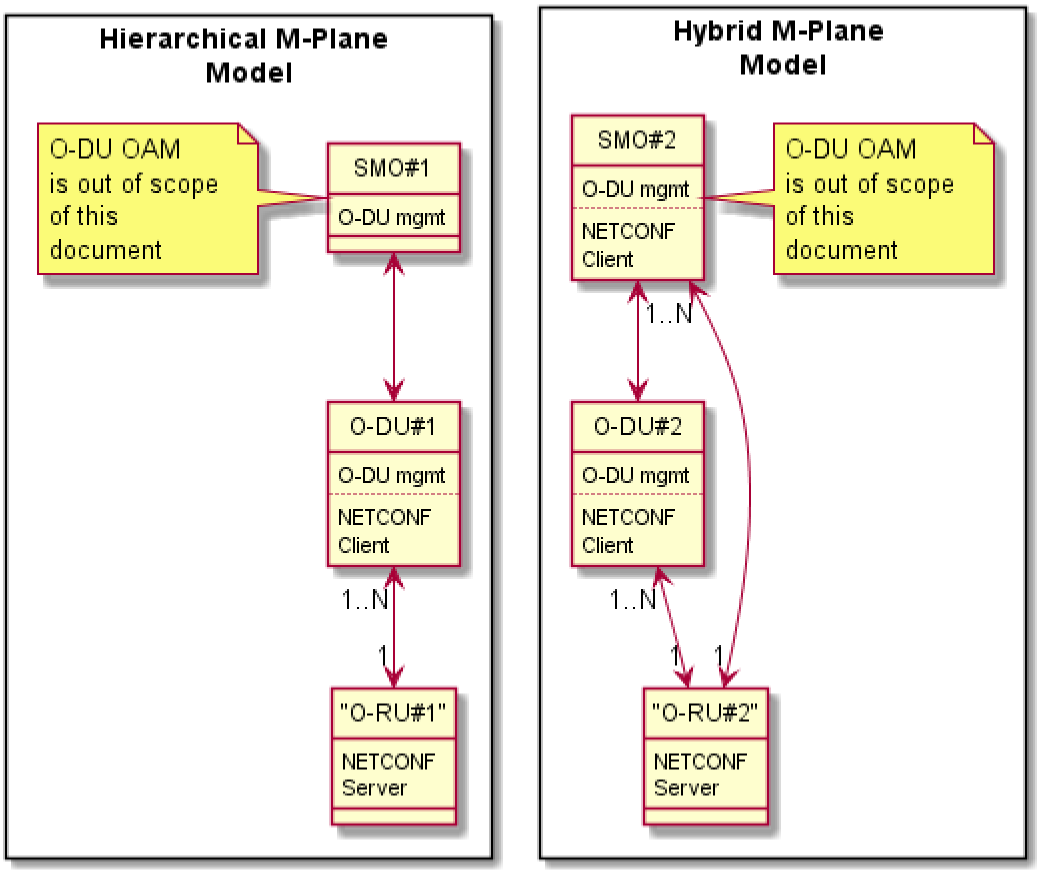


Figure 2 - M-Plane Architecture

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

NETCONF/YANG is used as the network element management protocol [3] and data modeling language [4]. Use of such a standardized framework and common modeling language simplifies integration between O-DU and O-RU as well as operator network integration (in terms of running service) in case of elements sharing a common set of capabilities. The framework supports integration of products with differing capabilities enabled by well-defined published data models. NETCONF also natively supports a hybrid architecture which enables multiple clients to subscribe and receive information originating at the NETCONF server in the O-RU.

2.1.3 Transport Network

Based on the transport topology, various modes of network connectivity are possible between O-RU and O-DU and SMO.

The basic requirement for M-Plane is to have end to end IP connectivity between the O-RU and the elements managing it (O-DU, SMO, or so called “O-RU Controllers”). The connectivity between the O-DU and SMO and its management plane are not in scope of this specification. The O-RU shall support either IPv4 or IPv6 and optionally support dual stack (IPv4 and IPv6).

> Note, in previous versions of this specification, only IPv4 was mandatory. In order to ensure backwards compatibility with equipment supporting earlier versions of this specification, an operator and vendor can agree to use a common IP version in the O-RU, O-DU and any other O-RU controllers.

2.1.4 M-Plane functional description

The M-Plane provides the following major functionalities to the O-RU. These features are implemented using the NETCONF provided functions.

“Start up” installation

During startup, the O-RU acquires its network layer parameters either via static (pre-configured in the O-RU) or dynamically via DHCP or DHCPv6. During this process the O-RU may acquire the IP address of the O-RU controller(s), in which case the O-RU establishes the NETCONF connectivity using the “call home” feature. When the O-RU is operating in an environment which include the O-RAN defined SMO, the O-RU may acquire the IP address of the event-collector(s), in which case the O-RU performs a pnfRegistration which triggers the SMO to establish NETCONF connectivity using the information recovered from the pnfRegistration procedure. The capability exchange is performed between the client and server as part of the initial NETCONF Hello exchanges. Details of these steps are provided in chapter 3.

Note, the use of “start up” terminology in this specification is distinct from the “startup” capability used in a NETCONF environment to indicate that a device supports separate running and startup configuration datastores. This specification makes specific reference to configuration which is required to be stored in “reset persistent memory”. The O-RU shall use this stored configuration as its “startup” configuration.

SW management

The M-Plane is responsible for software download, installation, validation and activation of new SW when requested by O-RU Controller. The software download is triggered by NETCONF RPC procedures, and the actual software package download is done using sFTP. In this version of the M-Plane specification, sFTP is the only defined file transfer protocol for software and file management.

Configuration management

Configuration management covers various scenarios like Retrieve Resource State, Modify Resource State, Modify Parameters and Retrieve Parameters. NETCONF **get-config** and **edit-config** RPCs shall be used for configuration parameter retrieval and updates at the O-RU

Performance management

Performance management describes the measurements and counters used to collect data related to O-RU operations. The purpose of Performance Management is optimizing the operation of the O-RU.

The measurement results are reported by two options:

1. **YANG Notification**: This option uses the stats definition of YANG model per measurement group. In this case, **get** rpc and/or notification will be used (see Chapter 7 for more details).

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

2. **File Upload:** This option uses the file upload procedure defined in File management. The measurement results are saved to a data file periodically.

Fault Management

Fault management is responsible for sending alarm notifications to the NETCONF Client. Fault Management allows alarm notifications to be disabled or enabled as well as alarm subscription.

File Management

File management allows the O-RU Controller to trigger an O-RU to perform upload of files stored on O-RU to O-RU Controller. The O-RU may provide different kinds of files and retrieved files can be used for various purposes. Simultaneous multiple file upload operations can be supported under the same sFTP connection between O-RU to O-DU/SMO.

2.2 Interfaces

The M-Plane interface is defined between the O-RU Controller and the O-RU. The protocol stack of the M-Plane interface is shown in Figure 3. The transport network layer is built on IP transport and SSH/TCP, and optionally TLS, is used to carry the M-Plane message between the O-RU Controller and the O-RU. As an option, the O-RU may support the capability to support asynchronous notifications to be sent using HTTPS. This option enables system optimization when the O-RU Controller corresponds to the SMO which is operating with a non-persistent NETCONF session to the O-RU.

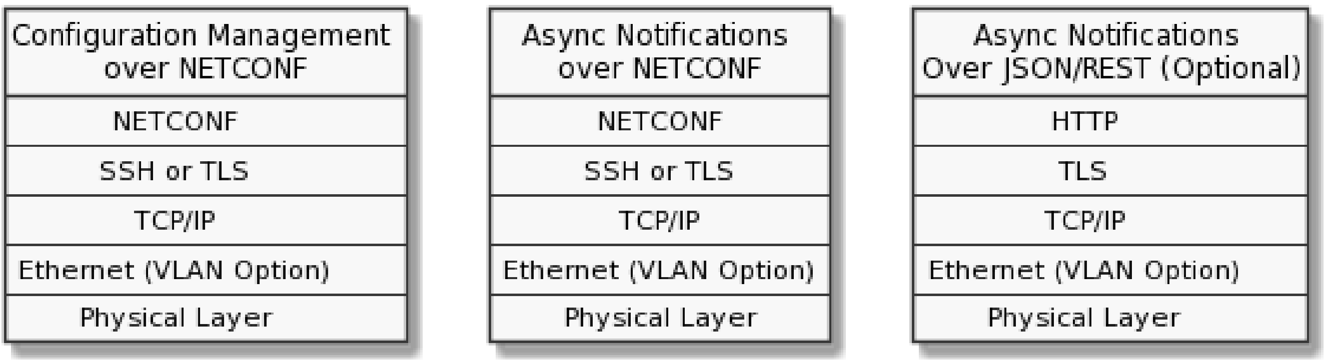


Figure 3: M-plane protocol stack

2.3 YANG Module Introduction

The data models representing the M-Plane are organized as a set of reusable YANG modules. It is also the intent to reuse the publicly available and generic YANG models as much as possible instead of developing customized O-RAN specific modules. Refer to the various chapters, Annex D and the repository of YANG models for more details on each of these modules.

2.4 Security

The M-Plane provides end to end security as a mandatory feature, see **Table 1**. The M-Plane security shall be supported using the SSHv2 layer in accordance with RFC 6242 [5]. TLS 1.2 in accordance with RFC 7589 [41] may be optionally supported. TLS 1.3 in accordance with RFC 8446 [42] may also be optionally supported in addition to TLS 1.2. RFC 6242 [5] and RFC 7589 [41] provide the procedures for interoperability with NETCONF implementations. If there are multiple NETCONF sessions established with a single O-RU, either SSH tunnels or TLS connections may be used and each session should be established over a separate SSH tunnel or TLS connection. For the O-DU, the operator may use SSH, TLS, or both. It is recommended that operators use TLS in production networks.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Plane	Integrity (protection from modifications)	Confidentiality (encryption protection)	Authentication (validity of the originator)	Remarks
M-Plane/ NETCONF	Yes	Yes	Yes	NETCONF transport: a) Mandatory support for SSHv2, RFC 6242 [5] b) Optional support for TLS 1.2, RFC 7589 [41]
Optional support of JSON/REST	Yes	Yes	Yes	HTTPS used for JSON/REST transport

Table 1- M-Plane Security

SSHv2 may be used to perform SSH server host authentication, key exchange, encryption, and integrity protection. It also derives a unique session ID that may be used by higher-level protocols. The end point (SSH client) authentication should be done as per RFC 4252 [6]. Chapter 3 describes the authentication approach based on username and password as well as based on X.509 certificates.

The SSHv2 transport level security (encryption algorithms, data integrity algorithms) shall be based on RFC4253 [7]. As per aes128-ctr shall be the mandatory ciphering protocol, and rest of the ones listed as optional. For data integrity, hmac-sha2-256 shall be the mandatory algorithm, and rest of the listed algorithms shall be optional. Public key-based host authentication shall be used for authenticating the server (RFC 4253) by the clients, and username/password based client authentication shall be done by the server as part of the SSH session establishment. The O-RU shall support the host key algorithms and key exchange methods defined in section 10.1 of RFC 5656 for securing the Secure Shell (SSH) transport.

> Note, in order to ensure backwards compatibility with equipment supporting earlier versions of this specification, an operator and vendor may agree to use one of the optional ciphering protocols.

As an additional option, both client and server may implement authentication based on X.509 certificates. With this option, RSA 2048 bit shall be supported for the Public Key algorithm, aes128-ctr shall be supported for the cyphering algorithm and hmac-sha2-256 shall be supported for integrity algorithm.

> Note, the above specification will be replaced with a cross reference to the O-RAN Security Task Group Guidelines once such is published.

TLS 1.2 based on RFC 5246[38] performs mutual authentication, key exchange, encryption, and integrity protection to ensure trusted communication between the NETCONF server (O-RU) and the NETCONF client (O-DU or SMO). NETCONF implementations may support X.509 certificate based authentication using TLS 1.2 based on RFC 7589 [41]. When X.509 based authentication is used, NETCONF server identity is based on RFC 6125 [40] and NETCONF client identity is specified in RFC 7589[41].

TLS 1.2 implementations shall support the following TLS Cipher Suites with SHA-256 and AES Galois Counter Mode in accordance with RFC 7540 [42] and 3GPP TS 33.210 [46]:

- ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289 [47].
- DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288 [48].

It is mandatory that TLS implementations follow the rules on allowed cipher suites given in TLS 1.2 (RFC 5246). Implementations may include additional TLS cipher suites that provide mutual authentication and confidentiality as required by NETCONF in RFC 6241 [3]. Only cipher suites with AEAD (e.g. GCM) and PFS (e.g. ECDHE, DHE) and recommended by IANA [45] may be optionally supported. . The disallowed cipher suites in RFC 7540 [49], Appendix A, shall not be used. The vendor and operator need to be prepared to replace integrity and/or ciphering algorithms if the current algorithm in use is compromised or deprecated. TLS 1.2 shall follow TLS profiling defined in 3GPP TS 33.210 [46] section 6.2.3.

Operators may select the authentication mechanism and protocol to use as shown in Table 2.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Protocol	PKIX (Public Key Infrastructure with X.509 Certificates)	Simple Public Key	Password-based Authentication
TLS 1.2	Optional to support / Optional to use	Not specified in RFCs 5246/8446	Not specified for use with NETCONF
SSHv2	Optional to support/Optional to use	Used for SSH Server authentication by SSH client. Mandatory to support / Optional to use	Used for SSH Client authentication by SSH server. Mandatory to support / Optional to use

Table 2: Mandatory and Optional Features for O-RU Authentication

Chapter 3 “Start up” installation

This chapter provides the overall start-up mechanism from the power-on of O-RU to available in service.

Pre-condition:

- Power-ON for O-RU/NETCONF Server or O-RU restart operation.

  Power-ON for O-RU controller/NETCONF Client(s) and/or pnfRegistration event-collector.

- Physical interface(s) is(are) connected.

Post-condition:

- O-RU is ready for the radio transmission to the air on at least one carrier if packet transmission received from O-DU
- O-RU is ready for the packet transmission to the O-DU if radio reception received at the air on at least one carrier.
- At least one O-RU Controller/NETCONF client with either “super-user” or “hybrid-odu” access privileges can control the carrier configuration of the O-RU/NETCONF server in O-RU.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

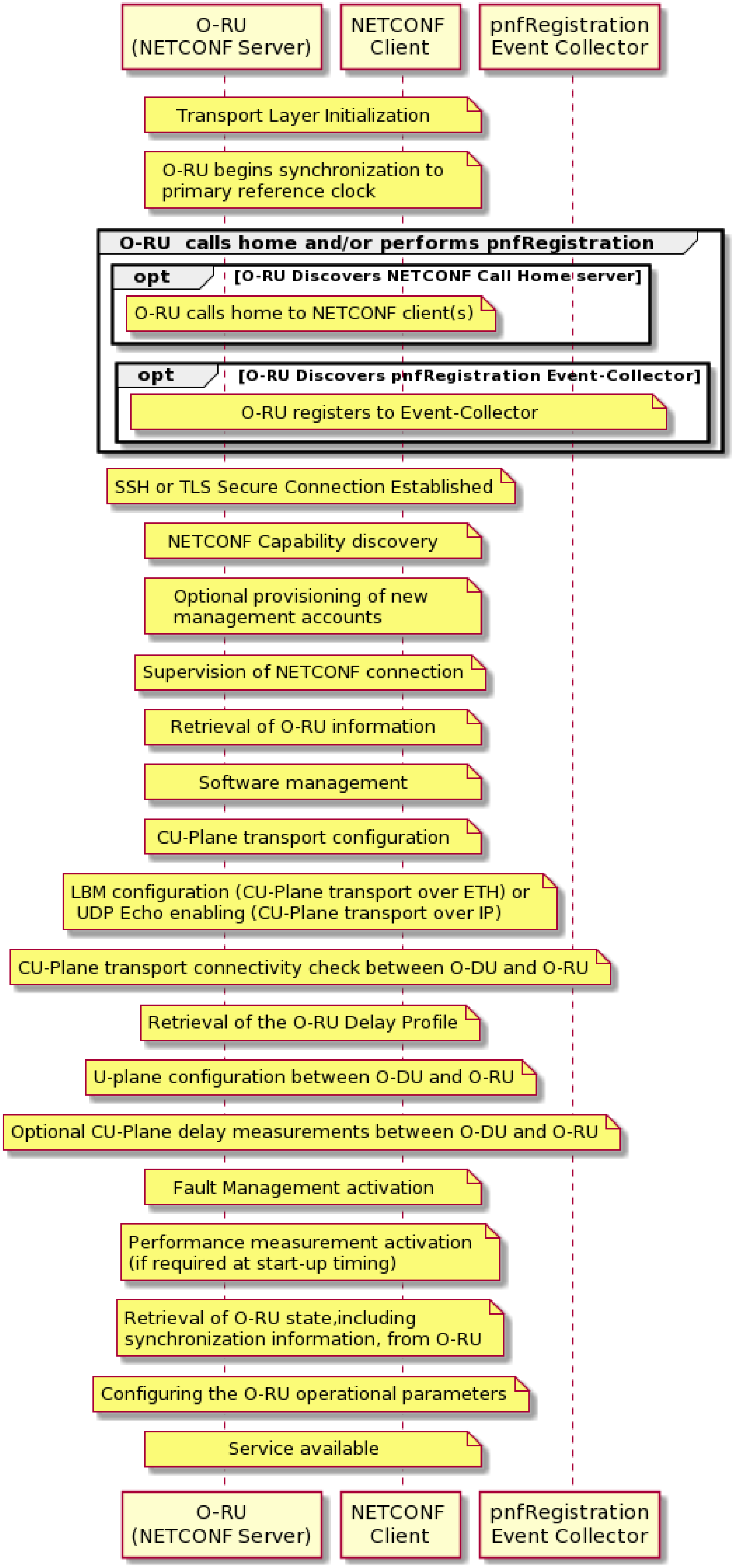


Figure 4: Overall of Start-Up Installation

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

At the power-on of O-RU or following an O-RU restart, the following procedures are performed.

1. O-RU performs M-Plane transport layer resolution (DHCP, MAC, VLAN, IP, etc.) and recovers IP address(es) of O-RU controller(s) and/or pnfRegistration event-collector.

2. O-RU begins synchronization of the O-RU against a Primary Reference Clock. (Note: step 2 may be in parallel with step 1 for some O-RU implementation.)

3. O-RU performs NETCONF Call Home to discovered O-RU controller(s) and/or performs pnfRegistration to discovered event-collector.

4. O-RU controller performs SSH or TLS connection establishment.

5. O-RU and O-RU controller perform NETCONF capability discovery.

6. O-RU controller performs optional provisioning of new management accounts (typically only performed once during pre-staging)

7. O-RU and O-RU controller perform supervision of NETCONF connection.

8. O-RU controller performs retrieval of O-RU information.

9. O-RU controller performs SW management.

10. O-DU performs CU-Plane transport configuration

11. (opt) O-DU performs LBM configuration (CU-Plane over ETH) or enables UDP Echo (CU-Plane over IP).

12. (opt) O-DU performs initial C/U-Plane transport connectivity checking between O-DU and O-RU.

13. O-RU controller retrieves the O-RU delay profile from the O-RU.

14. O-RU controller performs U-Plane configuration between O-DU and O-RU. C/U-Plane transport connectivity between O-DU and O-RU is configured as part of this step.

15. O-DU optionally performs C/U-Plane delay measurements between O-DU and O-RU if the O-RU supports it.

16. O-RU controller performs Fault Management activation.

17. O-RU controller activates performance measurement (if required at start-up timing).

18. O-RU controller retrieves O-RU state, including synchronization information, from O-RU.

19. O-RU controller configures the O-RU operational parameters.

20. Service available.

Note, the synchronization procedures started in step 2 needs to be completed before service is available.

Note: periodic CU-Plane connectivity check is not considered as the part of start-up. Once configured in start-up phase, CU-Plane connectivity check can later be performed periodically and at any time in run-time.

This chapter mainly covers 1 and 3 to 7 as sub-sections.

Cross Reference of other chapters:

The detail of 2. Synchronization management is described in Chapter 10.

The method of 8 and 17 retrieval of O-RU information is described in Chapter 6.

The detail of 9. SW management is described in Chapter 5.

The detail of 12. C/U-Plane transport connectivity checking between O-DU and O-RU is described in Chapter 4.

The detail of 13. Retrieval of the O-RU delay profile and 13. C/U-Plane delay measurements are described in Chapter 4.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

The detail of 14. U-plane configuration is described in Chapter 12, and C/U-Plane transportation configuration is described in Chapter 4.

The detail of 17. Performance management is described in Chapter 7.

The detail of 16. Fault management is described in Chapter 8.

The method of 19. Control to make service available is described in Chapter 12.

3.1 Management Plane Transport aspects

This sub-section provides the M-plane transport establishment scenario between O-RU and O-RU controller(s), such as O-DU and/or SMO. The transport layer address of M-plane is only the target in this section. Transport aspects of the C plane and U plane are covered in Chapter 4.

Pre-condition:

- Physical interface is connected.

- When operating in an environment using call-home, the NETCONF server and NETCONF Client(s) have an identical NETCONF call home port configured, to ensure the NETCONF client listens on the same port used by the NETCONF Server.

Post-condition:

- Transport Layer address(es) for M-plane are known to O-RU and O-RU controllers.

- O-RU is aware of the physical port(s) for M-plane, e.g., if there are multiple ports in the O-RU.

- O-RU is aware of the VLAN(s) to be used for M-Plane, e.g., if VLANs are used in the transport network.

- Then O-RU is ready to establish TCP connection for NETCONF call home and/or for PNF registration.

For the transport establishment, there are the following alternatives:

a) Manual transport layer address configuration in O-RU. This configuration contains the addresses for O-RU and NETCONF client(s) and/or the event-collector. The method to manually configure the O-RU is out of scope in this specification. Assuming manual configuration is successful, the NETCONF server shall be able to recover this configured information and use the o-ran-mplane-int.yang model to communicate this operational-state to a NETCONF client.

b) If IPv4 is supported, DHCP server provides O-RU's transport layer address information together with the identity of the NETCONF client and/or the identity of the event-collector. This identity encodes either the transport layer address or FQDN of the NETCONF client or event-collector. If an FQDN is signaled, the O-RU shall use the DNS server address provided by the DHCP server to recover the IP address corresponding to FQDN of the NETCONF client or event-collector.

c) If IPv6 is supported, Stateless Address Auto-Configuration (SLAAC) is used to configure the O-RU's transport address with the DHCPv6 server providing the identity of the NETCONF client and/or event-collector. This identity encodes either the transport layer address or FQDN of the NETCONF client or event-collector. If an FQDN is signaled, the O-RU shall use the DNS server address provided by the DHCPv6 server to recover the IP address corresponding to FQDN of the NETCONF client or event-collector.

Note, a NETCONF client can receive a hint as to whether an O-RU supports a particular IP version by using the **get** rpc to recover the list of **interfaces** supported by the O-RU and using the presence of the augmented **ipv4** container or **ipv6** container in the o-ran-interfaces module as an indication that a particular IP version is supported.

The O-RU uses the o-ran-dhcp.yang model to be able to expose information signaled by the DHCP server.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

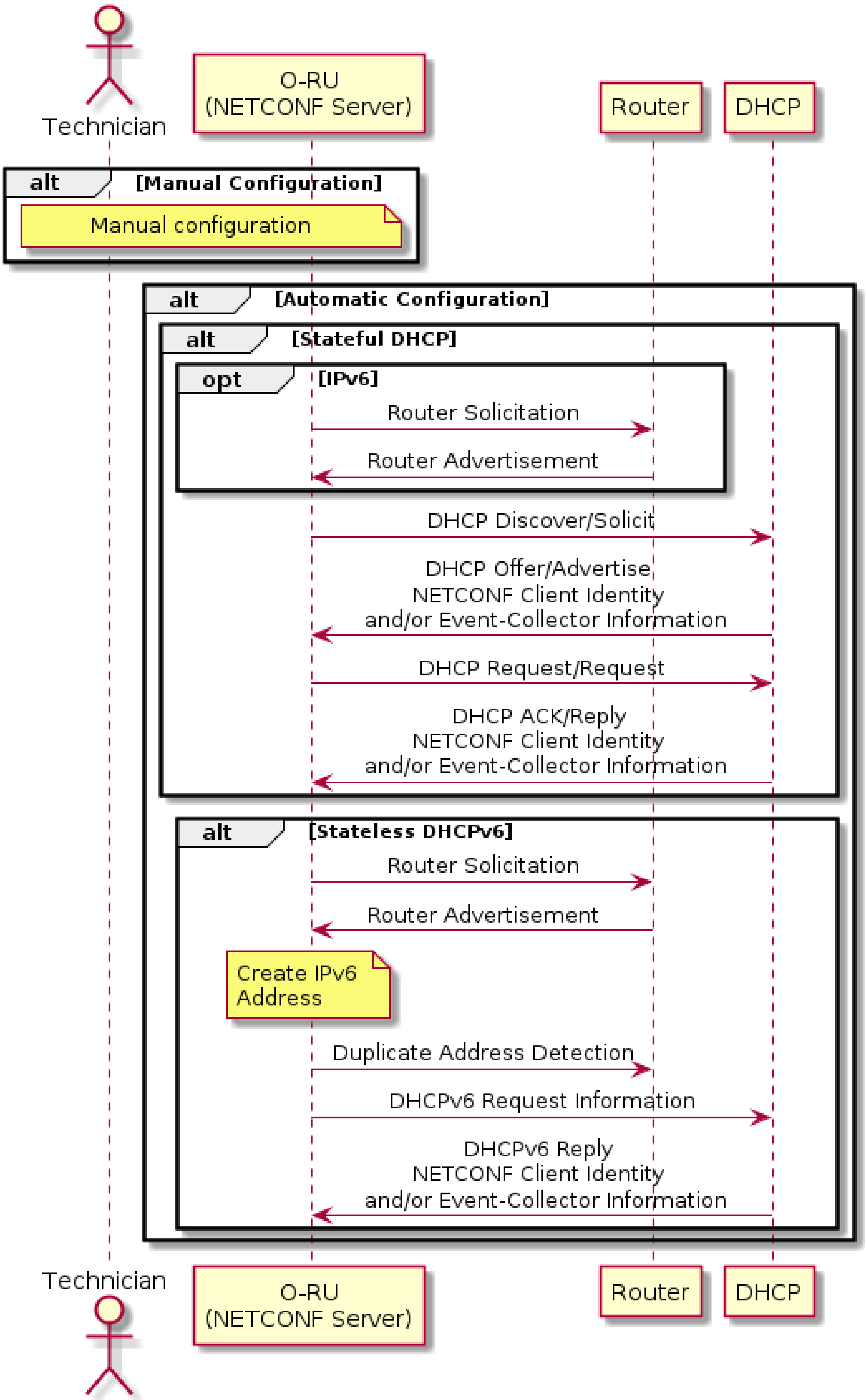


Figure 5: Transport Layer Establishment for M-plane

Transport Layer interface related information for M-plane contains at least the physical port number, the hardware address of the Ethernet port, VLAN-ID, local IP address, remote IP address, Default Gateway address and Subnet mask.

In the case of option b) and c), the following subsections are used:

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

- O-RU identification in DHCP messages from O-RU.
- VLAN discovery aspect for M-plane.
- IP address assignment to O-RU.
- Discovery of address information of O-RU controller(s) and/or Event-Collector.

3.1.1 O-RU identification in DHCP

The O-RU shall identify itself to DHCP servers by using DHCP option(s) using the **vendor-class-data** string within the o-ran-dhcp YANG model. If the O-RU supports IPv4, there are two alternatives. One uses option 60 Vendor Class Identifier, RFC2132 [8]. The other uses option 124 Vendor Identifying Vendor Class Option, RFC3925 [9]. An O-RU implementing IPv4 shall support at least one of these options. If the O-RU supports IPv6, then it shall identify itself using the DHCPv6 Vendor Class Option.

DHCPv4 Vendor Class Option:

- Option: 60
- Vendor Class Identifier Option 60: string

The format of the vendor class string shall be configured to one of the following three options:

1. ““o-ran-ru2/<vendor>, e.g., “o-ran-ru2/vendorA”
2. “o-ran-ru2/<vendor>/<product-code>”, e.g., “o-ran-ru2/vendorA/ORUAA100”
3. “o-ran-ru2/<vendor>/<product-code>/<serial-number>”,e.g. , “o-ran-ru2/vendorA/ORUAA100/FR1918010111”

DHCPv4 Vendor-Identifying Vendor Class Option:

- Option: 124
- Enterprise number: O-RAN-alliance 53148
- Vendor-Class-Data: the format of the string shall follow the rules defined for the DHCPv4 Vendor Class Option

DHCPv6 Vendor Class Option:

- Option: 16
- Enterprise number: O-RAN-alliance 53148
- Vendor-Class-Data: the format of the string shall follow the rules defined for the DHCPv4 Vendor Class Option

The DHCP Server may use this information when allocating IP addresses or when configuring management plane O-RU Controller(s) information in the O-RU.

3.1.2 Management Plane VLAN Discovery Aspects

The O-RU will be connected to one or more Ethernet ports. The transport systems may be realized such that these Ethernet ports may be configured either as an access port, where untagged Ethernet frames are used, or as a trunk port, where multiple VLANs are configured. At start up, the O-RU will typically not be able to immediately determine whether its ports are attached to remote transport equipment configured for access or trunk mode operation.

Once an O-RU completes its boot-up sequence and Ethernet connectivity is detected on at least one of its Ethernet interfaces, the O-RU starts management plane connection establishment.

The O-RU needs to determine whether it is connected to an access port or a trunk port. In particular, when connected to a trunk port, the O-RU needs to additionally determine the VLAN identity/ies used to support the management plane communication(s). The VLAN(s) used to support management plane communications can be identified by the DHCP server replying to the DHCP DISCOVER message, as described in sections 3.1.4 and 3.1.6.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Note, an O-RU which supports IPv6 may infer that a VLAN is not used to support management plane communications if it receives an IPv6 Router Advertisement without either the “managed address configuration” or “other configuration” bits set.

An O-RU may have been previously configured with management plane VLAN information, for example storing the last VLAN(s) used for management plane connectivity, and/or being previously configured with a range of management plane VLANs by a NETCONF client that has been stored in reset-persistent memory. The O-RU may use this information to optimize its discovery of the VLAN ID(s) used for management plane connectivity.

If the O-RU does not have previously configured management plane VLAN information, the O-RU shall attempt to discover DHCP servers on all of its Ethernet ports using untagged Ethernet frames.

If the O-RU does not receive a DHCP OFFER from a DHCP server using untagged frames, or previously configured VLANs, the O-RU should attempt to contact a DHCP server using individual VLANs on all of its Ethernet ports.

3.1.3 O-RU Management Plane IP Address Assignment

Automatic IP address assignment for the O-RU management plane can be achieved using different techniques:

1. IPv4 configuration using DHCPv4, RFC2131 [10] enables DHCP servers to configure IPv4 network address(es) on the O-RU. An O-RU implementing IPv4 shall support the behavior specified in RFC 4361 [33], using stable DHCPv4 node identifiers in their dhcp-client-identifier option.

Note: a network realized with multiple DHCP servers should ensure that their configurations are coordinated to ensure a common default gateway is provisioned in an O-RU which receives multiple DHCPv4 responses, e.g., when received over different interfaces.

Note: an O-RU may indicate that it supports configuration of routing information using RFC 3442, enabling static routes to be used by the O-RU when determining how to route uplink packets, e.g., when the O-RU supports multiple interfaces.

For O-RUs that support IPv6, both stateful and stateless address assignment procedures are supported:

2. IPv6 Stateless Address Auto-Configuration (SLAAC), RFC4862 [11] enables the O-RU to generate link-local and global addresses.

Note: a network realized with multiple IPv6-enabled routers that support dynamic address assignment is expected to use RFC 4191 to configure the preference of the default route prefixes learnt by the O-RU using SLAAC.

3. IPv6 State-full address configuration uses DHCPv6, RFC3315 [12] and enables DHCP servers to configure IPv6 network address(es) on the O-RU. DHCPv6 is transported using UDP, using the link-local address on the O-RU and a link-scoped multicast address on the DHCP server.

Note: the above does not restrict the realization of the DHCP server, which may be integrated with the O-DU, may be provided by the transport system, or may be accessed via a relay.

The DHCP server should operate using static bindings, i.e., ensuring an O-RU identified by a particular client hardware address will be re-allocated the same management plane IP address, e.g., after performing an O-RU reset procedure.

3.1.4 O-RU Controller Discovery

This section provides how to automatically discover the O-RU Controller address(es).

O-RUs that have obtained their IPv6 addresses by stateless address auto-configuration, shall use stateless DHCPv6, RFC3736 [13], to obtain management plane configuration information.

Other O-RUs operating using stateful IPv4 or IPv6 address allocations shall obtain management plane configuration information during IP address allocation.

The O-RU as NETCONF Server shall be able to recover NETCONF Client information using the following DHCP Options, RFC8572 [14]:

- DHCPv4 OPTION V4_SZTP_REDIRECT [143]
- DHCPv6 OPTION_V6_SZTP_REDIRECT [136]

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

These options are defined in [14] and are used to deliver bootstrap-server-list information to the O-RU. The O-RU shall use these options to recover the NETCONF client information using the above IANA defined DHCP options. The O-RU is not required to implement the remainder of the zerotouch capabilities defined in [14].

The above DHCP option provided information is encoded as a list of one or more server URIs, of the format “https://<ip-address-or-hostname>[:<port>]” signaled to the O-RU. The DHCP server shall ensure that all NETCONF client information is encoded with these options, including the optional port information using the IANA assigned ports specified in RFC8071 [15]. The O-RU shall use the included port information to decide whether to call home using NETCONF/SSH or NETCONF/TLS. If no call home port is provided, the O-RU shall attempt to call home using NETCONF/SSH.

Other O-RUs which have had their IP address(es) manually configured, shall also have their O-RU Controller(s) manually configured.

For IPv4, the O-RU may request the OPTION_V4_SZTP_REDIRECT by including its option code in the Parameter Request List (55) in DHCP discover/request messages.

For IPv6, the O-RU may request the OPTION_V6_SZTP_REDIRECT option by including the requested option codes in the Option Request Option.

Note, these operations are optional because the DHCP server will already be aware that it is communicating with an O-RU based on the recovered vendor class option.

To enable O-RUs to operate in legacy environments where the DHCP server has not been enhanced with IANA defined DHCP options for zero touch NETCONF capability, an O-RAN defined vendor specific option can be used to signal all NETCONF client information to the O-RU using option 43 for DHCPv4 and option 17 for DHCPv6. Multiple instances of NETCONF client information may be signaled, encoded as a sequence of type/length/value fields.

The definition of the types used within the DHCPv4 option 43/DHCPv6 Option 17 depends on the vendor-class option reported by the O-RU in its DHCP messages.

When a legacy O-RU reports its vendor-class using the “o-ran-ru” prefix, the following types are defined:

Type: 0x01 – O-RU Controller IP Address

Type: 0x02 – O-RU Controller Fully Qualified Domain Name

When the O-RU reports its vendor-class using the “o-ran-ru2” prefix, the following types are defined:

Type: 0x81 – O-RU Controller IP Address

Type: 0x82 – O-RU Controller Fully Qualified Domain Name

Type: 0x86 – O-RU Call home protocol

In all cases, the Type is followed by the length, which is the hexadecimal encoding of length of value field in octets, and the Value.

When Type corresponds to an O-RU Controller IP Address, the value encodes IPv4 address(es) in hexadecimal format. For example, a single server with IPv4 address 198.185.159.144 will be encoded in an option 43 TLV as

Type 0x81 (or x01 for legacy)

Length: 0x04

Value: C6 B9 9F 90

When Type corresponds to an O-RU Controller Fully Qualified Domain Name, this encodes the string representation of domain name, using ACSII encoding (i.e., following for encoding used for the domain name in the Host Name DHCP Option 12). For example, a server with FQDN “controller.operator.com” will be encoded in an option 43 TLV as

Type 0x82 (or x02 for legacy)

Length: 0x17

Value: 63 6F 6E 74 72 6F 6C 6C 65 72 2E 6F 70 65 72 61 74 6F 72 2E 63 6F 6D

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

The format of the DHCPv6 option 17 follows the format of the DHCPv4 encoding, with the additional inclusion of an Enterprise Number prior to the TLV option data. The IANA allocated private enterprise number to be used with DHCPv6 option 17 is 53148.

When Type corresponds to the call home protocol, the value encodes whether an O-RU shall call home using NETCONF/SSH or NETCONF/TLS using the IANA defined ports in [15]. If no call home protocol type is provided, the O-RU shall use NETCONF/SSH. The format is encoded as follows:

Value 00 - O-RU shall attempt to call home using NETCONF/SSH

Value 01 - O-RU shall attempt to call home using NETCONF/TLS

For example, a DHCP server wanting to trigger the call home procedure using NETCONF/TLS will encode the option 43 TLV as

Type: 0x86

Length: 0x01

Value: 01

3.1.5 Multi-Vendor Plug-and-Play

As described in sub-section 3.3.2, an O-RU may optionally support certificate enrollment using CMPv2. 3GPP 32.509 specifies how the O-RU supporting IPv4 can discover the IP address or FQDN of one or more Certification Authority (CA/RA) servers using DHCP Option 43.

An O-RU supporting IPv6 and certificate enrollment using CMPv2 shall additionally support the signaling of vendor specific options using DHCPv6 option 17. The format of the DHCPv6 option 17 follows the format of the DHCPv4 encoding, with the additional inclusion of an Enterprise Number prior to the TLV option data. The IANA allocated private enterprise number to be used with DHCPv6 option 17 is 53148 (as allocated by IANA to O-RAN Alliance).

> Note, in case of a point-to-point connection between the O-RU and the O-DU, the O-DU needs to provide means for the O-RU to access the operator CA/RA for the O-RU certificate enrollment at the IP address conveyed as described above. If the FQDN option is used instead, the O-DU needs to provide means for the O-RU to access an operator DNS server in addition.

An O-RU shall report any discovered multi-vendor plug-and-play servers using the o-ran-dhcp YANG model.

3.1.6 Event-Collector Discovery

This section describes how an O-RU automatically discovers the Event-Collector to which it shall send its pnfRegistration notification. The support by an O-RU of PNF Registration to a discovered Event-Collector is optional and hence this section only applies to those O-RUs that support this optional capability.

O-RUs that have obtained their IPv6 addresses by stateless address auto-configuration, shall use stateless DHCPv6, RFC3736 [13], to obtain Event-Collector information. Other O-RUs operating using stateful IPv4 or IPv6 address allocations shall obtain Event-Collector information during IP address allocation. Other O-RUs which have had their IP address(es) manually configured, shall also have their Event-Collector(s) and Event-Collector Notification Format manually configured.

The O-RU shall be able to recover Event-Collector information using O-RAN defined vendor specific option to signal Event-Collector information to the O-RU using option 43 for DHCPv4 and option 17 for DHCPv6.

The definition of the types used within the DHCPv4 option 43/DHCPv6 Option 17 are as follows:

Type: 0x83 – Event-Collector IP Address

Type: 0x84 – Event-Collector Fully Qualified Domain Name

Type: 0x85 – Event-Collector Notification Format

In this version of the specification, the operation of an O-RU when receiving multiple instances of the Event-Collector IP Address and/or Event-Collector FQDN information is not defined.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

In all cases, the Type is followed by the length, which is the hexadecimal encoding of length of value field in octets, and the Value.

When Type corresponds to an Event-Collector IP Address, the value encodes IPv4 address(es) in hexadecimal format. For example, an Event-Collector with IPv4 address 198.185.159.144 will be encoded in an option 43 TLV as

Type 0x83

Length: 0x04

Value: C6 B9 9F 90

When Type corresponds to an Event-Collector Fully Qualified Domain Name, this encodes the string representation of domain name, using ACSII encoding (i.e., following for encoding used for the domain name in the Host Name DHCP Option 12). For example, a server with FQDN “collector.operator.com” will be encoded in an option 43 TLV as

Type 0x84

Length: 0x17

Value: 63 6F 6C 6C 65 63 74 6F 72 2E 6F 70 65 72 61 74 6F 72 2E 63 6F 6D

In this version of the specification, the operation of an O-RU when receiving an Event-Collector FQDN that is subsequently resolved by the O-RU to more than one IP address (i.e., returning multiple Address records) is not defined.

The format of the DHCPv6 option 17 follows the format of the DHCPv4 encoding, with the additional inclusion of an Enterprise Number prior to the TLV option data. The IANA allocated private enterprise number to be used with DHCPv6 option 17 is 53148.

When Type corresponds to an Event-Collector Notification Format, the value encodes in what format the Event-Collector expects to receive asynchronous notifications. In this version of the specification, only a single format is defined:

Value 00 - Event-Collector expects the notification to be signalled using the format as specified in the ONAP VES event listener specification [36].

For example, an Event-Collector expecting the pnfRegistration notification to be signaled using the ONAP defined format will encode the option 43 TLV as

Type 0x85

Length: 0x01

Value: 00

3.2 NETCONF Call Home to O-RU Controller(s)

The O-RU aims to have NETCONF sessions with all of the known O-RU Controller(s), either discovered using the DHCP options defined in Section 3.1.4, provisioned by an existing NETCONF client, or statically configured. An O-RU controller may attempt to autonomously initiate a NETCONF session with the O-RU, e.g., triggered by the pnfRegistration procedure. In order to support NETCONF clients corresponding to known O-RU Controllers that either do not attempt to initiate a NETCONF session with the O-RU, or are prohibited from doing so, e.g., because of Network Address Translation limitations, the O-RU shall call home to all known O-RU Controllers with which it does not already have an active NETCONF session.

If the O-RU is unable to establish a NETCONF session with any of the O-RU Controller(s), the O-RU shall use the “re-call-home-no-ssh-timer” to repeatedly re-perform the call home procedure to all with which is does not have an established NETCONF session.

> Note, the same value of timer shall be used, irrespective of whether SSH or TLS is being used to transport the NETCONF session.

The O-RU shall use RFC 8071 [15] whereby the O-RU (NETCONF Server) initiates a TCP connection to the NETCONF client. The port used by the O-RU shall be the one signaled using the RFC 8572 DHCP option [14], else, if no port was signaled, the O-RU shall use the IANA-assigned port 4334 to indicate that the O-RU wants to use SSHv2 to secure the NETCONF connection and the IANA-assigned port 4335 to indicate that the O-RU wants to use TLS to secure the NETCONF connection. When the

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

NETCONF client accepts a TCP connection on the allocated port, it initiates an SSH session/TLS connection with the NETCONF Server. Using this SSH session/TLS connection, the NETCONF client initiates a NETCONF session.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

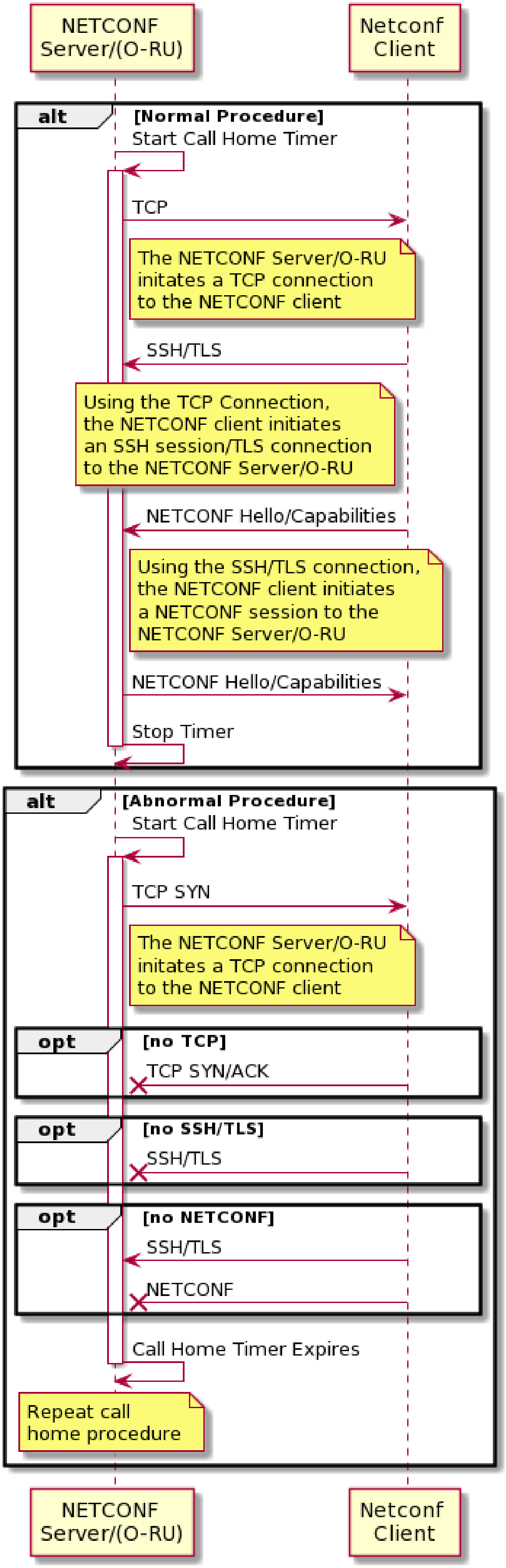


Figure 6: Outline of NETCONF call home procedure

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

The O-RU shall ensure that a persistent connection to any NETCONF client with "sudo" privileges is maintained by actively testing the aliveness of the connection using the keep-alive mechanism defined in [15]. The establishment of NETCONF client privileges is covered in Section 3.4.

3.3 NETCONF Connection Establishment

The identity of the NETCONF server (O-RU) shall be verified and authenticated by the NETCONF client according to local policy before password-based authentication data or any configuration or state data is sent to or received from the NETCONF server.

When using SSHv2, public key-based host authentication shall be used for authenticating the server (RFC 4253) by the clients. In addition, server authentication based on X.509 certificates may also be provided [31].

When using TLS, X.509 certificate-based authentication shall be used for mutual authentication between the NECTONF client and NETCONF server.

Note: SSHv2 based public key-based host authentication requires that the SSH server (O-RU) public keys are provisioned in the NETCONF client (e.g., O-DU and/or SMO). As an alternative, RFC4251 mentions that "a possible strategy is to only accept a host key without checking the first time a host is connected, save the key in a local database, and compare against that key on all future connections to that host.". This option simplifies the key management procedure as it doesn't require to pre-populate them in O-DU/SMO (SSH client) but obviously at the price of degraded security, therefore the support of this option shall be configurable and left to operator's choice.

3.3.1 NETCONF Security

As specified in clause 2.4, this version of the O-RU Management Plane Specification uses TLS 1.2 or SSHv2 for mutual authentication between the NETCONF server (O-RU) and the NETCONF client (O-DU or SMO). Support for TLS 1.2 is optional and support for SSHv2 is mandatory as described in Table 2.

If multiple NETCONF sessions are established to an O-RU, those sessions shall be established over separate SSH tunnels/TLS connections.

3.3.2 NETCONF Authentication

This version of the O-RU Management Plane Specification supports SSHv2 using password authentication method for SSHv2 [6] and client authentication based on X.509 certificates [31], as well as optional support for TLS 1.2 using X.509 certificate-based authentication.

The identity of the NETCONF server (O-RU) shall be verified and authenticated by the NETCONF client (O-DU or SMO) according to local policy before authentication data or any configuration or state data is sent to or received from the server.

The identity of the NETCONF client (O-DU or SMO) shall be verified and authenticated by the NETCONF server (O-RU) according to local policy (X.509 certificate-based or username/password) to ensure that the incoming NETCONF client request is legitimate before any configuration or state data is sent to or received from the NETCONF client. The server shall also perform proper authorization of the client before accepting any request.

If authentication is based on X.509 certificates, for the purposes of user authentication, the mapping between certificates and user names is provided by the SubjectAltName field of the X.509 certificate, which means that the user name is coded in the subjectAltName. The username is determined from the subjectAltName using the rules defined in RFC 7589. For the purposes of NETCONF server authentication, RFC 7589 [41] specifies server identity based on RFC 6125 [40].

Upon initial system initialization, the O-RU is configured with a default account. The specific details of the default account are to be agreed between operator and vendor. An example of a default user account for account-type PASSWORD is one with username "oranuser". An example of a default user account for account-type CERTIFICATE is map type "san-rfc822-name" with an rfc822-name of "oranuser@o-ran.org".

The default account may be of account-type PASSWORD, in which case a default password also needs to be defined and configured in the O-RU, for example "o-ran-password".

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Note: As the default account may be operator specific, this may require that the O-RU provides facilities to configure securely this default account at installation time (i.e. before the O-RU is connected to the O-RU Controller).

If user authentication is based on X.509v3 certificate during O-RU plug and play, to support zero touch for the first NETCONF connection, the O-RU shall support the default mapping between certificate and default NETCONF account which will map any authenticated X.509 v3 certificate to this default O-RAN account.

Note: The trust anchor for O-RU shall be provisioned automatically with online CA server during O-RU Plug and Play, and it must be same as the trust anchor of the O-RU Controller(s), thus avoiding the need for manual configuration of the peer trust anchor for O-RU.

The default account is a member of the “sudo” access control group (see Section 3.4 for details of groups/privileges) as it can be used to create other accounts (see Section 3.3.3).

The identity of the SSH client (NETCONF client) shall be verified and authenticated by the SSH server (O-RU) according to local policy to ensure that the incoming SSH client request is legitimate before any configuration or state data is sent to or received from the SSH client.

Upon initial system initialization, the NETCONF client can authenticate itself to the O-RU using SSH Authentication, with the agreed default username and password.

If authentication based on X.509 certificates according to [31] is supported by SSH client and server, the certificates need to be installed at initial system initialization, or can be obtained through certificate enrolment with operator’s PKI (certificate enrolment as defined by 3GPP with CMPv2 protocol between the NE and the operator’s CA).

3.3.3 User Account Provisioning

The NETCONF client with suitable privileges may provision user accounts on the O-RU, including the accounts (users) name, password, group (see Section 3.4 for details of groups/privileges) and whether a particular account is enabled or disabled.

- The username **name** is a string between 3-32 characters. The first character must be a lowercase letter. The remaining characters can be a lowercase letter or a number.
- The **account-type** is an enumeration, indicating whether password or certificate-based authentication is used for this account.
- The **password** is a string between 8-128 characters. Allowed characters in the password field include lowercase and uppercase letters, numbers and the special characters: ! $ % ^ ( ) _ + ~ { } [ ] . – The password leaf is not present for those user accounts associated with certificate-based authentication.
- The access control **group** associated with an account (see Section 3.4 for details of groups/privileges).
- Whether an account is **enabled.** The YANG model ensures that at least one user account is always enabled on the O-RU

The new account information (user **name**, **password**, access **group** and whether the account is **enabled**) shall be stored in reset-persistent memory in O-RU.

If certificate based client authentication is used no password needs to be provisioned. At time of SSH connection, user’s authorization is done based on the X.509 certificate’s SubjectAltName field that codes the associated account’s name.

When other user account (sudo) is created, the NETCONF client closes existing NETCONF session as described in section 3.7. Then, the O-RU disables the default account and default account stays disabled over the resets. The default account becomes enabled when the O-RU is reset to the factory default software by following the procedures defined in subsection 5.7. Any other way to enable the default account is not precluded as O-RU vendor implementation matter.

The security principle defined in this section shall follow those defined for the default account and default mapping, i.e.~~,~~ the O-RU Controller shall create a new mapping.

Note : Depending on the EE/CA certificate of the O-RU Controller, the map type can still be specified but with specific fingerprint of the EE/CA certificate or based on SubjectAltName of EE/CA certificate as specified in sub-section 3.3.2

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

3.4 NETCONF Access Control

This subsection defines the access control for NETCONF clients. Its motivation is that when multiple NETCONF clients (users) are defined, the NETCONF access control mechanism enables the NETCONF server to limit some operations for one client but allow full access for another client. In particular, for hybrid access configuration as introduced in Chapter 2, this allows the privileges associated with the NETCONF client in the O-DU to be distinct and different from the privileges associated with the NETCONF client in the SMO.

In order to support interoperable access control management, the NETCONF Server shall use the IETF NETCONF Access Control Model [RFC8341].

Currently six access control **groups** are defined per NETCONF session: “sudo”, “smo”, “hybrid-odu”, “nms”, “fm-pm”, and “swm”. The table below maps the group **name** to different privileges. Privileges are defined per namespace for read “R”, write “W” and execute “X” rpc operations or subscribe to Notifications.

Note: when operating in hybrid management, the definition of above groupings does not preclude the NETCONF client in a centralized network management system from being configured “sudo” privileges that permit it to edit the configuration used by an O-RU. However, importantly the operation of the O-DU in those situations may not be defined. For example, an O-DU when operating with an O-RU which receives an autonomous reset RPC from a centralized NMS may not result in the O-DU recovering the **o-ran-operations:operational-info/operational-state/restart-cause** from the O-RU to then determine that an NMS triggered reset has been performed. In order to reduce the possibility of such a scenario, it is recommended that when operating in hybrid mode of operation, the NETCONF client in the O-DU is associated with the “hybrid-odu” privilege group and the NETCONF client in the SMO is associated with the “smo” privilege group.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Module Rules	sudo	nms	fm-pm	swm	smo	hybrid-odu
"urn:o-ran:supervision:x.y"	RWX	---	---	---	---	RWX
"urn:o-ran:hardware:x.y"	RWX	RW-	---	---	RW-	R--
"urn:ietf:params:xml:ns:yang:ietf-hardware"	RWX	RWX	R-X	---	RWX	R-X
“urn:ietf:params:xml:ns:yang:iana-hardware”	R--	R--	R-X	---	R--	R--
"urn:o-ran:user-mgmt:x.y"	RWX note1	---	---	---	RWX note1	RWX note1
"urn:o-ran:fm: x.y "	R-X	R-X	R-X	---	R-X	R-X
"urn:o-ran:fan: x.y "	R--	R--	R--	---	R--	R--
"urn:o-ran:sync: x.y "	RWX	RWX	R--	---	RWX	R-X
"urn:o-ran:delay: x.y "	RW-	R--	R--	---	R--	RW-
"urn:o-ran:module-cap: x.y "	RW-	R--	R--	---	R--	RW-
"urn:o-ran:udpecho: x.y "	RW-	R--	---	---	RW-	R--
"urn:o-ran:operations: x.y "	RWX	RW-	R--	---	RWX	RWX
"urn:o-ran:uplane-conf: x.y "	RWX	RWX	R--	---	R--	RWX
“urn:o-ran:beamforming: x.y”	R-X	R-X	R--	---	R--	R-X
"urn:o-ran:lbm: x.y "	RW-	RW-	R--	---	RW-	R--
"urn:o-ran:software-management: x.y "	R-X	R-X	R--	R-X	R-X	R--
"urn:o-ran:file-management: x.y "	--X	--X	--X	---	--X	---
" urn:o-ran:message5: x.y "	RW-	R--	R--	---	R--	RW-
"urn:o-ran:performance-management: x.y "	RWX	RWX	RWX	---	RWX	R-X
"urn:o-ran:transceiver: x.y "	RW-	RW-	R--	---	RW-	R--
"urn:o-ran:externalio: x.y "	RWX	RWX	---	---	RWX	R--
"urn:o-ran:ald-port: x.y "	RWX	RWX	---	---	RWXnote 3	RWX
"urn:o-ran:interfaces: x.y "	RWX	RWX	R--	---	RWX	R--
"urn:ietf:params:xml:ns:yang:ietf-ip"	RW-	RW-	R--	---	RW-	R--
"urn:ietf:params:xml:ns:yang:ietf-interfaces"	RW-	RW-	R--	---	RW-	R--
"urn:o-ran:processing-elements: x.y "	RW-	RW-	R--	---	RW-	RW-
" urn:o-ran:mplane-interfaces: x.y "	RW-	RW- note2	R--	---	RW-	R--
"urn:o-ran:dhcp: x.y "	R--	R--	R--	---	R--	R--
“urn:o-ran:ald: x.y”	--X	---	---	---	--X note 3	--X
“urn:o-ran:troubleshooting: x.y”	--X	--X	--X	---	--X	---
“urn:o-ran:trace: x.y”	--X	--X	--X	---	--X	---
"urn:o-ran:laa: x.y "	RW-	RW-	---	---	R--	RW-
"urn:o-ran:laa-operations: x.y "	--X	---	---	---	---	--X
"urn:o-ran:antcal: x.y "	RWX	R--	---	---	R--	RWX
“urn:ietf:params:xml:ns:yang:ietf-netconf-acm”	RW-	R--	R--	R--	RW-	RW-
"urn:ietf:params:xml:ns:yang:ietf-yang-library"	R-X	R-X	R-X	R-X	R-X	R-X
“urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring”	R-X	R-X	R-X	R-X	R-X	R-X
Note1: The rule list for "urn:o-ran:user-mgmt:1.0” shall additionally deny reading of the password leaf by any NETCONF client Note2: The rule list for “urn:o-ran:mplane-int:1.0” shall additionally deny the writing of the **configured-client-info** container for NETCONF sessions with “nms” group privileges. Note 3: While the rule list for models related to Antenna Line Devices (ALD) permit SMO configuration privileges, the operation of the current architecture, including requiring the use of regular NETCONF RPCs to tunnel heartbeat messages to the ALD, may limit the scalability of scenarios where the SMO is responsible for the ALD Controller function described in sub-section 11.3.						

1 **Table 3 : Mapping of account groupings to O-RU module privileges (continues over page)**

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Module Rules	sudo	nms	fm-pm	swm	smo	hybrid-odu
“urn:ietf:params:xml:ns:yang:ietf-netconf-notifications”	--X	--X	--X	--X	--X	--X
"urn:o-ran:shared-cell:x.y"	RW-	RW-	---	---	R--	RW-
"urn:o-ran:ethernet-fwd:x.y"	RW-	RW-	---	---	RW-	R--
“urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications”	---	---	---	---	RWX	---
"urn:o-ran:ves-sn:x.y"	---	---	---	---	RW-	---

Table 3 (Continued): Mapping of account groupings to O-RU module privileges

This mapping shall be encoded in the **rule** list in ietf-netconf-acm.yang model. This rule list shall be unmodifiable by any NETCONF client.

The same model is responsible for configuring the mapping between different **user-names** and **groups**.

3.5 NETCONF capability discovery

The O-RU advertises its NETCONF capabilities in the NETCONF Hello message. The Hello message provides an indication of support for standard features defined in NETCONF RFCs as well as support for specific namespaces.

NETCONF capabilities are exchanged between the O-RU and the NETCONF client(s). Examples of capabilities include [3]:

- Writable-running Capability
- Candidate Configuration Capability and associated Commit operation
- Discard change operation
- Lock and un-lock operations
- Confirmed commit Capability
- Cancel commit operation
- Rollback on error capability
- Validate Capability
- Startup configuration capability
- URL capability
- XPATH capability
- Notifications
- Interleave capability

All O-RAN O-RUs shall support the XPATH capability, NETCONF Notifications and at least one of the writeable-running and candidate configuration capabilities.

The NETCONF client uses the **get** RPC together with sub-tree based <filter> and XPATH based <filter> to recover particular sub-trees from the O-RU. Please see Chapter 6 for more information on NETCONF based configuration management.

In order to avoid interactions between the operation of supervision watchdog timer (see section 3.6) and the confirmed commit timer (default value set to 600 seconds in RFC 6241), when using the NETCONF confirmed commit capability, a NECTONF client with “sudo” privileges shall ensure the confirmed-timeout is less than the **supervision-notification-interval** timer (default value 60 seconds in o-ran-supervision.yang).

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

3.6 Monitoring NETCONF connectivity

When having a session with a NETCONF client with "sudo" or "hybrid-odu" access privileges, the O-RU operates a watchdog supervision timer to ensure that the session to the NECTONF client is persistent, as described in section 3.4. Additionally, the O-RU provides NETCONF Notifications to indicate to remote systems that its management system is operational. This supervision is intended to be used with the NETCONF client associated with the operation of the peer to the O-RAN Radio's lower layer split.

> Note, if another NETCONF client is discovered or configured, e.g., in hybrid mode of operation, and is used to establish a NETCONF session with "sudo" access privileges, the same watchdog supervision rules shall apply, meaning that a NETCONF server shall support the operation of individual supervision watchdog timers for each NETCONF client with "sudo" or "hybrid-odu" privileges .

The NETCONF client is responsible for automatically enabling the operation of the supervision watchdog timer by creating supervision-notification subscription, whenever the O-RU has an established NETCONF session to a client with "sudo" or "hybrid-odu" privileges.

The O-RU uses two timers, referred generically as watchdog timers, to support the bi-directional monitoring of NETCONF connectivity;

- Notification timer:
  Value: Equal to **supervision-notification-interval** (default value: 60s)
  Operation: The O-RU sends **supervision-notification** to those NETCONF clients that have subscribed to receive such notifications. The O-RU sends **supervision-notification**, at the latest when the timer expires. The O-RU Controller confirms that NETCONF connectivity to the O-RU is operational by receiving the notification.

- Supervision timer:
  Value : Equal to **supervision-notification-interval** (default value: 60s) + **guard-timer-overhead** (default : 10s)
  Operation: The O-RU goes into supervision failure operation when the timer expires. This means that a NETCONF client with sudo privileges should repeatedly reset this supervision timer. The O-RU confirms that NETCONF connectivity to the O-RU Controller is operational.

The O-RU enables watchdog timers when it receives an rpc create-subscription from a NETCONF client with sudo privileges. The notification timer shall start when the O-RU receives a create-subscription rpc, but how the O-RU treats the supervision timer is up to O-RU's implementation based on the above definition. After the watchdog timers have been enabled, the O-RU is responsible for sending **supervision-notification** after the expiry of the notification timer. An O-RU Controller with "sudo" or "hybrid-odu" privileges shall be prepared to receive **supervision-notification** at any time when the watchdog timers are enabled.

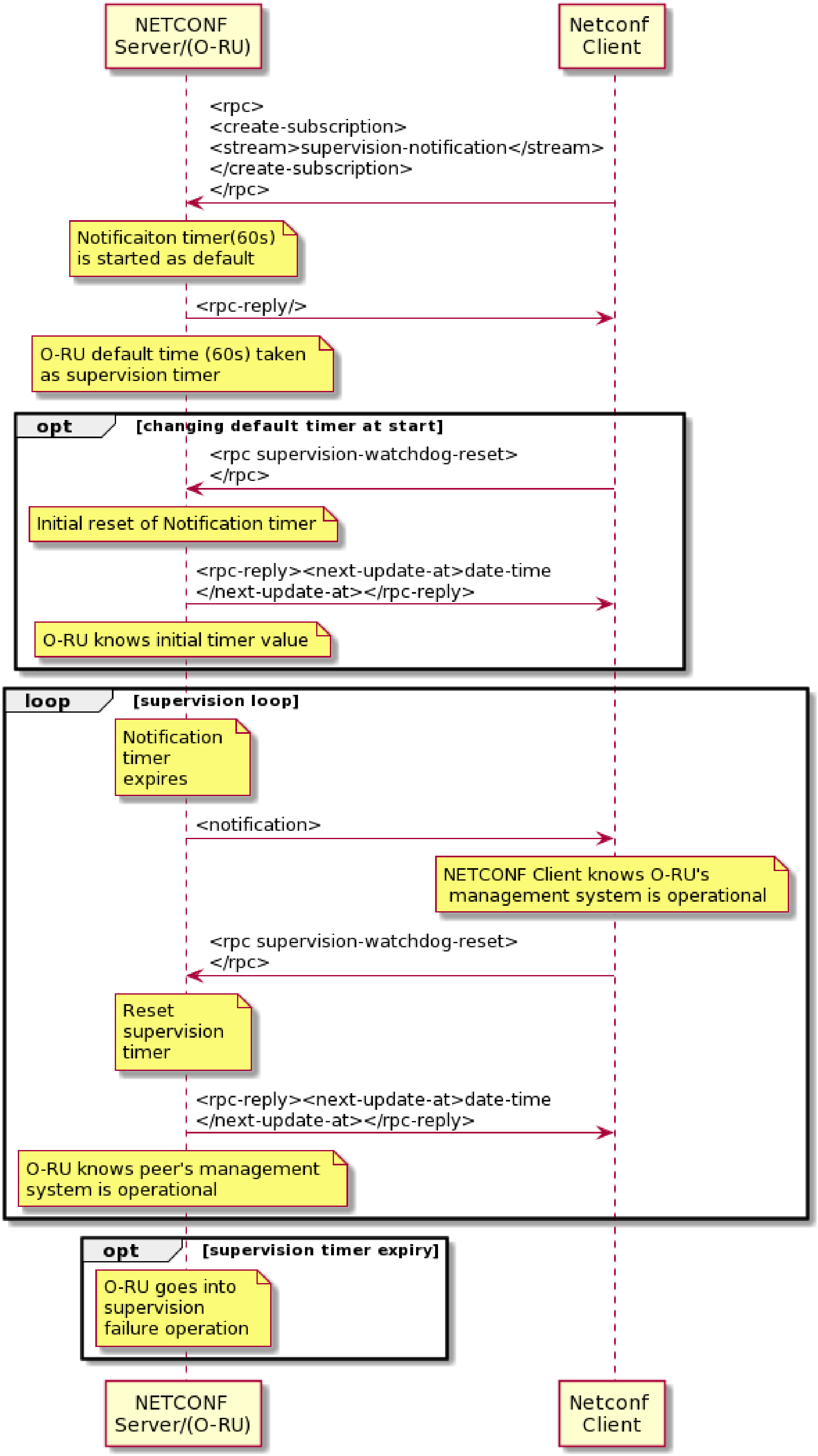
The NETCONF client is responsible for sending **supervision-watchdog-reset** RPC in order not to cause the Supervision timer to expire, and the O-RU should send next notification timestamp as **next-update-at** in reply. (Note : **next-update-at** is just informative). In the **supervision-watchdog-reset** RPC, the NETCONF client may configure new values for the watchdog timers. When the O-RU receives an rpc **supervision-watchdog-reset**, it is responsible for resetting its supervision timer and notification timer.

The NETCONF client can set new value of watchdog timer without receiving **supervision-notification** from the O-RU. The new value is taken into use immediately with respect to **supervision-watchdog-reset** RPC content. The next notification should be expected not later than at the moment addressed in timestamp provided by RPC reply.

> Note, if another NETCONF client has locked the running configuration, e.g., when operating in hybrid mode of operation, and if the O-RU Controller attempts to configure a new value of the watchdog timer(s) by sending the **supervision-watchdog-reset** RPC, then the RPC operation to reset the watchdog timer will succeed, but the related backend implementation to modify the watchdog timer(s) may fail. In such circumstances, the O-RU may use the **error-message** in the RPC output to indicate to the O-RU Controller that the configuration modification has failed.

If the supervision timer expires, the O-RU will enter "supervision failure" condition, as described in Chapter 11. If all NETCONF sessions to NECTONF clients with "sudo" privileges are closed, the O-RU shall immediately disable operation of the supervision timer.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ



1

2 **Figure 7: Monitoring NETCONF Connectivity**

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Note: This figure uses **create-subscription** for the single stream "**supervision-notification**". In order to subscribe multiple notifications, the appropriate **create-subscription** message is required. Please refer to section 8.2 for the appropriate example of **create-subscription** of multiple notifications.

3.7 Closing a NETCONF Session

A NETCONF client closes an existing NETCONF session by issuing the RPC **close-session** command. The O-RU shall respond and close the SSH session or TLS connection. The O-RU shall then re-commence call home procedures, as described in Section 3.2.

Note, under normal operations, it is expected that at least one NETCONF session with “sudo” or “hybrid-odu” privileges are long-lived and used to repeatedly reset the O-RU’s supervision watchdog timer. NECTONF clients associated with other privilege groups are not expected to operate using persistent NETCONF sessions.

Note, if a NETCONF client has been previously become known to an O-RU by being configured using NETCONF, and the NETCONF client is subsequently removed from the O-RU’s configuration, e.g., by a second NETCONF client with “sudo” privileges, the NETCONF server shall force the termination of the NETCONF session to the removed client.

3.8 PNF Registration

The support by an O-RU of PNF Registration to a discovered Event-Collector is optional and hence this section only applies to those O-RUs that support this optional capability. An O-RU that support pnfRegistration shall also support the Monitoring the Communications Channel between O-RU and Event-Collector as defined in sub-section 15.6.

3.8.1 PNF Registration Procedure

The pnfRegistration notification is a JSON encoded message sent from the O-RU to the discovered Event-Collector using REST/HTTPS. As a pre-condition to performing PNF Registration, the O-RU first receives the Event-Collector information encoded in a DHCP/DHCPv6 option as described in sub-section 3.1.6. The O-RU shall attempt to establish a HTTP connection to the discovered Event-Collector using TLS to authenticate the connection. It shall then signal the pnfRegistration notification over the HTTP/TLS connection. The sending of the pnfRegistration notification is repeated periodically until the SMO establishes a NETCONF session with the O-RU.

An O-RU that is performing the PNF registration procedure whilst simultaneously performing the call home procedure described in sub-section 3.2, shall be able to determine that the SMO has established a NETCONF session with the O-RU. This is identified by the O-RU analyzing the source IP address from which the NETCONF originates, based on the assumption that the NETCONF session from the SMO originates from an IP address that is distinct from the IP address(es) of the “known O-RU Controller(s)” to which the O-RU is simultaneously performing the call home procedure.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

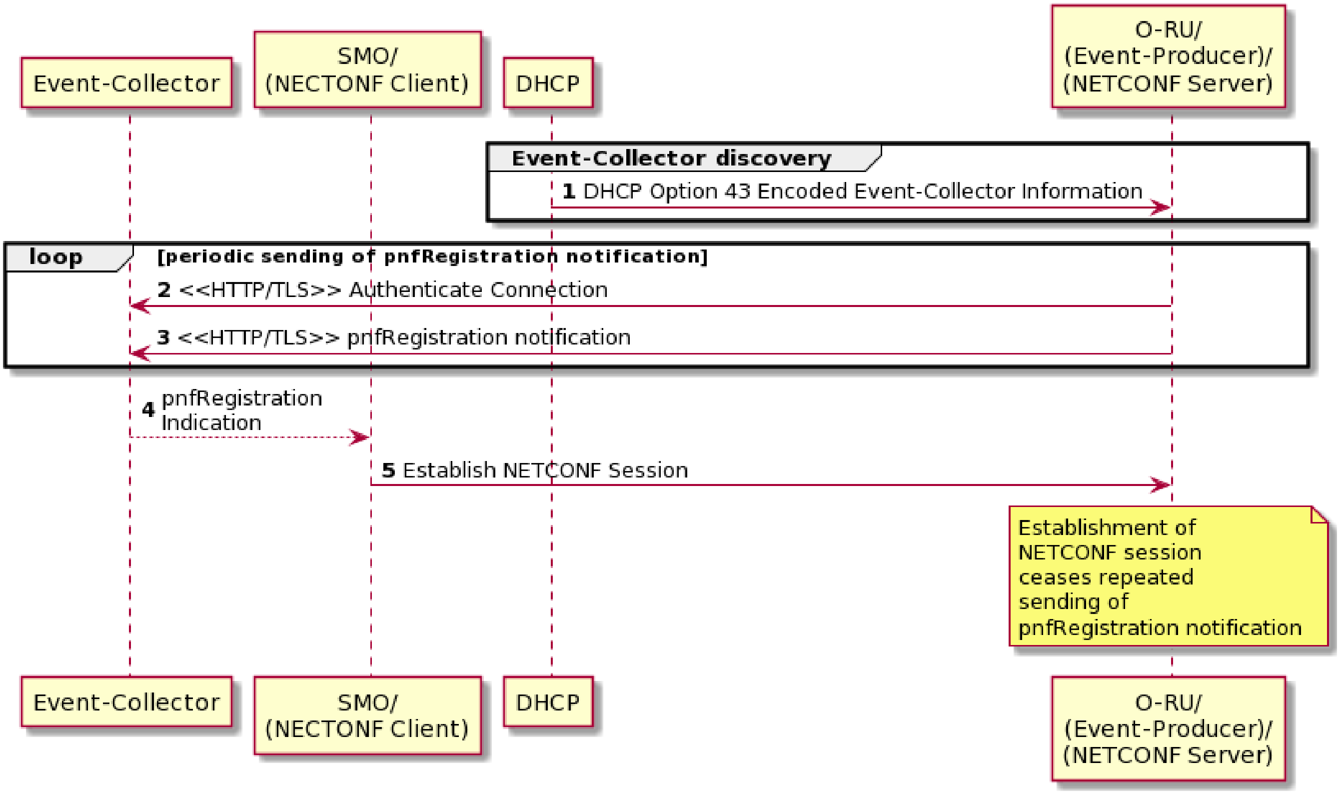


Figure 8: PNF Registration Procedure

3.8.2 Encoding of PNF Registration Notification

In this version of the specification, the encoding of the pnfRegistration notification follows the ONAP definition [37].

The pnfRegistration notification shall include the IP address information necessary for a NETCONF client to establish IP connectivity to the NETCONF Server in the O-RU, i.e., shall include the field oamV4IpAddress when the O-RU has a configured IPv4 interface and/or the field oamV6IpAddress when the O-RU has a configured IPv6 interface.

The contents of the pnfRegistration notification are derived from the O-RU's configuration database. An O-RU shall support the **o-ran-hardware.yang** model revision 5.0.0, or later, which defines the schema nodes corresponding to unitFamily and unitType values and the **o-ran-operations.yang** model revision 5.0.0, or later, which defines the schema nodes corresponding to the version of pnfRegistration fields.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

PnfRegistration Notification Field	Mandatory/ Conditional/ Optional	YANG Operational Data
lastServiceDate	O	/hw:hardware/hw:component/or-hw:last-service-date
macAddress -	O	/if:interfaces/if:interface/o-ran-int:mac-address
manufactureDate	O	/hw:hardware/hw:component/hw:mfg-date
modelNumber	M	/hw:hardware/hw:component/hw:model-name
oamV4IpAddress	C	/if:interfaces/if:interface/ip:ipv4/ip:address/ip:ip
oamV6IpAddress	C	/if:interfaces/if:interface/ip:ipv6/ip:address/ip:ip
pnfRegistrationFieldsVersion	M	/o-ran-ops:operational-info/o-ran-ops:declarations/o-ran-ops:supported-pnf-registration-fields-version
serialNumber	M	/hw:hardware/hw:component/hw:serial-num
softwareVersion	M	/hw:hardware/hw:component/hw:software-rev
vendorName	M	/hw:hardware/hw:component/hw:mfg-name

Table 4: Mapping from O-RU's Operational Data to PnfRegistration fields

Chapter 4 O-RU to O-DU Interface Management

An O-RU has a number of network interfaces, including Ethernet, VLAN and IP interfaces. This section describes the management of these network interfaces.

4.1 O-RU Interfaces

The O-RU's configuration for its interfaces is defined using the o-ran-interfaces.yang module. This module augments the standard ietf-interfaces.yang and ietf-ip.yang modules. The O-RU's interfaces are built on a layering principle where each interface has a unique **name**.

All interfaces are referenced by their **port-number** and **name.** The base interface corresponds to the Ethernet interface. These leafs describe the maximum transmission unit (**l2-mtu**), the hardware-address as well as optional alias mac addressees that may be used to transport the CU plane. Above the Ethernet interface are VLAN interfaces. Both Ethernet and VLAN interfaces can support IP interfaces. IP interfaces are defined using the standard ietf-ip.yang model. Accordingly, each IP interface can have an IPv4 and/or IPv6 interface(s) defined. Operational state associated with these interfaces provide additional detail of the layer 3 configuration, including prefix(es), domain name servers and default gateway addresses.

Finally, leafs associated with CoS and DSCP marking are defined, enabling independent configuration of CoS and DSCP markings for u-plane, c-plane and m-plane traffic. As a default, all user-plane flows are marked identically by the O-RU. Optionally, the interfaces can be configured to support enhanced user plane marking for up-link traffic whereby different CoS or DSCP values can be configured. This enables individual receive endpoints in the O-RU to be configured with different markings to then enable differentiated handling of up-link flows by the transport system.

Because the o-ran-interfaces model defines augments to the ietf-interfaces model, the O-RU can leverage the definition of operational state in ietf-interfaces to optionally report packet and byte counts on a per interface basis. A single RPC is defined in the o-ran-interfaces module, to enable these counters to be reset.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

4.2 Transceiver

The o-ran-transceiver YANG module is used to define operational state for the pluggable transceiver module (like SFP, SFP+, SFP28, XFP and QSFP, QSFP+, QSFP28, QSFP56). Each transceiver is associated with a unique **interface-name** and **port-number**.

A digital diagnostic monitoring interface for optical transceivers is used to allow access to device operating parameters. As specified in SFF-8472 [16] and SFF-8636 [34], data is typically retrieved from the transceiver module in a file. This file may be obtained from O-RU by the NETCONF client. Please see Chapter 9 for more details.

With QSFP form factor, the optical links may be multi-wavelengths (4xTx & 4xRx) and/or multi-fibers (MPO - Multifiber Parallel Optic). The QSFP digital diagnostic interface [34] describes the use of optical lanes and the O-RU interface management defines alarm 29 :“transceiver fault” for all media lanes.

The byte with offset i (i=0, …, 511) from the beginning of the file is the byte read from data address i of the transceiver memory at two-wire interface address 0xA0 if i<256, otherwise it is the byte read from data address i-256 of the transceiver memory at two-wire interface address 0xA2. The retrieved data is stored in the file without any conversion in binary format.

The O-RU stores data from the transceiver module on transceiver module detection during startup. The data from the transceiver module is saved in the file. A NETCONF client can upload it by using the File Upload procedure defined in chapter 9. The O-RU does not synchronize contents of the file with transceiver memory in runtime, therefore bytes representing dynamic information are expected to be outdated. The O-RU does not remove the file on transceiver module removal. If a transceiver module is inserted during File Upload procedure then the procedure may provide a file with previous content or fail (with failure reason as listed in File Upload procedure). If the O-RU is unable to retrieve the data from the transceiver module or it is not present then the O-RU does not create the file or removes the file created earlier (note that File Upload procedure requesting non-existing file shall fail).

The file name shall have the following syntax:

sfp_{portNumber}.sffcap

where {portNumber} is the value of **port-number** leaf of the corresponding list of port-transceiver data. Examples: sfp_0.sffcap, sfp_1.sffcap.

4.3 C/U Plane VLAN Configuration

Within the o-ran-interfaces YANG model, each named Ethernet interface includes a leaf to indicate whether VLAN tagging is supported. By default, VLAN tagging is enabled on all interfaces. This permits an O-RU to autonomously discover that it is connected to a trunk port, as described in Section 3.1.2.

When an O-RU is connected to a trunk port, VLANs will also typically be assigned to the C/U plane connections. The VLAN(s) used to support C/U plane transport may be different from the VLAN(s) used to support management plane connectivity. The VLAN assigned to the U-Plane must be the same as the VLAN assigned to the C-Plane for any given eAxC_ID. When different VLANs are used, the C/U plane VLANs shall be configured in the O-RU by the NETCONF client. In such circumstances, as defined in o-ran-interfaces, the NETCONF client shall configure separate named interfaces for each active VLAN. This configuration will define a C/U-Plane named VLAN interface as being the **higher-layer-if** reference for the underlying Ethernet interface and the underlying Ethernet interface is defined as being the **lower-layer-if** reference for the named VLAN interface.

4.4 O-RU C/U Plane IP Address Assignment

In this release, the support for C/U plane transport over UDP/IP is optional and hence this section only applies to those O-RUs that support this optional capability.

An O-RU that supports C/U plane transport over UDP/IP shall support IPv4 and/or IPv6 based transport. A NETCONF client can receive a hint as to whether an O-RU supports a particular IP version by using the **get** rpc to recover the list of **interfaces** supported by the O-RU and using the presence of the augmented **ipv4** container or **ipv6** container in the o-ran-interfaces YANG module as an indication that a particular IP version is supported.

The IP interface(s) used to support UDP/IP based C/U plane transport may be different than the IP interface(s) used to support management plane connectivity. When different IP interface(s) is/are used, the C/U plane IP interfaces shall be configured in

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

the O-RU by the NETCONF client by using the ietf-ip YANG model to configure the **IPv4** container and/or **IPv6** container. When defined by the NETCONF client, this interface shall be configured using either a named Ethernet interface (i.e., where the interface **type** is set to **ianaift:ethernetCsmacd)** and/or a named VLAN interface (i.e., where the interface **type** is set to **ianaift:l2vlan)**, depending upon whether VLANs are used to support IP based C/U plane traffic.

When a separate C/U plane IP interface is configured by the NETCONF client, additionally the NETCONF client may statically configure the IP address(es) on this/these interface(s). If the NETCONF client does not statically configure an IP address, the O-RU shall be responsible for performing IP address assignment procedures on the configured interfaces.

When an O-RU has not been configured with a static IP address, the O-RU shall support the IP address assignment using the following techniques:

When the O-RU supports IPv4:

1. IPv4 configuration using DHCPv4 [10].

and when the O-RU supports IPv6:

2. IPv6 Stateless Address Auto-Configuration (SLAAC) [11].
3. IPv6 State-full address configuration uses DHCPv6 [12].

4.5 Definition of processing elements

The CU-plane application needs to be uniquely associated with specific data flows. This association is achieved by defining an O-RU "processing element" which can then be associated with a particular C/U plane endpoint address [2] or delay measurement operation. Unless specified otherwise, a common processing element is required to be configured for the control and user-plane application components associated with any individual eAxC_ID.

The O-RU management plane supports different options for defining the transport-based endpoint identifiers used by a particular processing element (used depending on transport environment), supporting the following 3 options:

- Processing element definition based on usage of different (alias) MAC addresses;
- Processing element definition based on a combination of VLAN identity and MAC address; and
- Processing element definition based on UDP-ports and IP addresses.

  Note: there is no well-defined source port currently allocated by IANA for the o-ran application and hence the NETCONF client is responsible for configuring this port number in the O-RU.

A processing element defines both the local and remote endpoints used with a specific data flow. The processing element definition includes its element **name** which is then used by other systems to refer to a particular processing element instance.

The o-ran-interfaces YANG model is used to define feature support for C/U plane transport based on alias MAC addresses and UDP/IP. The exchange of NETCONF capabilities is used to signal which optional capabilities are supported by the O-RU, as described in Annex C.

The o-ran-processing-elements YANG model uses a **processing-elements** container to define a list of processing elements. Each processing element is identified by a unique element **name**. Each processing element references a particular **interface-name** used to support the data flows associated with a particular processing element. Depending upon the type of C/U plane transport session, additionally leafs are configured that specify MAC addresses, and/or VLANs and/or IP addresses and/or UDP ports used to identify a particular processing element.

The O-RU may discard any received CU-plane messages , i.e., eCPRI/IEEE 1914 frames/packets, which are not transported using a configured processing element.

4.6 O-DU Verification of C/U Plane Transport Connectivity

As described above, there will likely be multiple C/U-plane data flows being exchanged between the O-DU and the O-RU. In order to enable checks verifying end-to-end transport connectivity between the O-DU and O-RU, the O-RU shall support transport connectivity check capabilities using a request/reply function.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Using that connectivity monitoring procedure, reachability/connectivity checks between user plane endpoints can be performed by the O-DU:

- During O-RU configuration, to validate the transport configuration
- At runtime to monitor network connectivity

In packet networks connectivity checking is usually done by exchanging probe-messages between the endpoints. The periodicity of this exchange depends on the use case. For availability measurement, the only use case relevant for this specification, the periodicity is usually between 1 and 60 seconds.

Two different network protocols are defined for performing the transport connectivity check procedure:

- For C/U sessions over Ethernet: Loop-back Protocol (LB/LBM) as defined by IEEE 802.1Q (amendment 802.1ag) [17].
- When the O-RU supports C/U sessions over IP: UDP echo, RFC 862 [18].

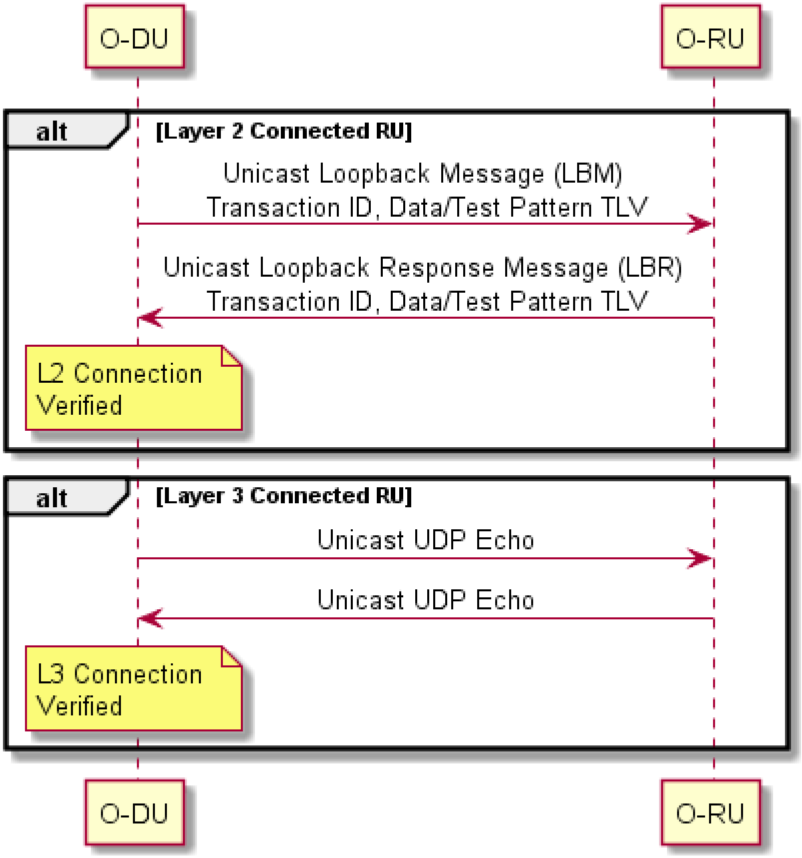


Figure 9: C/U Plane Transport Connectivity Verification

4.6.1 Ethernet connectivity monitoring procedure

If the O-RU and O-DU are operating their C/U sessions on Ethernet, the transport connectivity verification checks operate at the Ethernet layer. In this O-RU Management Plane Specification, the protocol for Ethernet connectivity monitoring is based on the Loop-back Protocol as defined by IEEE 802.1Q (amendment 802.1ag) [17].

For the purpose of connectivity monitoring all C/U -plane messaging endpoints in the fronthaul network are part of the same Maintenance Entity (ME). They each get the assigned the role of a Maintenance association End Point (MEP) for LBM.

The sending of Loop-back Messages (LBMs) is administratively initiated and stopped in the O-DU. Therefore, sending LBM requests needs to be requested by an administration entity, specifying an Ethernet MAC address assigned to the O-RU responder. In this O-RU Management Plane Specification requests are always sent from an O-DU to a unicast Ethernet MAC address of an O-RU.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

4.6.1.1 Validating the transport configuration

After setting up a U/C-plane session between an O-DU and an O-RU, the O-DU can test whether connectivity exists as per the configuration. To achieve that, at the time a U/C-plane messaging endpoint becomes operational at an O-RU, it starts an LBM responder application which automatically responds to incoming LBM requests on that endpoint. Based on a configuration command the O-DU starts sending out a predefined number of LBM requests to its O-RU(s) at a predefined interval, storing the information received in LBM responses from the O-RU(s) in an internal database. O-RU(s) are identified by both Ethernet MAC address and the CU plane VLAN.

Note, the O-RU shall be able to respond to LoopBack Messages received from different remote Maintenance Association Endpoints

In case the configuration of the session is indeed correct, the O-DU should receive LBM responses from the O-RU(s) within a time frame dependent on the network latency and the O-RU's reaction time. If LBMs from the O-RU(s) are being received, the session is determined to be operational.

4.6.1.2 Monitor network connectivity

After the procedure described in section 4.6.1.1 has been executed successfully, a further procedure may be executed continuously to maintain the connectivity status. To achieve this the O-DU continues to send out LBM requests at the configured interval. It also keeps track of LBM responses received.

Based on the LBM responses received the O-DU shall decide on the connectivity status. Connectivity shall be assumed to be available as long as LBM responses from the O-RU(s) are being received at the configured interval. Connectivity shall be assumed not available if no LBM response from the particular O-RU has been received for an interval that is as long as 3 x the configured LBM request interval or longer.

4.6.1.3 Managing Ethernet connectivity monitoring procedure

An O-DU may have one or more Ethernet interfaces that have to support the Ethernet connectivity monitoring procedure. This section describes the management of this function. The module described here is based on (i.e. a subset of) the mef-cfm module defined by the Metro Ethernet Forum [19]. This is to allow for a later extension of the module to the full feature set of mef-cfm.

The YANG module provided below supports the configuration and fault management of the Loop-back Protocol as defined by IEEE 802.1Q (amendment 802.1ag).

Derived from MEF CFM YANG, the subset of type definitions are defined as part of the o-ran-lbm.yang.

4.6.2 IP connectivity monitoring procedure

If the O-RU and O-DU are connected using IP (and UDP/IP is being used to transport the C/U plane), these transport connectivity verification checks operate at layer 3. Layer 3 connection verification is based on the O-RU supporting the UDP echo server functionality, RFC 862 [18]. The NETCONF client is responsible for enabling the UDP echo server in the O-RU, triggering the O-RU to listen for UDP datagrams on the well-known port 7. When a datagram is received by the O-RU, the data from it is sent back towards the sender, where its receipt can be used to confirm UDP/IP connectivity between the endpoints.

4.6.2.1 Managing IP Connectivity Monitoring Procedure

This section describes the management of the UDP echo functionality. The NETCONF client uses the **enable-udp-echo** leaf in the udp-echo YANG model to control operation of the UDP echo server in the O-RU. The NETCONF client is able to control the DSCP marking used by the O-RU when it echoes back datagrams using the **dscp-config** leaf. Additionally, the NETCONF client can recover the number of UDP Echo messages sent by the O-RU by using **echo-replies-transmitted** operational state.

An O-DU may have one or more IP interfaces that have to support the UDP/IP connectivity monitoring procedure. An O-RU with its UDP echo server enabled shall be able to respond to UDP datagrams originated from any valid source IP address.

4.7 C/U-Plane Delay Management

The Intra-PHY lower layer fronthaul split has the characteristic of a stringent bandwidth and tight latency requirement. The CUS-Plane specification [2] describes how the propagation delay incurred due to distance between the O-DU and O-RU is an important parameter in defining the optimization of windowing and receive-side buffering operations. This section describes the procedures that are used to manage the delay parameters for the fronthaul split.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

4.7.1 Delay Parameters

The reference points for delay management are introduced in [2] and included for reference in Figure 10.

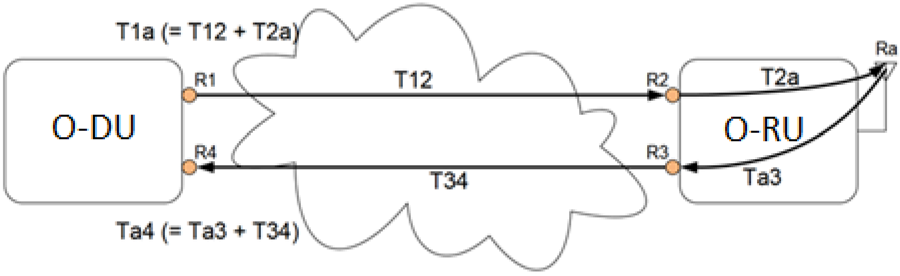


Figure 10: Definition of reference points for delay management

Important delay parameters related to the operation of the O-RU are referred to as the O-RU delay profile. As the delay characteristics for an O-RU may vary based on air interface properties, a table of the parameters is provided based on a combination of sub-carrier spacing (SCS) and channel bandwidth. When considering the downlink data direction, these parameters include:

T2a_min : Corresponding to the minimum O-RU data processing delay between receiving the last data sample over the fronthaul interface and transmitting the first IQ sample at the antenna.

T2a_max: Corresponding to the earliest allowable time when a data packet is received before the corresponding firs IQ sample is transmitted at the antenna.

Using the above parameters, (T2a_max – T2a_min): The difference between these two parameters corresponds to the O-RU reception window range.

T2a_min_cp_dl: Corresponding to the minimum O-RU data processing delay between receiving downlink real time control plane message over the fronthaul interface and transmitting the corresponding first IQ sample at the antenna.

T2a_max_cp_dl: Corresponding to the earliest allowable time when a downlink real time control message is received before the corresponding first IQ sample is transmitted at the antenna.

Tcp_adv_dl: Corresponding to the time difference (advance) between the reception window for downlink real time Control messages and reception window for the corresponding IQ data messages.

The delay parameters related to the operation of the O-RU for the uplink data direction include:

Ta3_min: Corresponding to the minimum O-RU data processing delay between receiving an IQ sample at the antenna and transmitting the first data sample over the fronthaul interface.

Ta3_max: Corresponding to the maximum O-RU data processing delay between receiving an IQ sample at the antenna and transmitting the last data sample over the fronthaul interface.

Using the above parameters, (Ta3_max – Ta3_min): The difference between these two parameters corresponds to the O-RU transmission window range.

T2a_min_cp_ul: The minimum O-RU data processing delay between receiving real time up-link control plane message over the fronthaul interface and receiving the first IQ sample at the antenna.

T2a_max_cp_ul: The earliest allowable time when a real time up-link control message is received before the corresponding first IQ sample is received at the antenna.

When requested, all O-RUs shall signal the table of statically “pre-defined” values of the above parameters, for different supported combinations of SCS and channel bandwidth over the management plane interface. This will typically occur during the initial startup phase.

4.7.2 Reception Window Monitoring

The O-RU shall monitor operation of its reception window, monitoring the arrival of packets received over the fronthaul interface relative to the earliest and latest allowable times as defined by the values of T2a_max and T2a_min respectively.

See Annex B.2 for information on reception window counters.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

4.8 O-RU Adaptive Delay Capability

O-RUs may optionally support the ability to optimize their buffers based on information signalled concerning the configuration of the O-DU, e.g., including the O-DU delay profile, together with transport delay information, which may have been derived by the O-DU by using a delay measurement procedures operated by the O-DU or by other techniques. This section describes such optional O-RU buffer optimization functionality.

An O-RU that supports the optional adaptive timing capability shall indicate such to the O-RU Controller client by exchanging NETCONF capabilities, as described in Section 6.2 and Annex C indicating that it supports the ADAPTIVE-O-RU-PROFILE feature. An O-RU Controller may then provide the O-RU with the O-DU delay profile based on a combination of sub-carrier spacing (SCS) and channel bandwidth, comprising the following parameters:

T1a_max_up: Corresponding to the earliest possible time which the O-DU can support transmitting an IQ data message prior to transmission of the corresponding IQ samples at the antenna

TXmax: Corresponding to the maximum amount of time which the O-DU requires to transmit all downlink user plane IQ data message for a symbol.

Ta4_max: Corresponding to the latest possible time which the O-DU can support receiving the last uplink user plane IQ data message for a symbol

RXmax: Corresponding to the maximum time difference the O-DU can support between receiving the first user plane IQ data message for a symbol and receiving the last user plane IQ data message for the same symbol.

T1a_max_cp_dl: Corresponding to the earliest possible time which the O-DU can support transmitting the downlink real time control message prior to transmission of the corresponding IQ samples at the antenna.

In addition to the O-DU delay profile, the O-RU-Controller provides the O-RU with the transport network timing parameters:

T12_min: Corresponding to the minimum delay between any O-DU and O-RU processing elements

T12_max: Corresponding to the maximum delay between O-DU and O-RU processing elements

T34_min: Corresponding to the minimum delay between any O-RU and O-DU processing elements

T34_max: Corresponding to the maximum delay between O-RU and O-DU processing elements

The O-RU may use this information to adapt its delay profile, ensuring that the inequalities defined in Annex B of [2] are still valid

The O-RU controller will typically provide this information during the O-RU's start-up procedure. If an O-RU receives the adaptive delay configuration information when operating a carrier, the O-RU shall not adapt its O-RU delay profile until all carriers operating using the O-RU buffers have been disabled. Once an O-RU has adapted its O-RU profile, it will include the newly adapted timing values when signaling its delay parameters to a NETCONF client.

4.9 Measuring transport delay parameters

An O-RU may optionally indicate that it supports the eCPRI based measurement of transport delays between O-DU and O-RU, as described in [2].

When indicated that it supports the optional eCPRI based delay measurement, an O-RU shall support the processing of delay measurement messages at any time, including whenever the C/U plane is configured. As well as the O-RU including its timing compensation values in the eCPRI delay measurement messages (tcv1 and tcv2 shown in the figure below), the YANG model additionally enables a NETCONF client to recover these parameters from the O-RU over the M-Plane interface.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

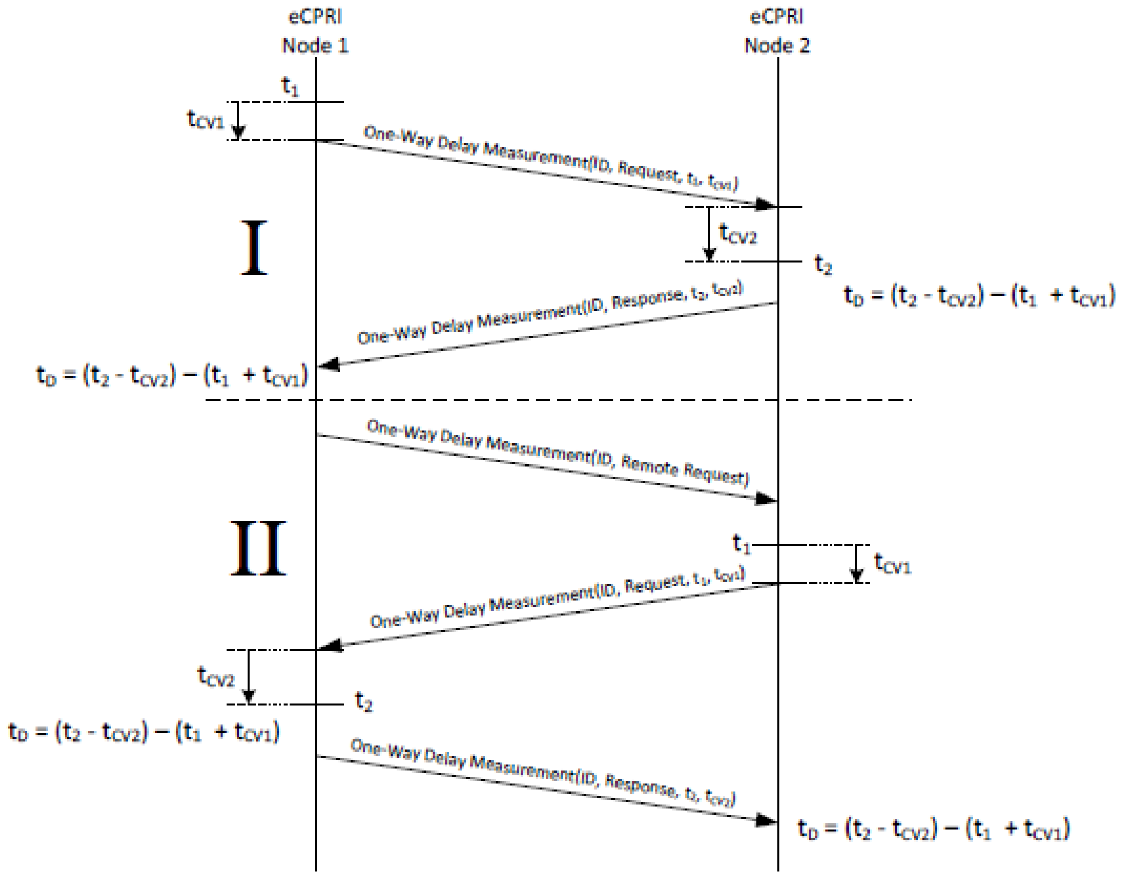


Figure 11: eCPRI One-Way Delay Measurement procedure [2]

An O-RU that supports the eCPRI based delay measurement capability, shall be able to support simultaneous operation of delay measurements over any processing element configured as described in section 4.5. For each processing element configured for use by the delay measurement procedure, the O-RU shall keep a record of the number of responses, requests and follow-up messages transmitted by the O-RU.

4.10 O-RU Monitoring of C/U Plane Connectivity

The O-RU is responsible for monitoring the C/U plane connection and raising an alarm if the logical C/U-plane connection associated with a processing-element fails. The O-RU uses a timer to monitor the C/U plane connection on a per processing-element basis. This timer is enabled only when at least one array-carrier using the processing-element is in the active state and is reset whenever it receives any C/U plane data flows associated with the particular processing-element. Because of the variety of PHY and C/U plane configurations, the O-RU cannot independently determine the minimum frequency of messages across the fronthaul interface. Therefore, as a default, the O-RU shall use a timer value of 160 milliseconds for monitoring the C/U plane connection. An O-RU may indicate that its C/U-plane monitoring timer is configurable, by the presence of the **cu-plane-monitoring** container in the o-ran-supervision.yang model. A NETCONF client can use the container to configure the O-RU's timer value, including being able to disable the operation of C/U Plane monitoring.

Note, if the O-RU supports this timer, then depending on how long the O-DU takes to initiate sending of C/U plane data flows, it may be advisable for the NETCONF client to initially disable the operation of the timer before carrier activation. Such an approach will avoid the O-RU sending spurious alarm notifications triggered by O-DU delays in initializing the sending of C/U plane data that exceed the default timer value. Once C/U plane data flows have commenced, the NECTONF client can re-configure the timer with the desired value and hence activate monitoring of the C/U plane connectivity by the O-RU.

4.11 Bandwidth Management

An O-RU can indicate the maximum bitrate able to be supported on those interfaces associated with a particular physical port using the optional **nominal-bitrate** leaf in the o-ran-transceiver YANG module. When the sustainable bitrate able to be supported by an O-RU is less than the combined bitrates of all its physical ports, an O-RU can use the optional **interface-grouping** container in the o-ran-interfaces YANG model to define the maximum sustainable rate able to be supported by an **interface-group-id** corresponding to a group of one or more physical interfaces. The same YANG model is used to augment the ietf-interfaces defined **interface** list with the **interface-group-id** to which the interface belongs.

> Note, the maximum sustainable bandwidth is calculated over one radio frame, meaning that the peak bandwidth can exceed the defined value over time periods shorter than one radio frame.

Chapter 5 Software Management

The Software Management function provides a set of operations allowing the desired software build to be downloaded, installed and activated at O-RU. Successful software activation operation does not mean an O-RU is running the just activated software build. An O-RU **reset** RPC is required to trigger the O-RU to take the activated software build into operational use.

A single software build is considered as set of internally consistent files compliant within such a build. Replacement of files within a build is prohibited, as this will cause software version incompatibility. Software build is a subject of versioning and maintenance and as such cannot be broken.

The use of compression and ciphering for the content of the software build is left to vendor implementation. The only file which shall never be ciphered is the manifest.xml file.

It is also Vendor's responsibility to handle SW Build / package / file integrity check.

The O-RU provides a set of so called "software slots" or "slots". Each slot provides an independent storage location for a single software build. The number of slots offered by O-RU depends on the device’s capabilities. At least two writable slots shall be available at the O-RU for failsafe update operation. Presence of read only slot is optional. The software slots are resources provided by the O-RU and as such are not the subject of creation and deletion. The size of individual software slots is fixed and determined by the O-RU's vendor and sufficient to accommodate the full software build.

Note: procedures used in Software Management are covered by o-ran.software-management.yang module.

5.1 Software Package

The software package is delivered by the O-RU vendor.

Each software package includes:

- manifest.xml
- software files to be installed on O-RU

The name of package should follow the following format:

“<Vendor Code><Vendor Specific Field>[#NUMBER].EXT”

Where:

- *Vendor Code* is a mandatory part which has two capital characters,
- *Vendor Specific Field* is any set of characters allowed in filename. The value must not include character “_” (underscore) or “#” (hash). The value can be defined per vendor for the human readable information. Version information is necessary in the *Vendor Specific Field* which defines load version,
- *NUMBER* is optional and used when the original file is split into smaller pieces – number after “#” indicates the number of a piece. Numbering starts from 1 and must be continuous,

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

- *EXT* is a mandatory part which defines the extension of filename. A vendor provides one or more software packages. Each software package shall be compressed by zip.

Note: the name of package needs a <Vendor Code> prefix to avoid issue if two vendors provide files with same name.

Note: the NETCONF client shall support ZIP functionality to enable support of compressed files types.

Note: the operator needs to manage and control which O-RU files will be stored and used in the file server, e.g., based on O-RU files provided by O-RU vendors and network configurations. The operator needs to make sure that only the expected version of O-RU files will be transferred from the file server to the O-RU. Different versions of files for O-RU with the same vendor and same product code should be avoided.

The content of the manifest.xml file allows to maintain software update process correctly in terms of compatibility between O-RU hardware and software build to be downloaded. The content of the Manifest file prohibits the O-RU from installing software builds designed for device based on different hardware. The format of the manifest.xml file is:

```
<xml>
<manifest version=”1.0”> /// @version describes version of file format (not the content)
    <products>
        <product vendor=”XX” code=”0818820\.x11” name=”RUXX.x11” build-Id=”1”/>
        <product vendor=”XX” code=”0818820\.x12” name=”RUXX.x12” build-Id=”1”/>
        <product vendor=”XX” code=”0818818\…” name=”RUYY” build-Id=”2”/>
        /// @vendor is as reported by O-RU
        /// @code is a regular expression that is checked against productCode reported by O-RU
        /// @name is optional and used for human reading - MUST NOT be used for other purposes!
        /// @buildId is value of build@id (see below)
    </products>
    <builds>
        <build id=”1” bldName=”xyz” bldVersion=”1.0”>
        /// @id is index of available builds.
        /// @bldName and @bldVersion are used in YANG (build-name, build-version)
            <file fileName=”xxxx” fileVersion=”1.0” path=”full-file_name-with-path-relative-to-
package -root-folder” checksum=”FAA898”/>
            <file fileName=”yyyy” fileVersion=”2.0” path=”full-file_name-with-path-relative-to-
package -root-folder” checksum=”AEE00C”/ >
            /// @fileName and @fileVersion are used in YANG (name, version)
            /// @path is full path (with name and extension) of a physical file, relative to package
root folder,used in YANG (local-path)
            /// @checksum is used to chech file integrity on O-RU side
        </build>
        <build id=”2” bldName=”xyz” bldVersion=”1.0”>
            <file fileName=”xxxx” fileVersion=”1.0” path=”full-file_name-with-path-relative-to-
package -root-folder” checksum=”FAA898”/>
            <file fileName=”yyyy” fileVersion=”2.0” path=”full-file_name-with-path-relative-to-
package -root-folder” checksum=”AEE00C”/ >
            <file fileName=”zzzz” fileVersion=”1.5” path=”full-file_name-with-path-relative-to-
package -root-folder” checksum=”ABCDEF”/ >
        </build>
    </builds>
</manifest>
</xml>
```

Note: Keywords in manifest.xml example are in **bold**, the keywords must be strictly followed considering of cross-vendor cases.

Note 2: Correspondence between content of manifest.xml tags and content of o-ran-software-management.yang is:

- XML tag "product vendor" corresponds to content leaf "vendor-code",

- XML tag "code" corresponds to content of leaf "product-code",

- XML tag "build-Id" corresponds to content of leaf "build-id",

- XML tag "bldName" corresponds to content of leaf "build-name",

- XML tag "bldVersion" corresponds to content of leaf "build-version",

- XML tag "fileName" corresponds to content of leaf "name" in list "files",

-XML tag "fileVersion" corresponds to content of leaf "version" in list "files"

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

5.2 Software Inventory

Pre-condition:

- M-Plane NETCONF session established.

Post-condition:

- NETCONF client successfully collected the software inventory information from NETCONF server.

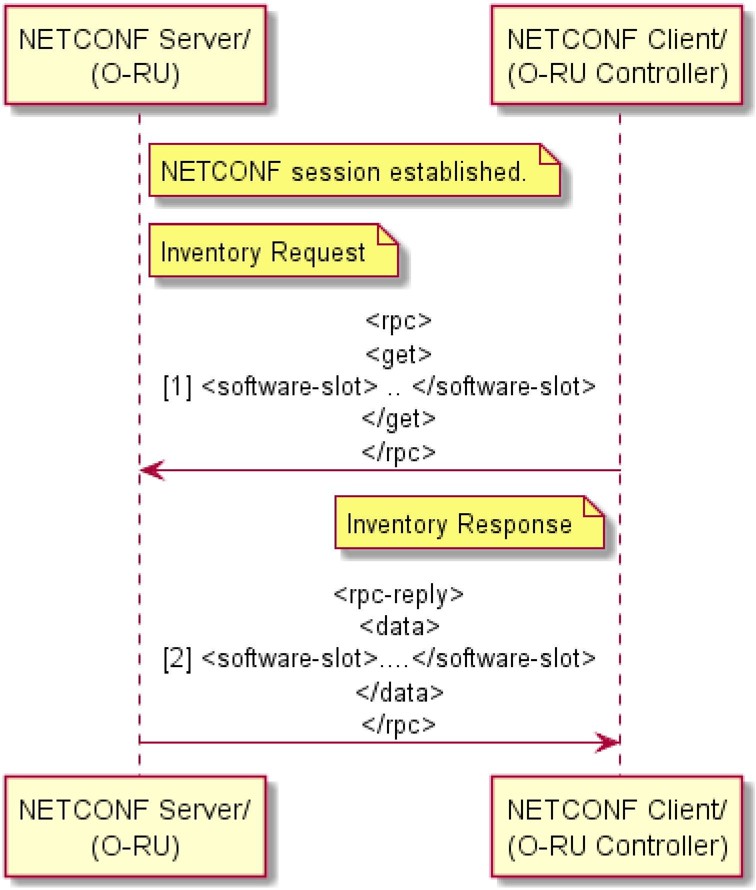


Figure 12: Inventory fetch call flow

Software Inventory is fetched by a NETCONF Client using the NETCONF **get** rpc filtered over the **software-slot** container. The response contains information about each software slot and its contents.

The following information is provided by software-inventory reply message:

a) **name** - name of the software slot (the name is defined by the O-RU vendor)

b) **status** - status of the software package. Status of the package can be

- **VALID** - Slot contains a software build considered as proven valid.

- **INVALID** - software build is not currently used by O-RU. The software is considered by the O-RU as damaged (e.g. wrong CRC).

Activation of a software slot containing an invalid software build shall be prohibited. Note: failed software install operation can cause a slot status to change to "Invalid".

- **EMPTY** - software slot does not contain a software package. Activation of an empty software slot shall be prohibited.

c) **active** - indicates if the software stored in particular slot is activated at the moment.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

- True - software slot contains an activated software build. Active::True can be assigned only for slots with status "Valid". At any time, only one slot in the O-RU MUST be marked as Active::True. The O-RU shall reject activation for software slots with status "Empty" and "Invalid".

- False - software slot contains passive software build or is empty

d) **running** - informs if software stored in particular slot is used at the moment.

- **True** - software slot contains the software build used by the O-RU in its current run.

- **False** - software slot contains a software build not used by O-RU at the moment.

e) **access** – informs about access rights for the current slot

- **READ_ONLY** – The slot is intended only for factory software, activation of such software slot means performing a factory reset operation and a return to factory defaults settings.

- **READ_WRITE** – slot used for updating software

f) **product-code** - product code provided by the vendor, specific to the product.

g) **vendor-code** - unique code of the vendor.

h) **build-id** - Identity associated with the software build. This id is used to find the appropriate build-version for the product consist of the vendor-code and the product-code.

i) **build-name** - Name of the software build.

j) **build-version** - Version of the software build for the product consist of the vendor-code and the product-code.

k) **files** – list of files in build

l) **name** – name of one particular file

m) **version** – version of the file

n) **local-path** - complete path of the file on local file system

o) **integrity** - result of the file integrity check

- **OK** – file integrity is correct

- **NOK** – file is corrupted

If a slot contains a file with integrity::NOK, the O-RU shall mark the whole slot with status::INVALID. The content of a software-slot is fully under O-RU's management - including removal of the content occupying the slot (in case the slot is subject of software update procedure), control of file system consistency and so on. The slot content shall not be removed until there is a need for new software to be installed.

Note: The empty slot parameters shall be as follows

name: up to vendor, not empty

status: "INVALID"

active: False

running: False

access: READ_WRITE

product-code: up to vendor

vendor code: up to vendor

build-name: null (empty string)

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

build-version: null (empty string)

files: empty

5.3 Download

Pre-condition:

- M-Plane NETCONF session established.
- O-RU Controller has subscribed to receive **download-event** notifications.

Post-condition:

- O-RU downloads all files specified and successfully stores the downloaded files in the O-RU's file system.

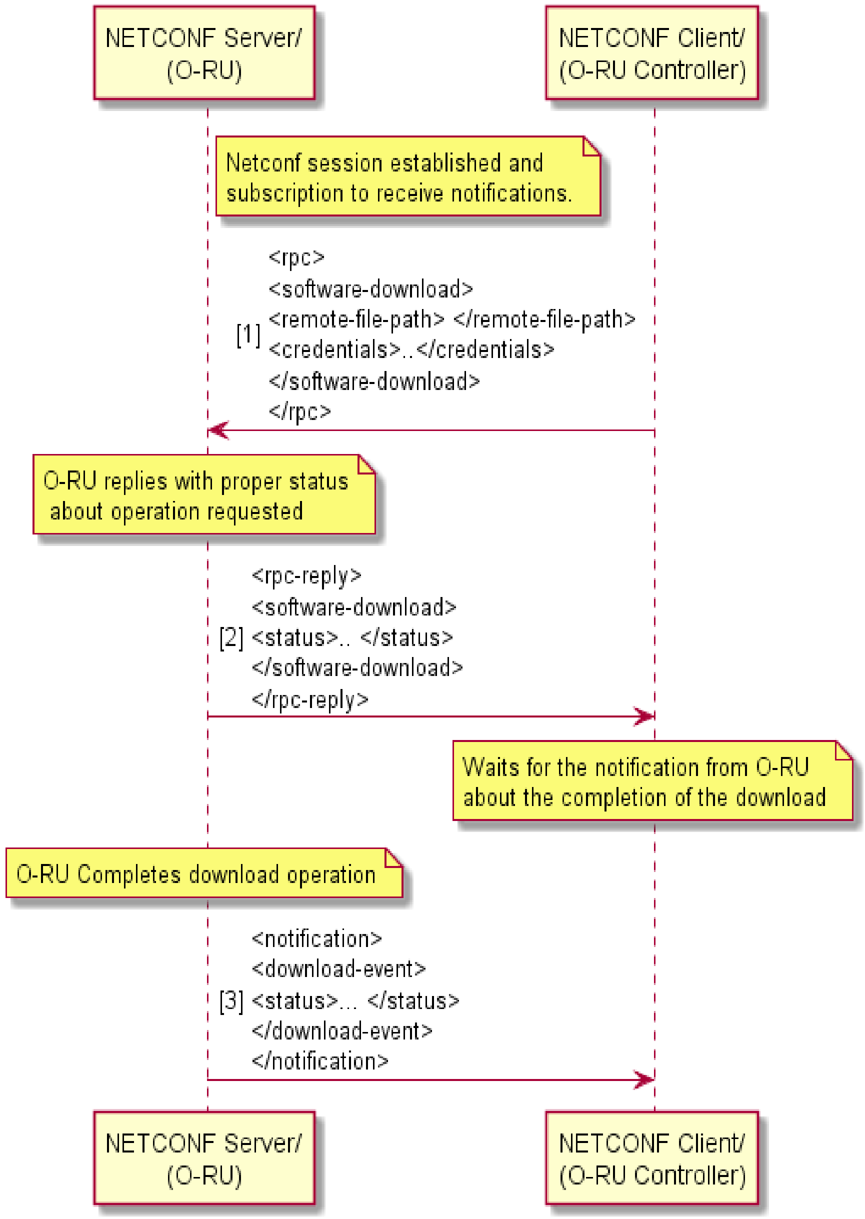


Figure 13: Software download call flow

Following types of authentications are supported for **software-download**:

a) password for RU authentication and list of public keys (DSA/RSA) for sFTP server authentication

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

b) certificate for both RU and sFTP server authentication

The **software-download** rpc is used to trigger the downloading of software to the O-RU. The download shall be performed using sFTP. The rpc specifies the URI of the remote location of the software files.

The O-RU shall send an immediate rpc-reply message with one of following statuses:

a) STARTED – software download operation has been started

b) FAILED – software download operation could not be proceeded, reason for failure in error-message

When the O-RU completes the software download or software download fails, the O-RU shall send NETCONF **download-event** notification with one of the following statuses:

a) COMPLETED

b) AUTHENTICATION_ERROR - source available, wrong credentials

c) PROTOCOL_ERROR – sFTP protocol error

d) FILE_NOT_FOUND - source not available

e) APPLICATION_ERROR - operation failed due to internal reason

f) TIMEOUT - source available, credentials OK, Operation timed out (e.g. source becomes unavailable during ongoing operation).

The above will be repeated until all files which are required for the O-RU and which belong to the software package have been downloaded to the O-RU.

5.4 Install

Pre-condition:

- M-Plane NETCONF session established.
- At least one software slot with status active::False and running::False exists in O-RU.
- Software Download has been completed successfully and files are available in O-RU.
- O-RU Controller has subscribed to receive **install-event** notifications.

Post-condition:

- O-RU software is installed in the specified target **software-slot**.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

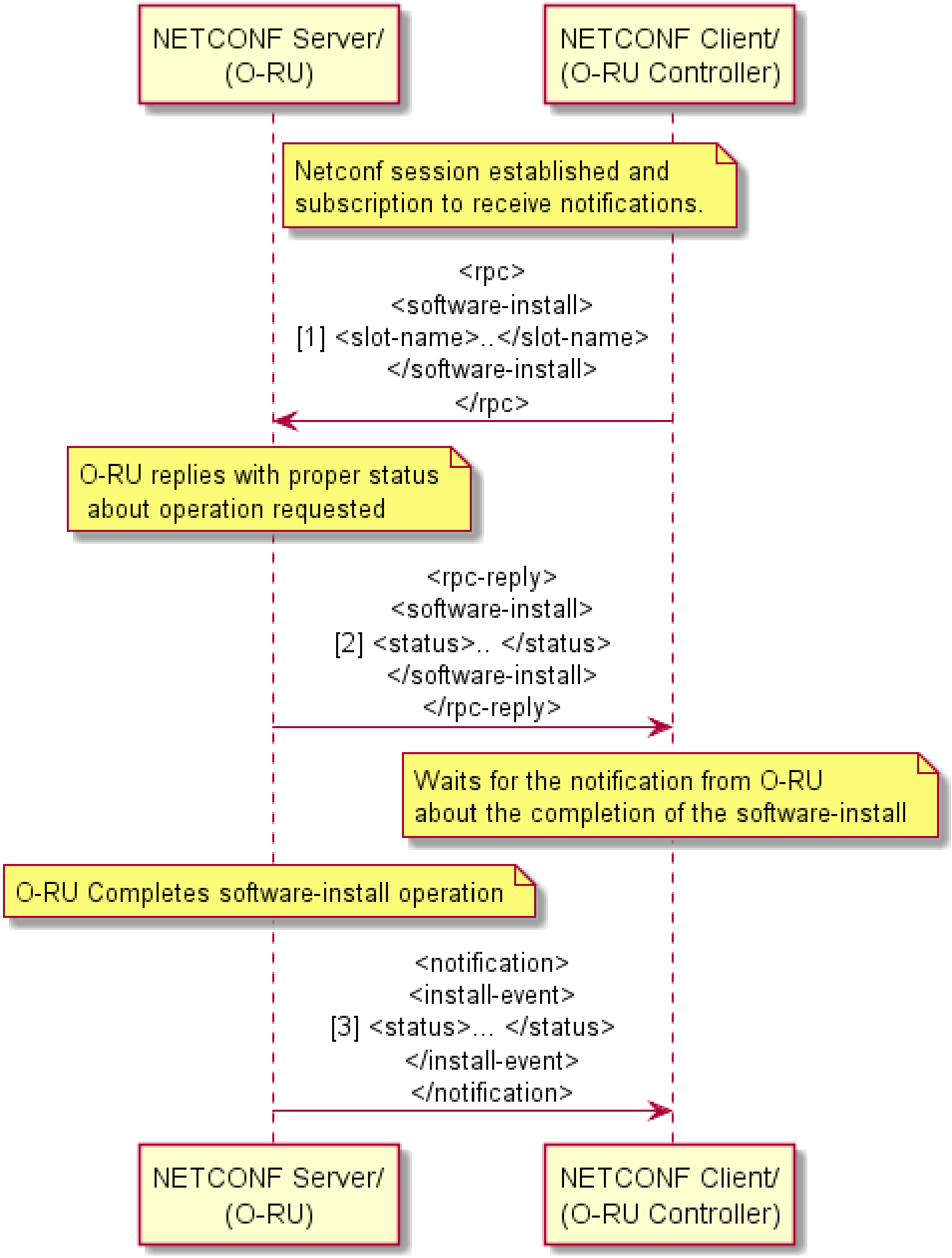


Figure 14 Software install call flow

NETCONF **software-install** rpc is used to install the previously downloaded software (all files provided in the package) to the specified target **software-slot** on O-RU**.** This slot must have status active::False and running::False.

The O-RU shall send an immediate rpc-reply message with one of following statuses:

a) STARTED – software install operation has been started.

b) FAILED – software install operation could not be proceeded, reason for failure in error-message.

When O-RU completes the software install or software install procedure fails, the O-RU will send NETCONF **install-event** notification with one of the following statuses:

a) COMPLETED - Install procedure is successfully completed.

b) FILE_ERROR – operation on the file resulted in in error, disk failure, not enough disk space, incompatible file format

c) INTEGRITY_ERROR – file is corrupted

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

d) APPLICATION_ERROR – operation failed due to internal reason

When the software install commences, the O-RU shall set the slot status to INVALID. After the install procedure finishes, the O-RU shall change the slot status to its appropriate status. This operation avoids reporting of inaccurate status when the install procedure is in operation or when it is interrupted (e.g., by spurious reset operation).

5.5 Activation

Pre-condition:

- M-Plane NETCONF session established.
- Software slot to be activated has status VALID.
- O-RU Controller has subscribed to receive **activation-event** notifications.

Post-condition:

- O-RU software activates to the version of software-slot.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

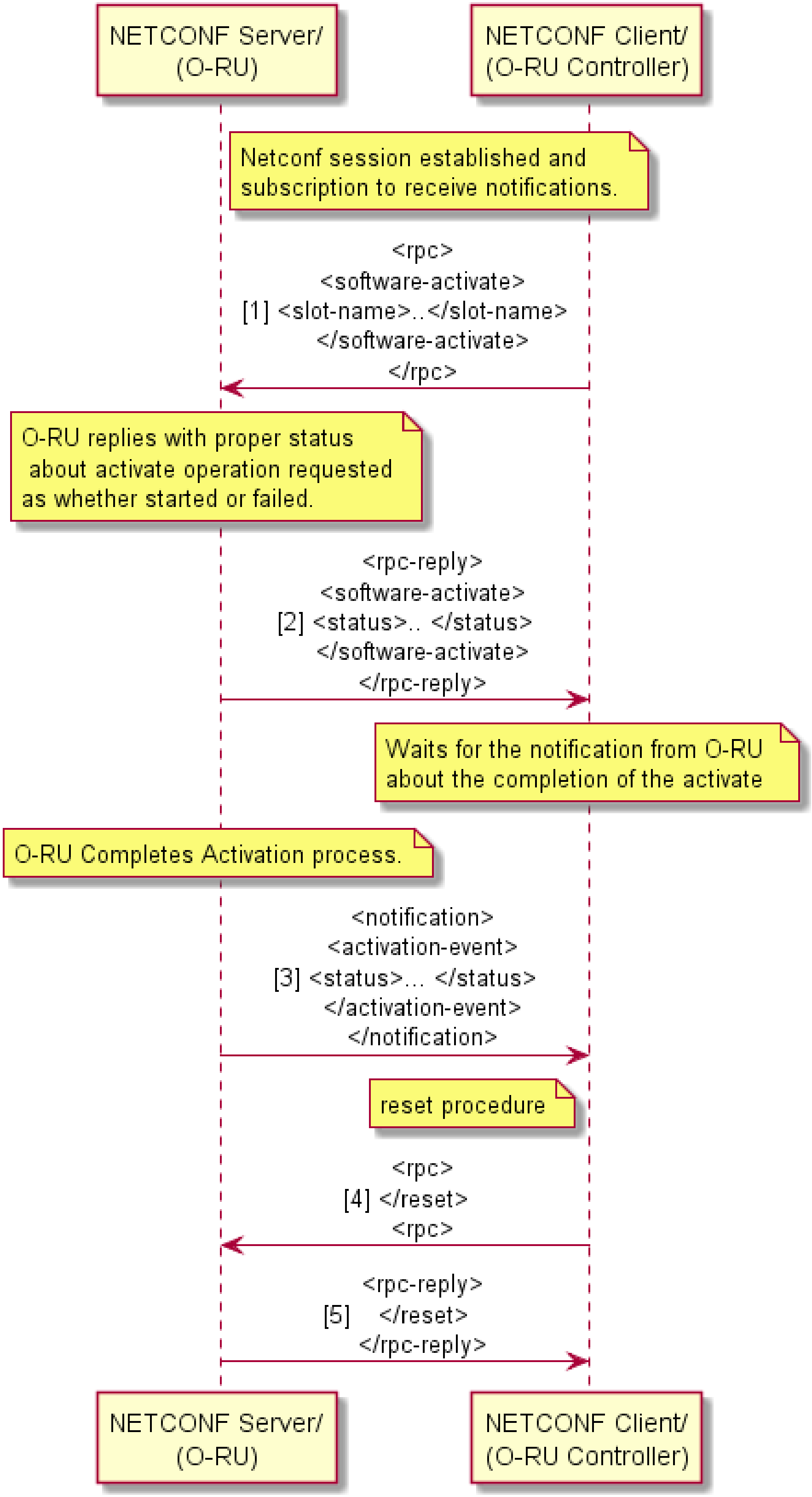


Figure 15 Software activation call flow.

NETCONF **software-activate** rpc is used to activate the software. The name of the software-slot is specified in the activate request.

The O-RU shall send an immediate rpc-reply message with one of following statuses:

a) STARTED – software activation operation has been started

b) FAILED – software activation operation could not be proceeded, reason for failure in error-message

When the activation is completed, the O-RU shall send the NETCONF **activation-event** notification with the status of activation. The following status is returned in the NETCONF **activation-event** notifications.

a) COMPLETED - Activation procedure is successfully completed. O-RU must be restarted via NETCONF **reset** rpc for the new software to be activated.

b) APPLICATION_ERROR - operation failed due to internal reason

Only one software slot can be active at any time. Thus, successful software-activate command will set active::True to the slot that was provided in the rpc and automatically will set active::False to the previously active slot.

NETCONF **reset** rpc shall be sent to O-RU to activate to the version of the software-slot. O-RU restarts and performs startup procedure as described in Chapter 3 as regular startup with new software version running.

5.6 Software update scenario

An example scenario of a successful software update procedure can be as follows:

1. NETCONF client performs a software inventory operation and identifies that an inactive and not-running slot for installing software is available so that it can download and install a software package.

   NOTE: This version of this specification does not distinguish between a software upgrade and a downgrade.

2. NETCONF client using the **software-download** rpc requests the O-RU to download a software package (if the software package contains several files, steps 2-4 need to be performed repeatedly until all files have been downloaded)
3. O-RU sends rpc response that download was started
4. O-RU finishes downloading the file(s) and reports this by sending the **download-event** notification
5. NETCONF client requests installation of the software using **software-install** rpc, and provides the slot name where the software needs to be installed along with a list of filenames to be installed (if the software package contains only one file, the list will contain only one entry)
6. O-RU sends rpc response that installation was started
7. O-RU sets installation slot status to INVALID
8. O-RU installs the software and after successful installation (with checksum control) changes status of the slot to VALID
9. O-RU notifies the NETCONF client that the installation is finished using **install-event** notification
10. NETCONF client requests the O-RU to activate the newly installed software using the **software-activate** rpc
11. O-RU sends rpc response that activation was started
12. For requested slot, O-RU changes active to True and at the same time sets activate to False on previously active slot
13. O-RU notifies NETCONF client about activation finished using the **activation-event** notification
14. NETCONF client restarts the O-RU forcing it to use the newly installed and activated software into use. O-RU restarts as regular startup with new software version running.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

5.7 Factory Reset

O-RU can be reset to the factory default software by activating the software-slot containing the factory default software and initiating NETCONF **reset** rpc. O-RU may clear persistent memory data during factory reset as vendor implementation option.

Chapter 6 Configuration Management

6.1 Baseline configuration

This chapter describes NETCONF standard operation (edit-config/get-config/get) [3] which belongs to the CM in Module to modify/retrieve any parameters in YANG modules. Examples below use o-ran-hardware as an example YANG module.

Two following scenarios are feasible for Configuration Management purposes.

- 2 phase (modify/commit) operation using writable running datastore
- 3 phase (modify/commit/confirm) operation using candidate datastore.

The 2-phase operation performs edit config on running datastore directly and confirm operation is not used.

All O-RUs shall support 2 phase operation, with 3 phase operations being optional.

6.1.1 Retrieve State

O-RU Controller is able to retrieve state which is defined in o-ran-hardware by using NETCONF <get> procedure:

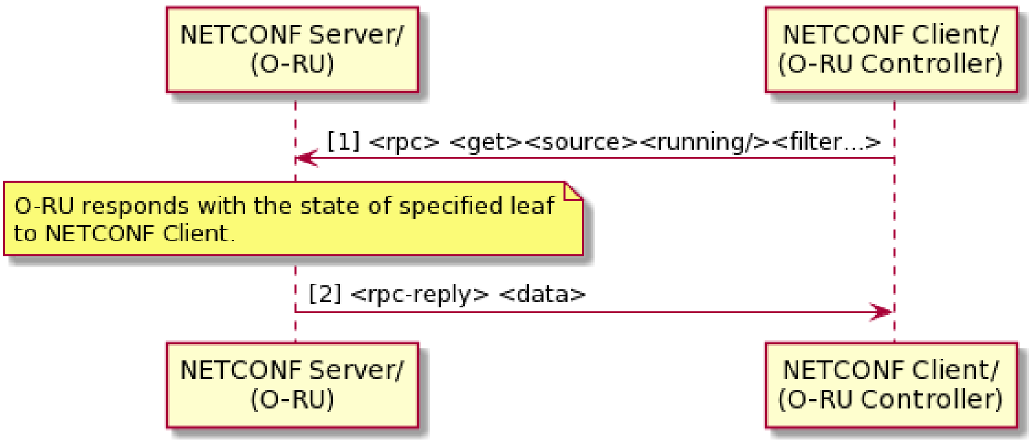


Figure 16 – Retrieve Resource State

Preconditions:

- O-RU Controller has completed exchange of NETCONF capabilities as part of connection establishment between O-RU and O-RU Controllers.

Post conditions:

- O-RU controller has retrieved O-RU state as per <get> request.

6.1.2 Modify State

O-RU Controller is able to change state which can be configurable by using NETCONF <edit-config> procedure without reset.

The configurable state are admin-state and power-state defined in o-ran-hardware.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

In case of a failure, an error will be returned. Please refer to RFC6241 Appendix A for error codes.

The vendor can define the behavior after the error occurred.

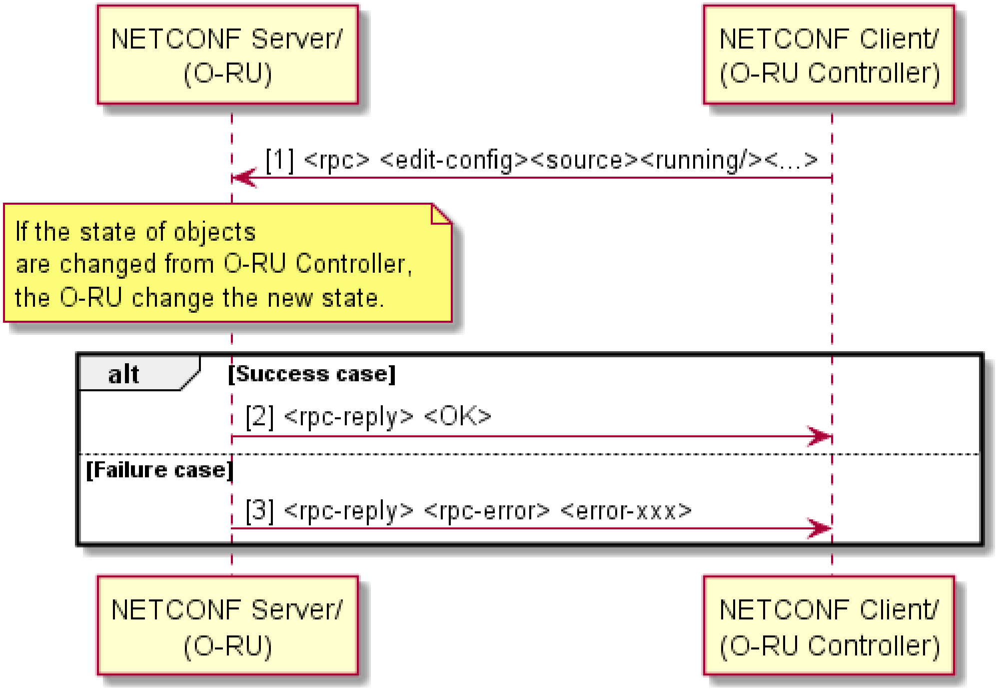


Figure 17 – Modify Resource State without reset

The followings are the information of state transition for each state.

[admin-state]

The admin-state transition diagram for the O-RU is illustrated in Figure 18 below.

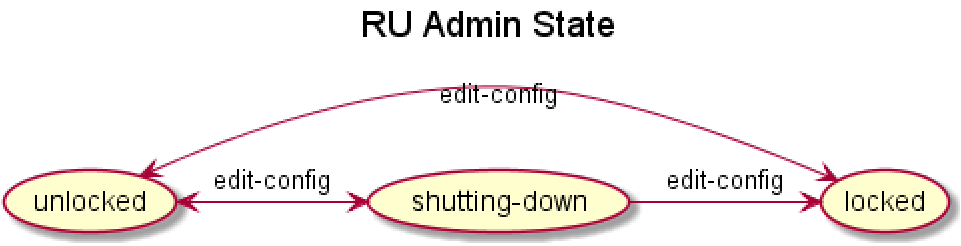


Figure 18 – Admin State

- locked: This state indicates that any resource activation is prohibited for the O-RU and all resources have been deactivated administratively.
- shutting-down: That usage is administratively limited to current instances of use. It's not mandatory (will be optional).
- unlocked: This state indicates that any resource activation is allowed and any resources can be active. The state "unlocked" is the initial state after the reset of the O-RU.

[power-state]

The power-state transition diagram for O-RU is presented in Figure 19 below. This state can be controlled by editing the parameter energy-saving-enabled.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Figure 19 – Power State

- awake: This state indicates that the O-RU is operating normally, i.e. not in energy saving mode. The state "awake" is the initial state after the reset of the O-RU.
- sleeping: This state indicates that the O-RU is in energy saving mode. M-plane connection and functions are alive whereas other C/U/S functions may be stopped to reduce energy consumption. This state is optional.

[oper-state]

O-RU Controller is able to change oper-state defined in o-ran-hardware of the O-RU by using remote procedure call **reset**. In this case, the O-RU responds <rpc-reply><ok> prior to reset operation. Whatever the previous state is, the O-RU oper-state starts from disabled when O-RU receives **reset**.

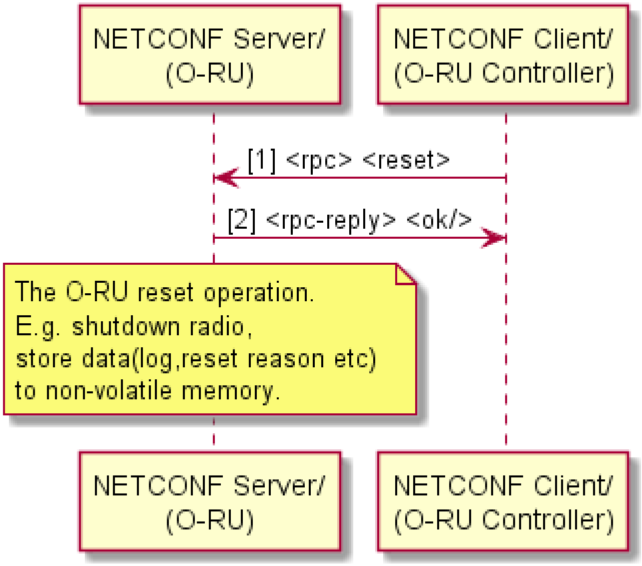


Figure 20 – Modify Oper State (reset)

The oper-state transition diagram for O-RU is presented in Figure 21 below.

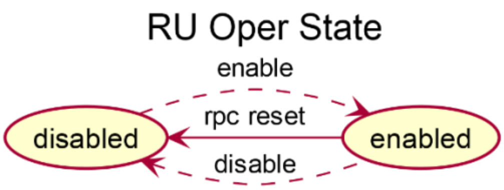


Figure 21 – Oper State

enabled: O-RU is partially or fully operational.

disabled: O-RU is not operational. This is the initial state of oper-sate after the reset of the O-RU

- O-RU Controller is able to reset the O-RU, even if the O-RU state is "disabled" or "enabled".

[availability-state]

The availability-state transition diagram for the O-RU is presented in Figure 22 below.

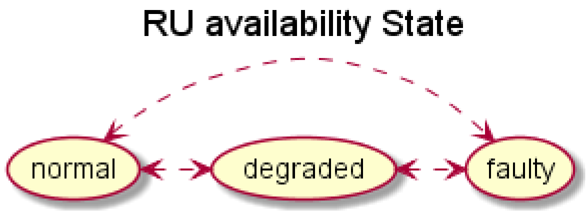


Figure 22: Availability State

The availability state is derived from detected and active faults and their impact to O-RU's operation. The availability state is not affected by faults caused by external reasons.

- normal: There is no fault.
- degraded: When major or critical fault affecting module or any of O-RU's subcomponents (e.g. transmitter) is active.
- faulty: The critical fault affecting whole O-RU is active and O-RU can't continue any services.

[usage-state]

The usage-state transition diagram for the O-RU is presented in Figure 23 below.

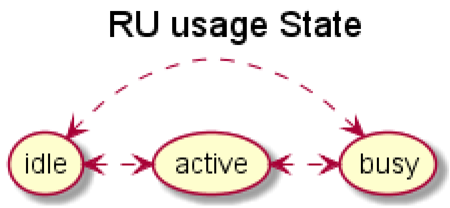


Figure 23 – usage State

idle: No carrier is configured in O-RU.

active: The carrier(s) is(are) configured in O-RU.

busy: No more carrier can be configured in O-RU.

6.1.3 Retrieve Parameters

O-RU Controller is able to retrieve parameters of the YANG module by using NETCONF <get> or <get-config> procedure

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

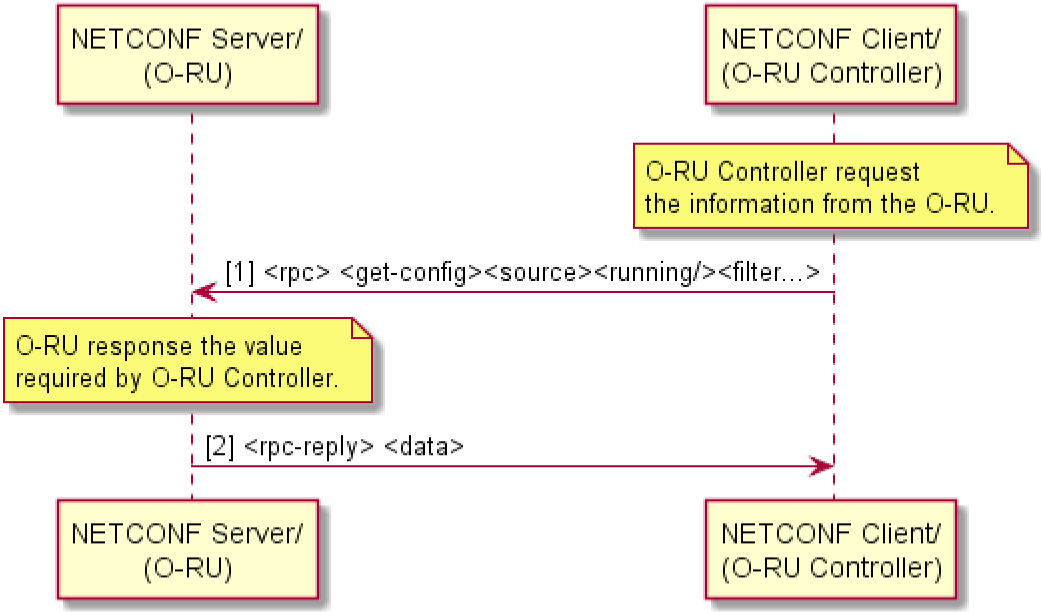


Figure 24 – Retrieve Parameters

Preconditions:

- O-RU Controller has completed exchange of NETCONF capabilities as part of connection establishment between O-RU and O-RU Controller(s).

Post conditions:

- O-RU controller has retrieved O-RU parameters as per <get><source><running/><filter> or <get-config><source><running/><filter> request.

6.1.4 Modify Parameters

Before an O-RU Controller modifies the configuration (candidate or running) of an O-RU, it shall first lock the target configuration. This prevents other NETCONF clients from changing the shared configuration database until the O-RU Controller releases the lock. If another NETCONF client has already locked the configuration datastore, then the O-RU will respond with a NETCONF error indicating that the requested lock is denied. Note, in such circumstances, the O-RU controller should wait for a period of time before re-attempting to modify the O-RU's configuration.

O-RU Controller is able to modify parameters of the YANG module by using the NETCONF <edit-config> procedure.

When supported by an O-RU, the O-RU Controller shall perform any required modify operations **ONLY** on the candidate configuration datastore before committing the validated configuration to the running configuration datastore. When an O-RU does not support the candidate configuration datastore, the O-RU Controller should take extreme care whenever modifying the running configuration datastore as such will likely impact system operation.

Note, validation of the modified configuration is based on:

1) basic YANG constraints (e.g., min-elements, range, pattern),

2) XPATH based YANG constraints (e.g., leafref, must and when statements), and

3) external code which implements YANG constraints (e.g., defined in O-RAN specifications, YANG description statements, etc.).

In case of failures, an error will be returned. Please refer to RFC6241 Appendix A for error codes.

The vendor can define the behavior after error occurred.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

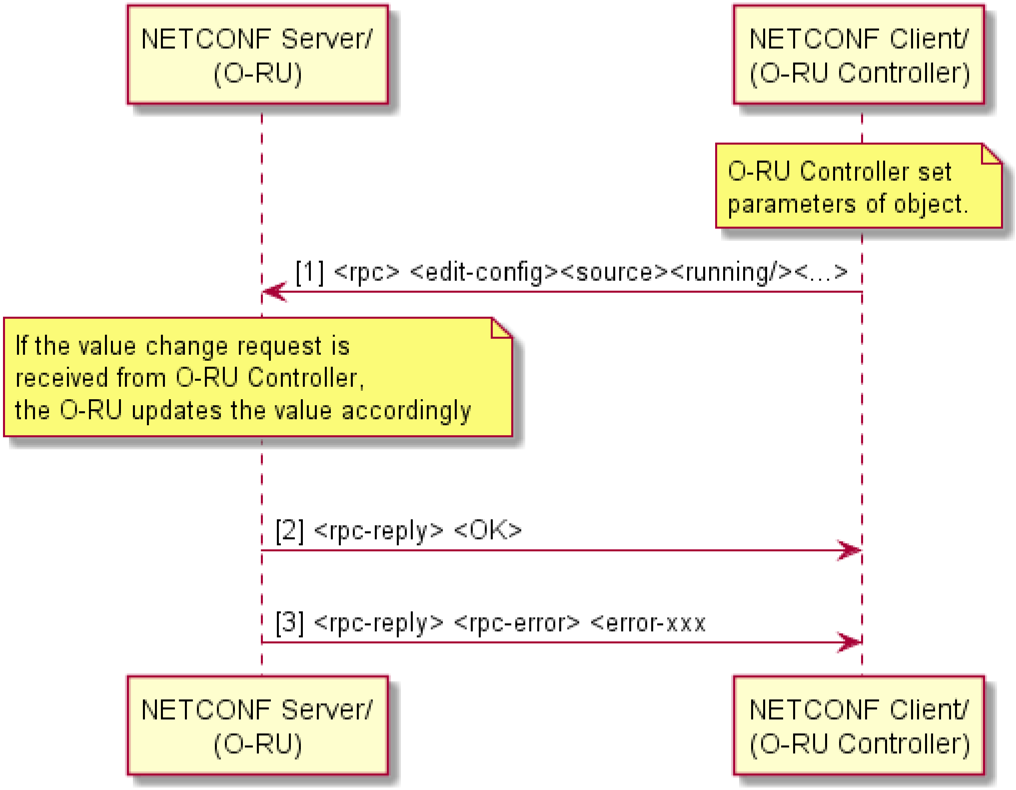


Figure 25 – Modify Parameters

Preconditions:

- O-RU Controller has completed exchange of NETCONF capabilities as part of connection establishment between the O-RU and O-RU Controller(s).

- O-RU Controller has locked the target configuration

Post conditions:

- O-RU controller has retrieved O-RU resource state as per <edit-config> request

- Success case: The update is confirmed to O-RU Controller.
- Failure case: Failure reason is provided to O-RU Controller

Sequential processing is assumed. Only a single <edit-config> rpc is allowed at a time. Next <edit-config> rpc shall be performed after previous <edit-config> rpc reply.

Modify Parameters is used to:

- update parameters of existing leafs,

O-RU shall be allowed to reject <edit-config> in case the content is found to be against e.g., functions supported by O-RU - like carrier configured out of band.

After the modification procedure is complete, the O-RU Controller releases the lock on the target configuration.

6.2 Framework for optional feature handling

This section describes the common and optional features about Configuration Management.

An O-RU may have some features which are not supported by other O-RUs, i.e. optional feature(s). In this case, the O-RU needs to inform the O-RU controller which features the O-RU can provide, and this can be achieved by exchanging NETCONF capabilities.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Some of the YANG models are optional for the O-RU to support. For example, in this version of the management plane specification, those models associated with External IO and Antenna Line Devices are not essential for the operation of the O-RAN fronthaul interface. Other mandatory models define optional feature capabilities.

The NETCONF Server shall use the ietf-yang-library model (RFC 7895) [20] to list the namespace of the models supported by the Server. If an O-RU/NETCONF server does not return the namespace associated with an optional YANG model, the NETCONF client determines that the O-RU does not support the optional capability associated with the model.

In addition, for each supported schema, the ietf-yang-library lists the YANG feature names from this module that are supported by the server. The details of optional models and features are defined in Annex C.

6.1.5 Deleting Parameters

Before an O-RU Controller deletes any configuration (candidate or running) of an O-RU, it shall first lock the target configuration. This prevents other NETCONF clients from changing the shared configuration database until the O-RU Controller releases the lock. If another NETCONF client has already locked the configuration datastore, then the O-RU will respond with a NETCONF error indicating that the requested lock is denied. Note, in such circumstances, the O-RU controller should wait for a period of time before re-attempting to delete the O-RU's configuration.

O-RU Controller is able to delete parameters of the YANG module by using the NETCONF <edit-config> procedure with the "operation" attribute set to delete.

If the configuration data does not exist, an <rpc-error> element is returned with an <error-tag> value of "data-missing".

When supported by an O-RU, the O-RU Controller shall perform any required delete operations **ONLY** on the candidate configuration datastore before committing the validated configuration to the running configuration datastore. When an O-RU does not support the candidate configuration datastore, the O-RU Controller should take extreme care whenever modifying the running configuration datastore as such will likely impact system operation.

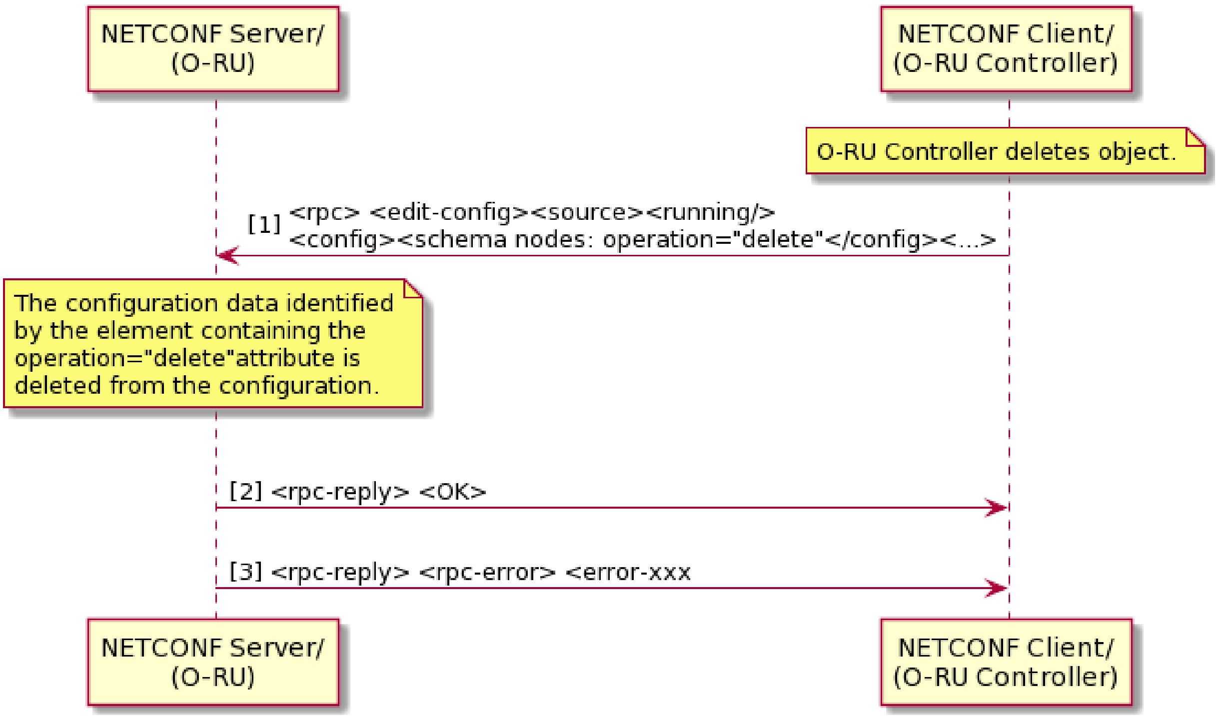


Figure 26 – Delete Parameters

Preconditions:

- O-RU Controller has completed exchange of NETCONF capabilities as part of connection establishment between the O-RU and O-RU Controller(s).

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

- O-RU Controller has locked the target configuration

Post conditions:

- O-RU controller has updated the O-RU resource state as per <edit-config> request

- Success case: The delete is confirmed to O-RU Controller.
- Failure case: Failure reason is provided to O-RU Controller

Sequential processing is assumed. Only a single <edit-config> rpc is allowed at a time. Next <edit-config> rpc shall be performed after previous <edit-config> rpc reply.

Delete Parameters is used to:

- delete parameters of existing configuration

After the delete procedure is complete, the O-RU Controller releases the lock on the target configuration.

6.3 M-Plane Operational State

The o-ran-mplane-int YANG model allows the O-RU to report the connectivity to NETCONF clients on a per sub-interface level. The client information includes the IP address(es) for the client(s) as well as the link-layer address used to forward packets towards the various management plane clients.

6.4 Notification of Updates to Configuration Datastore

6.4.1 Introduction

This sub-section defines an optional O-RU capability which allows O-RU Controllers to configure the O-RU to provide notifications of modifications to its YANG datastore. This capability can be used when the O-RU is operating in a hybrid environment with multiple simultaneous NETCONF sessions established to different O-RU controllers. Using this capability, one particular O-RU controller uses the NETCONF notifications functionality specified in RFC 6470 [35] to enable it to be automatically signalled changes to the O-RU's configuration made by a second O-RU controller. Additionally, if the O-RU supports a vendor specific interface to allow manual configuration, this functionality can also be used to signal such configuration modifications to an O-RU Controller.

6.4.2 Subscribing to updates from an O-RU

When an O-DU receives an indication from an O-RU that it supports the optional capability to support notification of updates to its configuration data store, as a minimum, it shall subscribe to the **netconf-config-change** notification.

An example subscription to the event notification stream is shown in Figure 27, where one O-RU Controller is receiving a notification that the configuration of the O-RU's timezone offset has been modified by a second O-RU controller.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2020-03-01T08:00:14.12Z</eventTime>
  <netconf-config-change xmlns= "urn:ietf:params:xml:ns:yang:ietf-netconf-notifications">
    <id>102</id>
    <changed-by>
      <username>nms-user</username>
      <session-id>1099</session-id>
      <source-host>10.10.10.10</source-host>
    </changed-by>
    <datastore>running</datastore>
    <edit>
      <target>/oran-ops:operational-info/oran-ops:clock/oran-ops:timezoneutc-offset</target>
      <operation>replace</operation>
    </edit>
  </netconf-config-change>
</notification>
```

Figure 27 – Example of a netconf-config-change notification

Chapter 7 Performance Management

This chapter provides the description of scenarios related to performance management. It consists of 2 functions. One is for the measurement activation and the other is the collection of measurement results.

7.1 Measurement Activation and De-activation

The measurement activation at the start-up installation is also allowed as described in chapter 3.

Pre-condition:

M-Plane is operational.

Post-condition:

Measurement is activated or deactivated as per NETCONF client's request.

This sub-section provides information about how to activate and de-activate the performance measurement via NETCONF <**edit-config**> to O-RU. The performance measurement is defined as o-ran-performance-management YANG module. In case of multiple NETCONF clients, only one NETCONF client shall activate/deactivate the measurements in the O-RU.

In the performance-management YANG module, the following parameters are defined.

- group of the measurement results, e.g., **transceiver-measurement-objects**, **rx-window-measurement-objects, tx-measurement-objects** and **epe-measurement-objects**.
- **measurement-interval**: measurement interval for the **measurement-object**s to measure the performance periodically, e.g., 300, 600, 900 seconds. It is defined per the group of the measurement result.
- **measurement-object**: target metric to measure the performance, e.g., RX_POWER, TX_POWER, defined as key parameter.
- **active**: enable/disable the performance measurement per **measurement-object.** This value is Boolean. Default is FALSE.
- **start-time** and **end-time**: to report the time of measurement start and end for the **measurement-object** at each **measurement-interval**.
- **object-unit**: unit to measure the performance per object, e.g., O-RU, physical port number, antenna, carrier. The **object-unit** may be configurable Identifier **object-unit-id** means e.g., physical port number when object unit set to physical port number.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

- **report-info**: the reporting info to the **measurement-object**, e.g., MAXIMUM, MINIMUM, FIRST, LATEST, FREQUENCY_TABLE and COUNT. Multiple info can be considered for one object if necessary.
- Optional configurable parameter(s) for **report-info**: some configurable parameters to report, e.g., **function**, **bin-count**, **upper-bound**, **lower-bound**. For the **bin-count** configuration, it shall be less than the parameter **max-bin-count** that is the capability information of NETCONF server for the maximum configurable value for bin-count.
- Additional reporting information for **report-info**: some additional information to report info, e.g., date-and-time.

The detail of the parameters per **measurement-object** and the group of measurement result are defined in Annex B.

The **measurement-interval** of **measurement-object** may be set to common or different values per group of the measurement result.

It is allowed that the measurement is activated and deactivated at any time. When different parameter measurements have intervals with a common factor, the O-RU shall synchronize the boundary of these measurements aligned with this factor, irrespective of when the different measurements are activated. And all of start points of the **measurement-interval**s shall be synchronized to zero o'clock mid-night by using an equation {full seconds (hour, minute and second) modulo '**measurement-interval**' = 0}, in order to ensure the same start and end of the **measurement-interval**s between O-RUs. For more detail see the following illustration.

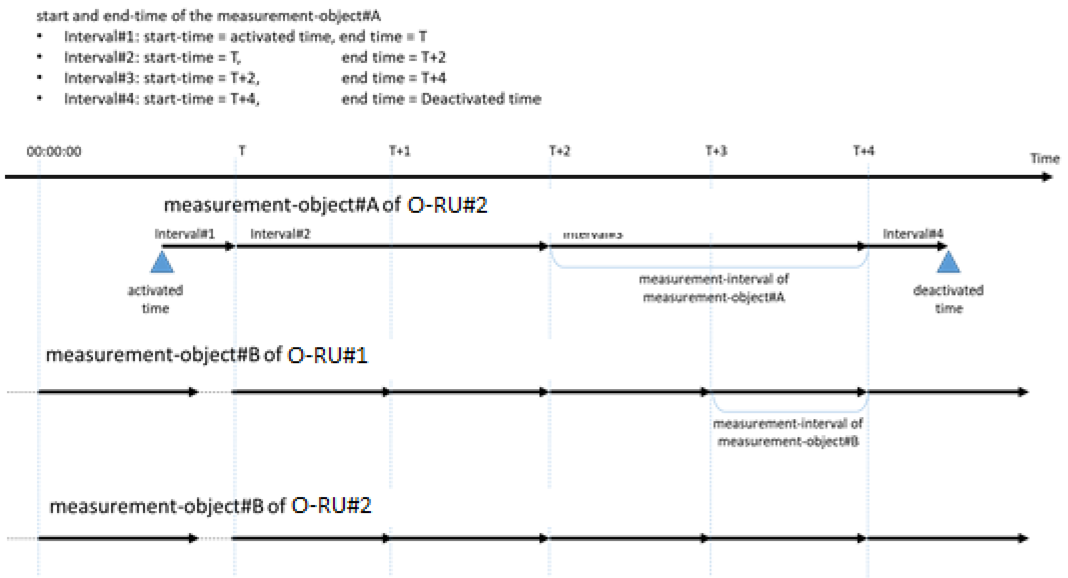


Figure 28: synchronization of measurement-interval.

The modification of the configurable parameters for the measurement shall be allowed while **active** for the corresponding **measurement-object** is set to FALSE.

All of the measurements are optionally supported in the O-RU.

The **report-info**, e.g., **count**, shall be started from 0 at the boundary of every **measurement-interval**. No accumulation is applied between the **measurement-interval**s.

7.2 Collection and Reporting of Measurement Result

This sub-section provides the description of scenarios used to collect measurement results. There are three options.

1. NETCONF process: Create-subscription from NETCONF client and NETCONF notification from NETCONF server are used.
2. File Management process: File upload mechanism is used for the measurement file from O-RU to configured file server(s) that O-RU can reach to.
3. Configured subscription process: Create configured subscription from O-RU as Event-Producer to Event-Collector.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Methods 1 and 2 are mandatory for the O-RU. Method 3 shall be supported by those O-RUs that support the optional NON-PERSISTENT-MPLANE feature. The method(s) to be used is the matter of NETCONF client.

In case of multiple NETCONF clients and/or Event-Collectors, the O-RU shall report the same notification-based measurement results to all subscribed NETCONF clients/Event-Collectors, and the O-RU shall upload file-based results to all configured fileservers.

7.2.1 NETCONF process

This process needs the NETCONF capability: urn:ietf:params:netconf:capability:notification:1.0

1. NETCONF client subscribes to one or more measurement group(s) and/or **measurement-object**(s) to collect the measurement result by sending NETCONF <**subscribe-notification**> to NETCONF server in the O-RU. In this message, startTime and stopTime for the notification may be configurable. NETCONF client can configure the **notification-interval** in the performance-measurement YANG module.

2. NETCONF server sends NETCONF notification messages periodically to the client as configured by the **notification-interval**. The NETCONF notification message contains subscribed measurement group(s) and/or **measurement-object**(s). The **notification-interval** doesn't need to be same as the **measurement-interval**. The notification timing different from the **measurement-interval** is a matter to O-RU implementation.

This procedure is described in the following figure.

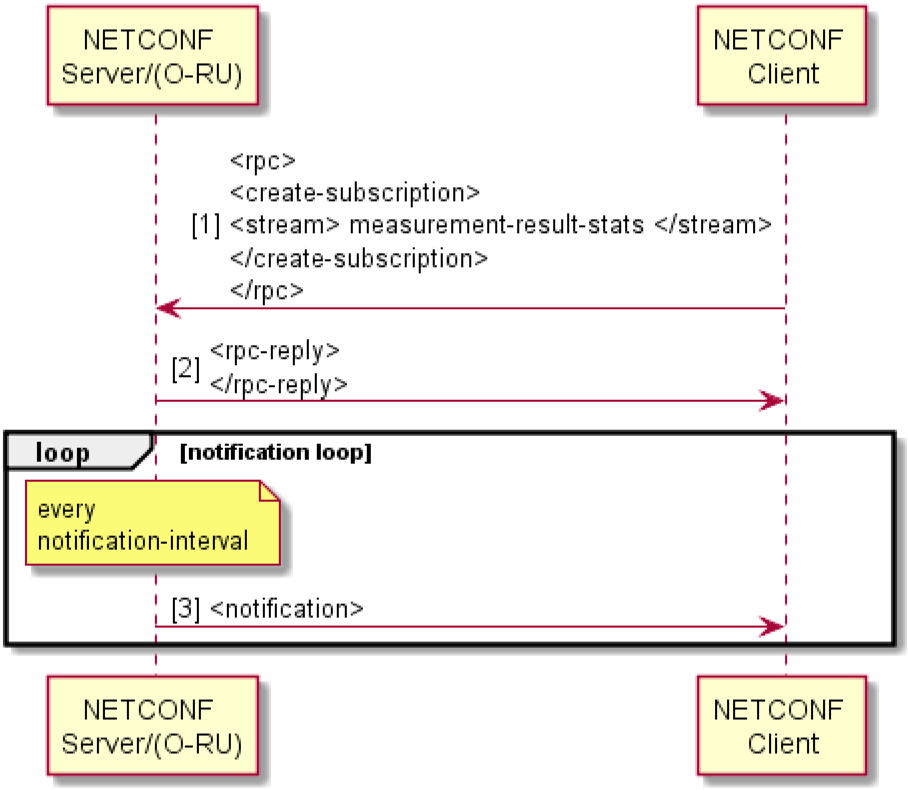


Figure 29: NETCONF process of Measurement Result Collection

Note: This figure uses **create-subscription** for the single stream "**measurement-result-stats**". In order to subscribe multiple notifications, the appropriate **create-subscription** message is required. Please refer to section 8.2 for the appropriate example of **create-subscription** of multiple notifications.

In order to terminate the subscription, the NETCONF client shall send <**close-session**> operation from the subscription session. If NETCONF session is terminated by <**kill-session**>, the subscribed notification is terminated as well.

This procedure is described in the following figure.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

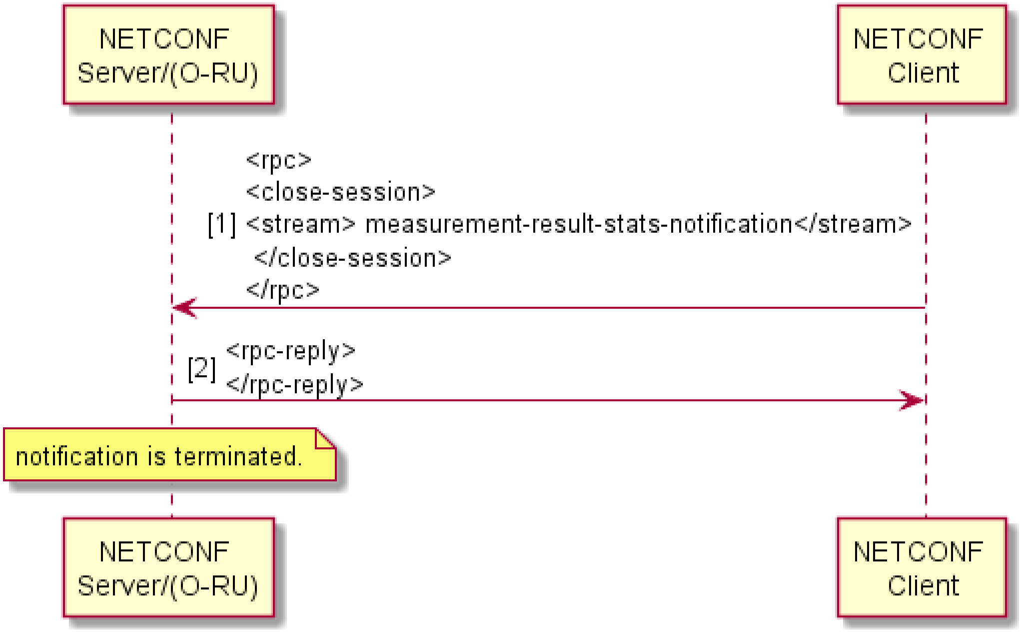


Figure 30: NETCONF process of Measurement Result Collection to end

When **notification-interval** is larger than **measurement-interval**, one notification may contain multiple stats which have consecutive periods indicating **start-time** and **end-time** for the measurement.

When **notification-interval** is smaller than **measurement-interval**, one notification may not contain the stats which **start-time** and **end-time** are not applicable to the period for the notification.

For example, when **notification-interval** = 60min, **measurement-interval** for **measurement-object**#A= 30min and **measurement-interval** for **measurement-object**#B = 15min, one notification contains 2 measurement results for **measurement-object**#A with consecutive **start-time** and **end-time**, and 4 measurement results for **measurement-object**#B with consecutive **start-time** and **end-time**.

For the other example, when **notification-interval** = 15min, **measurement-interval** for **measurement-object**#A= 30min and **measurement-interval** for **measurement-object**#B = 15min, one notification contains one measurement results for **measurement-object**#B but not for **measurement-object**#A. next notification contains both measurements result for **measurement-object**#A and #B.

7.2.2 File Management process

NETCONF client needs to configure a parameter of performance measurement YANG module '**enable-SFTP-upload**' to enable or disable the periodic file upload mechanism via NETCONF <**edit-config**>. Its default is FALSE.

In addition, the performance measurement YANG module defines **file-upload-interval**, **remote-SFTP-upload-path**, **credentials** information of the file server and **enable-random-file-upload** as configurable parameters.

Following types of authentications are supported for **performance file upload**:

a) Password for RU authentication and list of public keys (DSA/RSA) for sFTP server authentication

b) Certificate for both RU and sFTP server authentication

When the parameter **enable-SFTP-upload** is set to TRUE, O-RU shall store the performance measurement files in the generic folder in O-RU, i.e., O-RAN/PM/. Every **file-upload-interval**, O-RU pushes the latest file to upload to the **remote-SFTP-upload-path** of the configured SFTP servers if **enable-SFTP-upload** is set to TRUE. Otherwise, the performance measurement file is not created and uploaded. The number of maximum performance files to be stored in O-RU simultaneously is a matter for O-RU implementation. The O-RU shall manage its own storage space by deleting the older files autonomously.

The O-RU shall ensure that the **start-time** and the **end-time** within the name of the performance measurement file are synchronized with the same manner as **measurement-interval** by using **file-upload-interval**.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

If the parameter **enable-random-file-upload** is set to TRUE, the O-RU shall randomize the timing to upload SFTP file after the performance measurement file is ready to upload. The randomized timing is an O-RU implementation matter and shall not be later than next **file-upload-interval**.

The file name of the performance measurement is:

C<**start-time**>_<**end-time**>_<**name**>.csv

- Starting with a capital letter "C".
- Format of <**start-time**> and <**end-time**> can be local time or UTC.

  Local time format is YYYYMMDDHHMM+HHMM, indicating, year, month, day, hour, minute, timezone "+" or "-", hour and minute for the time zone.

  UTC format is YYYYMMDDHHMMZ, indicating, year, month, day, hour, minute and with a special UTC designator ("Z")

  Time zone offset is provided by **timezone-utc-offset** in o-ran-operation.yang.
- <**name**> in ietf-hardware is used
- "_" underscore is located between <**start-time**>, <**end-time**> and <**name**>
- File extension is "csv" as csv format file.

Example of measurement file is:

C201805181300+0900_201805181330+0900_ABC0123456.csv.

The file format of the performance measurement has following rule:

1. Each line starts with the **measurement-object** identifier, which measurement can be switched to TRUE or FALSE by **active** parameter. The identifier of each **measurement-object** is defined in Annex B.
2. After the **measurement-object** identifier, the name of **measurement-object**, **start-time**, **end-time** are followed.
3. Since the **report-info** results of any **measurement-object** are measured per **object-unit**, **object-unit-id** and set of **report-info** are repeated in one line.
4. When multiple **report-info** parameters exist per **object-unit**, all of the **report-info** are consecutively listed until the next **object-unit-id**. The order of parameters, such as object-unit-id, report-info and additional information for the report-info, shall be same as the order of those listed in NETCONF notification defined in o-ran-performance-management YANG module.

Example of measurement result in one line is:

1, RX_ON_TIME, 2018-05-18T13:00:00+09:00, 2018-05-18T13:30:00+09:00, 0, 123, AAAA, 1, 123, BBBB, 2, 123, CCCC, 3, 123, DDDD

- Measurement-object-identifier: 1
- Name of **measurement-object**: RX_ON_TIME
- **start-time**: 2018-05-18T13:00:00+09:00 as measurement **start-time**.
- **end-time**: 2018-05-18T13:30:00+09:00 as measurement **end-time**
- EAXC_ID: 0
- Count for EAXC_ID#0 : 123
- **name** of **transport-flow** information: AAAA
- :

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

- EAXC_ID: 3
- Count for EAXC_ID#3 : 123
- **name** of **transport-flow** information: DDDD

When **file-upload-interval** is larger than **measurement-interval**, one performance measurement file may contain multiple lines for the stats which have consecutive periods indicating **start-time** and **end-time** for the measurement.

When **file-upload-interval** is smaller than **measurement-interval**, one performance measurement file may not contain the line for the stats which **start-time** and **end-time** are not applicable to the period for the performance measurement file.

For example, when **file-upload-interval** = 60min, **measurement-interval** for **measurement-object**#A= 30min and **measurement-interval** for **measurement-object**#B = 15min, one performance measurement file contains 2 measurement result lines for **measurement-object**#A with consecutive **start-time** and **end-time**, and 4 measurement result lines for **measurement-object**#B with consecutive **start-time** and **end-time** as followings:

1, RX_POWER, 2018-05-18T13:00:00+09:00, 2018-05-18T13:15:00+09:00, 0, 123

1, RX_POWER, 2018-05-18T13:15:00+09:00, 2018-05-18T13:30:00+09:00, 0, 123

1, RX_ON_TIME, 2018-05-18T13:00:00+09:00, 2018-05-18T13:30:00+09:00, 0, 123, AAAA, 1, 123, BBBB, 2, 123, CCCC, 3, 123, DDDD

1, RX_POWER, 2018-05-18T13:30:00+09:00, 2018-05-18T13:45:00+09:00, 0, 123

1, RX_POWER, 2018-05-18T13:45:00+09:00, 2018-05-18T14:00:00+09:00, 0, 123

1, RX_ON_TIME, 2018-05-18T13:30:00+09:00, 2018-05-18T14:00:00+09:00, 0, 123, AAAA, 1, 123, BBBB, 2, 123, CCCC, 3, 123, DDDD

For the other example, when **file-upload-interval** = 15min, **measurement-interval** for **measurement-object**#A= 30min and **measurement-interval** for **measurement-object**#B = 15min, one performance measurement file contains one measurement result line for **measurement-object**#B but not for **measurement-object**#A. next performance measurement file contains both measurements result for **measurement-object**#A and #B as follows.

C201805181300Z+0900_201805181315+0900_ABC0123456.csv.

1, RX_POWER, 2018-05-18T13:00:00+09:00, 2018-05-18T13:15:00+09:00, 0, 123

C201805181315Z+0900_201805181330+0900_ABC0123456.csv.

1, RX_POWER, 2018-05-18T13:15:00+09:00, 2018-05-18T13:30:00+09:00, 0, 123

1, RX_ON_TIME, 2018-05-18T13:00:00+09:00, 2018-05-18T13:30:00+09:00, 0, 123, AAAA, 1, 123, BBBB, 2, 123, CCCC, 3, 123, DDDD

The performance measurement files stored in O-RAN/PM can be uploaded on-demand. For the file upload mechanism by on-demand way, **retrieve-file-list** and **file-upload** operations are used. For more detail, please refer to Chapter 9.

7.2.3 Configured Subscription Process

This optional process requires the O-RU to support configured subscriptions, as described in Chapter 15. The structure of the process follows the NETCONF process described in sub-section 7.2.1. However, instead of sending a NETCONF <create-subscription > to the NETCONF server in the O-RU to subscribe to the **measurement-result-stats** notifications, the NETCONF client installs the subscription via configuration of the O-RU's datastore. Based on configured subscriptions, the O-RU sends asynchronous YANG notifications over HTTPS to the configured Event-Collector.

In order to terminate the subscription, the NETCONF client shall delete the corresponding configuration in the O-RU. Immediately after the subscription is successfully deleted, the O-RU will send to a subscription state change notification indicating that the subscription has ended to the Event-Collector.

Note, unlike the NETCONF process described in section 7.2.1, the subscription to the subscribed notifications is not terminated when the NETCONF session used to establish the subscription is terminated.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Chapter 8 Fault Management

Fault management is responsible for sending alarm notifications to the configured subscriber, which will typically be the NETCONF Client unless the O-RU supports the configured subscription capability, as described in Chapter 15, when the configured subscriber may be an Event-Collector. FM contains Fault Management Managed Element and via this Managed Element alarm notifications can be disabled or enabled.

The NETCONF Server is responsible for managing an “active-alarm-list”. Alarms with severity “warning” are excluded from this list. When an alarm is detected it is added to the list; when the alarm reason disappears then the alarm is cleared - removed from the “active-alarm-list”. Furthermore, when the element that was the “fault-source” of an alarm is deleted then all related alarms are removed from the “active-alarm-list”.

The NETCONF Client can read “active-alarm-list” by **get** rpc operation.

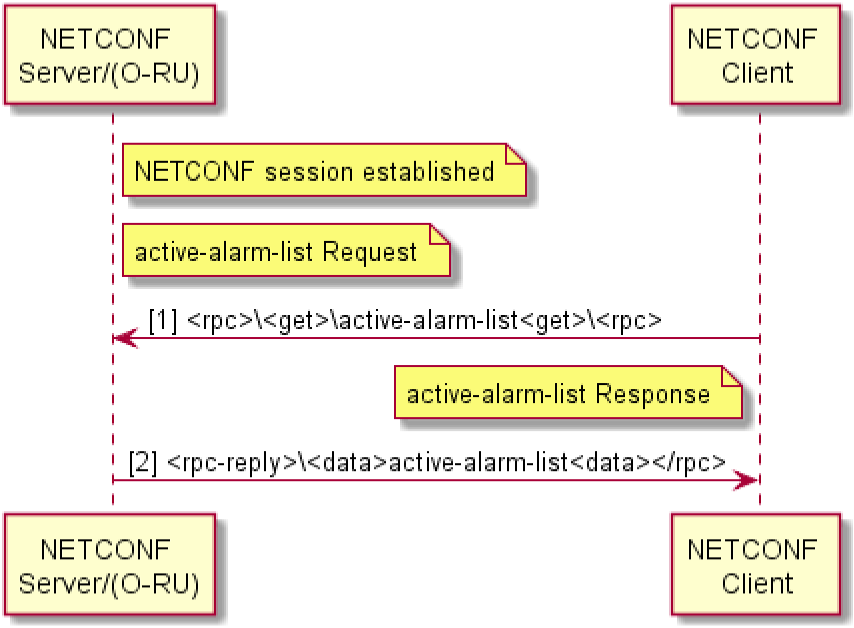


Figure 31: Read Active Alarms

8.1 Alarm Notification

The O-RU is responsible to send <alarm-notif> to a configured subscriber when the NETCONF Client has established a subscription to alarm notification and:

- a new alarm is detected (this can be the same alarm as an already existing one, but reported against a different “fault-source” than the existing alarm)
- an alarm is removed from the list

Removal of alarms from the list due to deletion of “fault-source” element is considered as clearing and cause sending of <alarm-notif> to the configured subscriber. This applies to alarms which were explicitly related to the deleted “fault-source” element. The rationale for such is to avoid misalignment between NETCONF Clients when one NETCONF Client deletes an element.

The O-RU reports in <alarm-notif> only for new active or cancelled alarms, not all active alarms.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

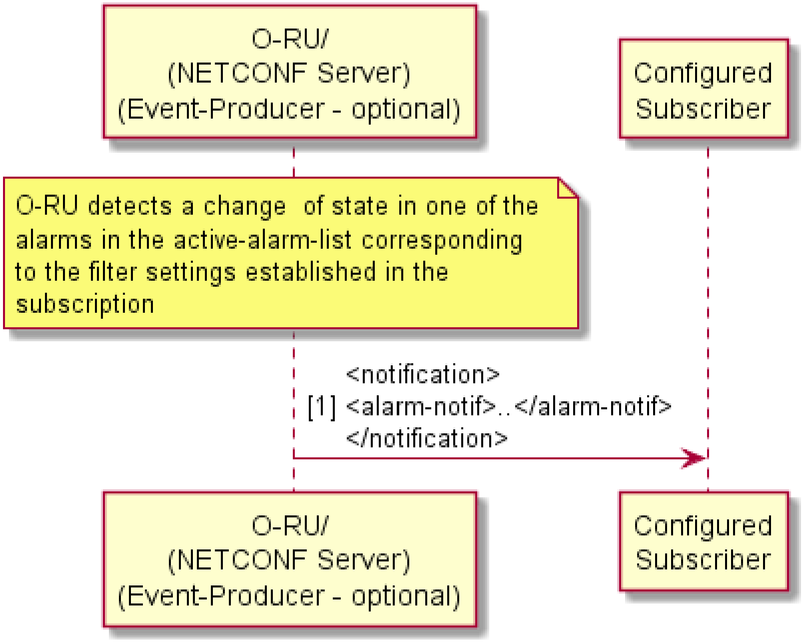


Figure 32: Alarm Notification

8.2 Manage Alarms Request to NETCONF Clients

The NETCONF Client can “subscribe” to Fault Management Element by sending **create-subscription**, RFC5277 [21], to NETCONF Server.

RFC5277 allows <create-subscription> below:

```
<netconf:rpc netconf:message-id="101"
    xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
  <create-subscription
    xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
   <filter netconf:type="subtree">
    <event xmlns="http://example.com/event/1.0">
     <eventClass>fault</eventClass>
     <severity>critical</severity>
    </event>
    <event xmlns="http://example.com/event/1.0">
     <eventClass>fault</eventClass>
     <severity>major</severity>
    </event>
    <event xmlns="http://example.com/event/1.0">
     <eventClass>fault</eventClass>
     <severity>minor</severity>
```

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

```
            </event>
          </filter>
        </create-subscription>
      </netconf:rpc>
```

Note: the NETCONF Client can disable/enable alarm sending only for all the alarms with same severity, not for single alarms.

The appropriate example for O-RAN YANG modules for **create-subscription** is as follows:

Case 1) NETCONF client subscribes **alarm-notif** filtering **fault-severity**: CRITICAL, MAJOR and MINOR and **measurement-result-stats** filtering **transceiver-stats** and **rx-window-stats** which **measurement-object** is RX_ON_TIME only:

```
<rpc xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <create-subscription
      xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <filter netconf:type="subtree">
      <alarm-notif xmlns="urn:o-ran:fm:1.0">
          <fault-severity>CRITICAL</fault-severity>
      </alarm-notif>
      <alarm-notif xmlns="urn:o-ran:fm:1.0">
          <fault-severity>MAJOR</fault-severity>
      </alarm-notif>
      <alarm-notif xmlns="urn:o-ran:fm:1.0">
          <fault-severity>MINOR</fault-severity>
      </alarm-notif>
      <measurement-result-stats xmlns="urn:o-ran:performance-management:1.0">
          <transceiver-stats/>
      </measurement-result-stats>
      <measurement-result-stats xmlns="urn:o-ran:performance-management:1.0">
          <rx-window-stats>
              <measurement-object>RX_ON_TIME</measurement-object>
           </rx-window-stats>
      </measurement-result-stats>
    </filter>
  </create-subscription>
</rpc>
```

Case 2) NETCONF client subscribes default event stream NETCONF to receive all notifications defined in O-RAN YANG modules:

```
<rpc xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <create-subscription xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <stream>NETCONF</stream>
  </create-subscription>
</rpc>
```

A high-level view of a NETCONF Client subscribing is shown directly below. After the NETCONF Client requests a subscription, the server sends an alarm-notif notification to the client when there is any change in the active alarms matching the filter specified in the subscription request.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

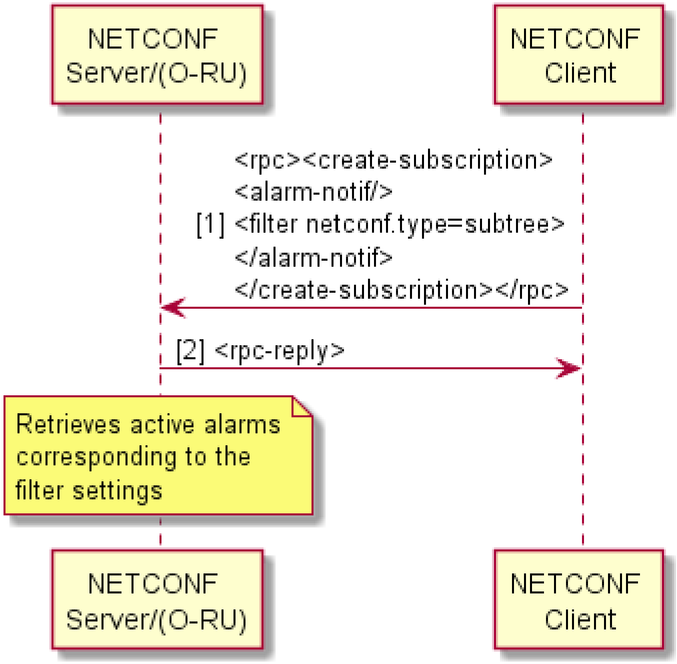


Figure 33: Manage Alarms Subscription Request

To terminate the subscription, the NETCONF client shall send a <close-session> operation from the subscription's session.

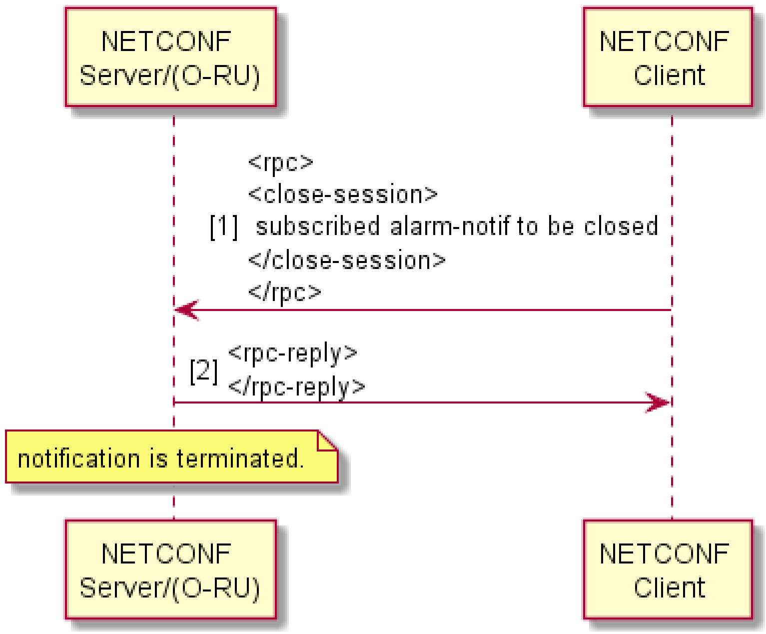


Figure 34: Terminating an Alarm Subscription

8.3 Fault Sources

Alarm notifications reported by NETCONF Server contain element "fault-source" which indicates the origin of an alarm. In general values of "fault-source" are based on names defined as YANG leafs:

- Source (Examples: fan, module, PA, port)

indicates that origin of the alarm within the O-RU. Value of "fault-source" is based on element name.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Note: In case the NETCONF Server reports an unknown “fault-source”, the NETCONF Client can discard the <alarm-notif>.

- Source (other than when an element is within the O-RU)

Value of fault-source may be empty or may identify the most likely external candidate; for example, antenna line.

Alarms with different “fault-id”, “fault-source” or "fault-severity" are independent:

- Multiple alarms with same “fault-id” may be reported with different “fault-source”.
- Multiple alarms with same “fault-source” may be reported with different “fault-id”.
- When an alarm with a "fault-id" and a "fault-source" is reported with a "fault-severity" and its severity of alarm condition is upgraded or degraded, NETCONF server reports a new alarm with the same "fault-id" and the same "fault-source" with the upgraded or degraded "fault-severity" with "is-cleared"::FALSE and clears the previous alarm with the report of the "fault-id", "fault-source" and "fault-severity" with "is-cleared":: TRUE.

The range of "fault-id" is separated to common and vendor specific. The common fault-ids are defined in Annex A and more number will be used in future. The vendor specific range for the fault-id shall be [1000 .. 65535].

Alarm notifications reported by the NETCONF Server contain names of the “affected-objects” which indicate elements affected by the fault. In case the origin of the alarm is within the O-RU, other elements than “fault-source” which will not work correctly due to the alarm are reported via “affected-objects”. In case the origin of the fault is outside of the O-RU, the O-RU elements which will not work correctly due to the fault are reported via “affected-objects”.

8.4 Manage Alarms Request to Event-Collector

This optional capability requires the O-RU to support configured subscriptions, as described in Chapter 15. The structure of the process follows the process described in sub-section 8.2. However, instead of sending a NETCONF <create-subscription> to the NETCONF server in the O-RU to subscribe to the **alarm-notif** notifications, the NETCONF client installs the subscription via configuration of the O-RU’s datastore. Based on configured subscriptions, the O-RU sends asynchronous YANG notifications over HTTPS to the configured Event-Collector.

In order to terminate the subscription, the NETCONF client shall delete the corresponding configuration in the O-RU. Immediately after the subscription is successfully deleted, the O-RU will send to a subscription state change notification indicating that the subscription has ended to the Event-Collector.

Chapter 9 File Management

This chapter specifies File Management for the O-RU. Following operations are supported as a File Management.

- upload (see subsection 9.2)

  File upload from O-RU to file server triggered by O-RU Controller.

- retrieve file list (see subsection 9.3)

  O-RU Controller retrieves the file list in O-RU.

- download (see subsection 9.4)

  File download from file server to O-RU triggered by O-RU Controller

  Note: file-download has different purpose with software-download specified in subsection 5.3. For example, file-download can be used for Beamforming configuration in subsection 12.4.

File transfers are done with sFTP. Following types of authentications are supported for **file management**:

a) Password for RU authentication and list of public keys (DSA/RSA) for sFTP server authentication

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

b) Certificate for both RU and sFTP server authentication

Following other sections are related with File Management.

- Subsection 7.2.2: File Management process (can be used for on demand file upload purpose, since subsection 7.2.2 covers periodic file upload)

- Subsection 11.2: Log Management

- Subsection 12.4: Beamforming Configuration

9.1 File System Structure

The file System structure of the O-RU is represented as a logical structure that is used by the file management procedures defined in the rest of this chapter. If the O-RU's physical file structure differs from the logical file structure defined below, the O-RU is responsible for performing the mapping between the two structures.

The O-RU shall support the standardized logical folders. In this version of the M-Plane specification, the following standardized folders are defined:

O-RAN/log/

O-RAN/PM/

O-RAN/transceiver/

And for those O-RU's supporting beamforming

O-RAN/beamforming/

The O-RU may additionally support vendor defined folders which are out of scope of this specification.

9.2 File Management Operation: upload

This subsection describes file upload method from O-RU to the fileserver. sFTP is used for File management, and one file can be uploaded by one upload operation. The O-RU Controller triggers file upload operation to O-RU.

Simultaneous multiple file upload operations can be supported under the same sFTP connection between O-RU to the fileserver. If the O-RU has the number of limitation to upload simultaneously as a capability, it is allowed that O-RU reports failure notification for the upload request which is larger than the capability. The behaviour of O-RU Controller is out of scope when O-RU Controller receives failure notification from the O-RU.

Following rpc is used for upload operation.

-rpc: **file-upload**

- input
  - local-logical-file-path: the logical path of file to be uploaded (no wildcard is allowed)
  - remote-file-path: URI of file on the fileserver
- output
  - status: whether O-RU accepted or rejected the upload request
  - reject-reason: the human readable reason why O-RU rejects the request (only applicable if status is rejected)

In the rpc-reply, status whether the O-RU receives the upload request or rejects due to some reason (e.g., the number of limitation to upload simultaneously) is replied. If rejected, the human readable reject reason is also replied.

In notification, the result of the upload process (successfully uploaded or failed upload) is replied in addition to local-logical-file-path and remote-file-path. If failure, the human readable reason is also replied.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Figure 35 shows the file upload sequence diagram.

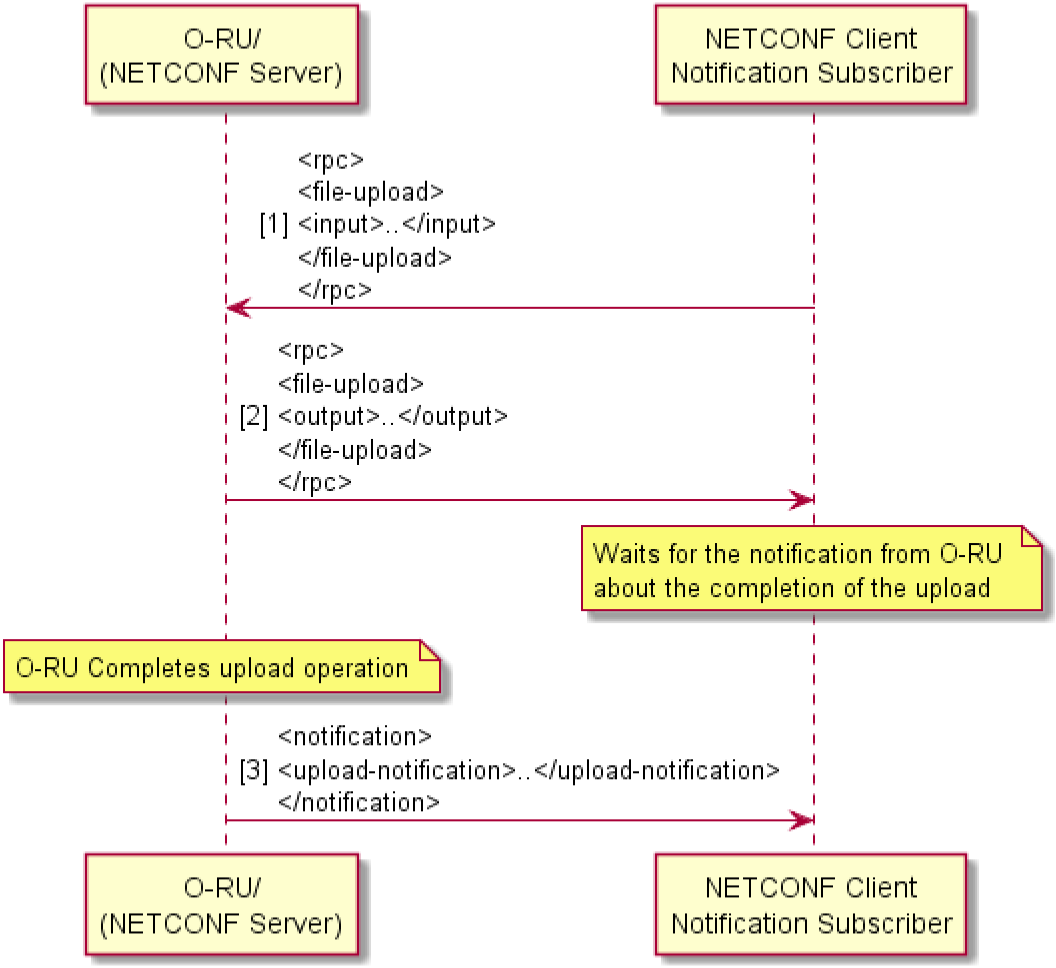


Figure 35: File Upload Sequence

9.3 File Management Operation: retrieve file list

This subsection describes file retrieve method which the O-RU Controller retrieves the file list from the O-RU. One or multiple files' information can be retrieved by one retrieve file list operation (use of wildcard is allowed). The O-RU Controller triggers the retrieve file list operation from the O-RU.

The following rpc is used for retrieve file list operation.

-rpc: **retrieve-file-list**

- input
  - logical path: the logical path of files to be retrieved (* is allowed as wild-card)
  - file-name-filter: the files which has the “file name filter” in the file name (* is allowed as wild-card)
- output
  - status: whether O-RU accepted or rejected the retrieve file list request
  - reject-reason: the human readable reason why O-RU rejects the request (only applicable if status is rejected)
  - file list

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

In rpc-reply, status whether the O-RU accepts the retrieve-file-list request or rejects due to some reason is replied. If rejected, the human readable reject reason is also replied.

Figure 36 shows the retrieve file list sequence diagram.

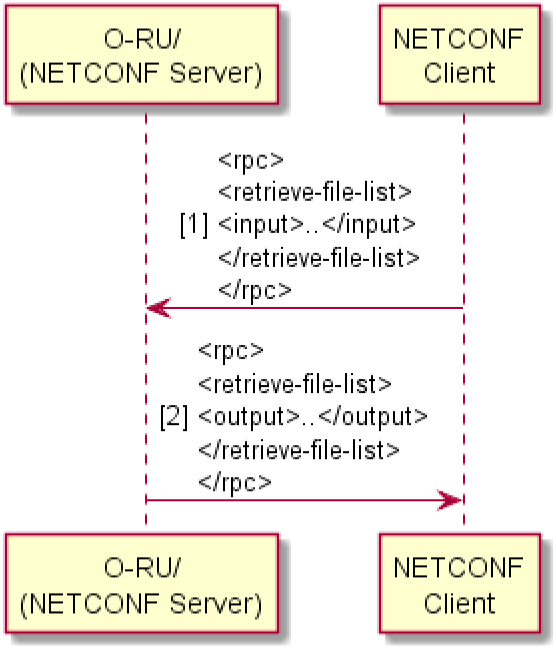


Figure 36: Retrieve File List Sequence

9.4 File Management Operation: download

This sub-chapter describes the file download method from O-RU Controller to O-RU. sFTP is used for File management, and one file can be downloaded by one download operation. O-RU Controller triggers the file download operation to O-RU.

Simultaneous multiple file download operations can be supported under the same sFTP connection between the O-RU and O-DU/SMO. If the O-RU has the number of limitation to download simultaneously as a capability, it is allowed that the O-RU reports a failure notification for the download request which is larger than the capability. The behaviour of the O-RU Controller is out of scope when O-DU/SMO receives failure notification from O-RU.

The following rpc is used for download operation.

-rpc: **file-download**

- input
    - local-logical-file-path: the logical path of file to be downloaded (no wildcard is allowed)
    - remote-file-path: URI of file on the fileserver
- output
    - status: whether O-RU accepted or rejected the download request
    - reject-reason: the human readable reason why O-RU rejects the request (only applicable if status is rejected)

In rpc-reply, status whether the O-RU receives the download request or rejects due to some reason (e.g., the number of limitation to download simultaneously) is replied. If rejected, the human readable reject reason is also replied.

In notification, the result of the download process (successfully downloaded or download is failure) is replied in addition to the local-logical-file-path and remote-file-path. If failure, the human readable reason is also replied. The figure below shows the file download sequence diagram.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

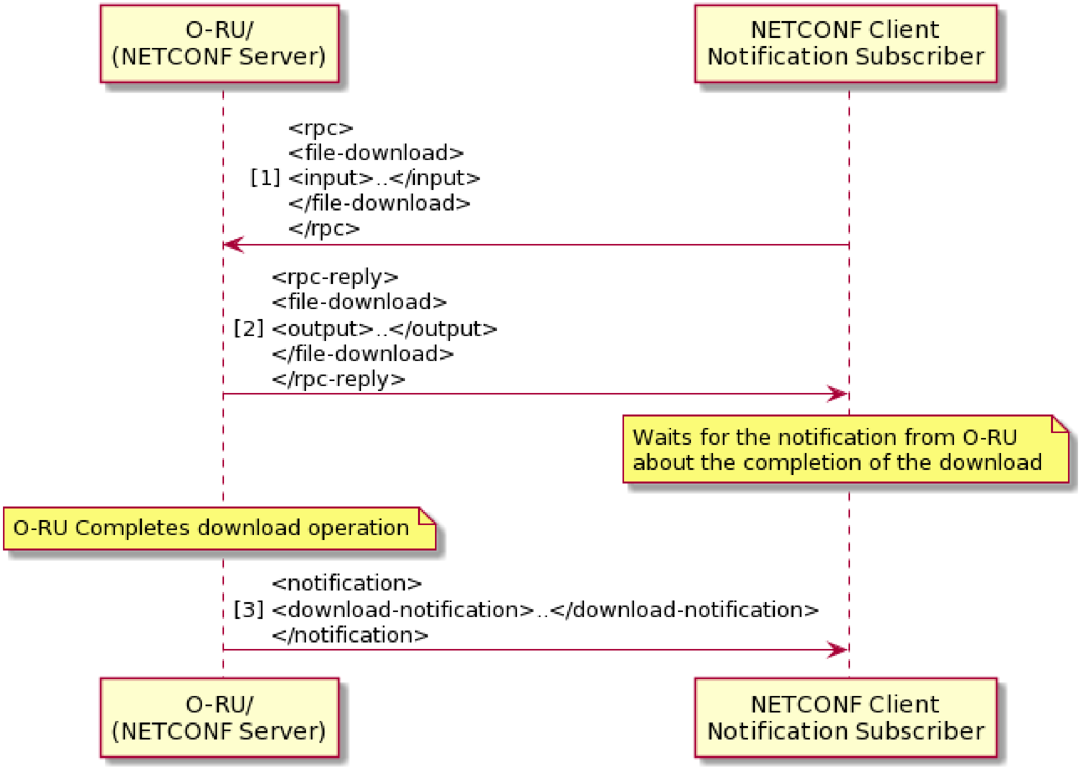


Figure 37: File Download Sequence

Chapter 10 Synchronization Aspects

This chapter provides the Management Plane's interactions with various aspects of the time synchronization of the O-RU. In general, the O-RU is responsible for managing its synchronization status, to select one or more synchronization input source(s) (based on vendor specific implementation) and assure that the resulting accuracy meets that required by the Radio Access Technology being implemented.

10.1 Sync Status Object

This **sync** container provides synchronization state of the module. If the O-RU Controller is interested in Sync status, it may configure a subscription to the **synchronization-state-change** notification in the O-RU. Event notifications will be sent whenever the state of the O-RU synchronization changes.

The State of O-RU synchronization is indicated by the following allowed values:

- **LOCKED:** O-RU is in the locked mode, as defined in ITU-T G.810.
- **HOLDOVER:** O-RU clock is in holdover mode.
- **FREERUN:** O-RU clock isn't locked to an input reference and is not in the holdover mode.

The **sync** container allows the O-RU to list via an array the synchronization sources which it is capable of supporting. The allowed values are:

- GNSS
- PTP
- SYNCE

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

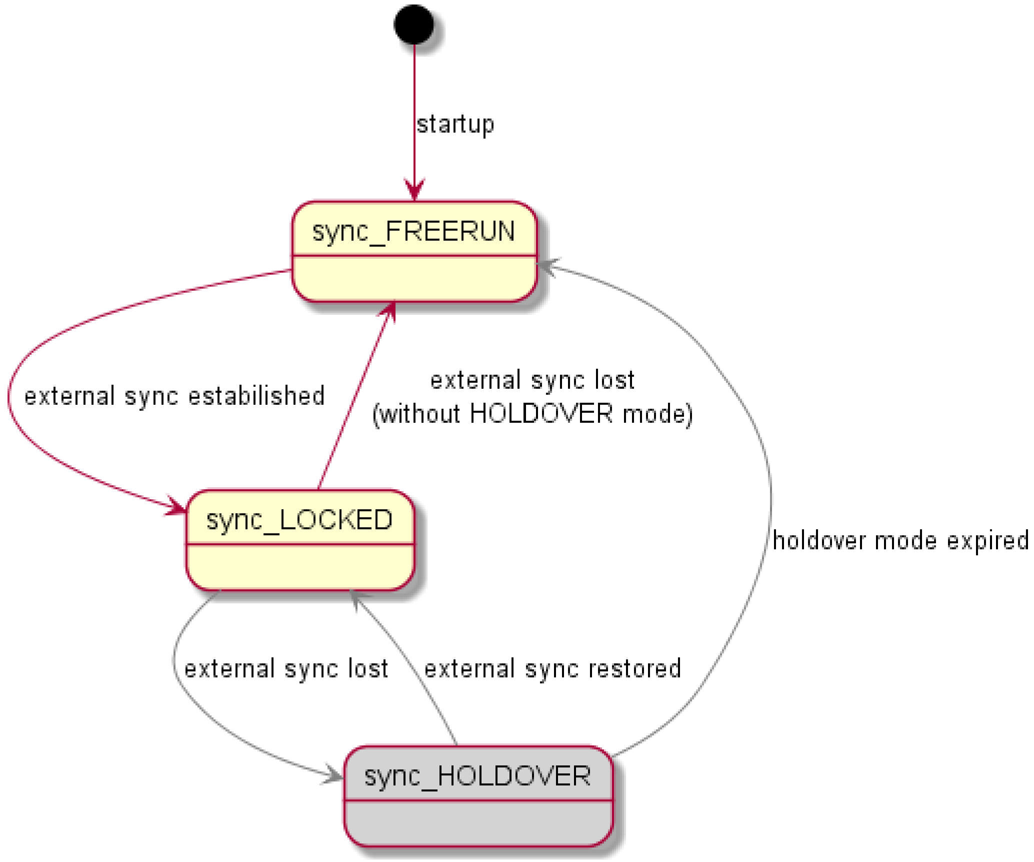


Figure 38: Allowed sync state transitions

Note: HOLDOVER mode is optional and depends on internal O-RU design.

10.2 Sync Capability Object

The module's synchronization capability is provided via this object. This indicates the accuracy of the derived Telecom Subordinate Clock (T-TSC) to which the module's design is capable of. For details on the actual capability levels, see section 9.3 of the O-RAN WG4 CUS plane specification [2]. There are two enumerations possible:

- CLASS_B
- ENHANCED

10.3 PTP Configuration

This container defines the configuration of Precision Time Protocol.

domain-number

This parameter indicates the Domain Number for PTP announce messages. Allowed values: 0 ~ 255.

Default: 24.

Note: ITU-T G.8275.1 [22] uses domain numbers in the range 24...43, but the entire range is allowed to ensure flexibility of the M-Plane specification. For ITU-T G.8275.2 domain numbers from range 44…63 shall be used.

accepted-clock-classes

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Contains the list of PTP acceptable Clock Classes, sorted in the descending order.

Note: The sender must generate the list of acceptable clock classes. The list must be sorted in descending order. Each accepted Clock Class value must appear only once in the list. Depending on implementation, the receiver may interpret the list in either of two ways:

a) use only the first (i.e. the maximum) item in the list, interpreting it as a threshold value for acceptable clock classes, while ignoring all other items in the list;

b) use the whole list, interpreting it as an explicit list of acceptable clock classes.

Default: 7, 6

clock-class

The PTP Clock Class accepted by the O-RU. Allowed values: 0 ~ 255.

Note: Not all values are compliant to [22], but the entire range is allowed in M-plane specification to ensure flexibility. The values can be validated/filtered on the receiver side, if necessary.

ptp-profile

Defines which PTP profile will be used.

Allowed values:

- G_8275_1 (multicast over Ethernet will be used, see: ITU-T G.8275.1)
- G_8275_2 (unicast over IP will be used, see: ITU-T G.8275.2)

Default: G_8275_1.

delay-asymmetry

Defines the static phase error in the recovered PTP timing signal to be compensated at the O-DU. The error is defined in units of nanoseconds in the range ±10 000 ns. According to ITU-T G.810 [23] and IEEE1588v2-2008 [24] the sign of the parameter is interpreted as follows:

- If the phase error to be compensated is negative, then the recovered timing signal shall be advanced by the time interval equal to the configured value to compensate the error.
- If the phase error to be compensated is positive, then the recovered timing signal shall be delayed by the time interval equal to the configured value to compensate the error.

Default: 0

Note: Modification of this parameter may have impact on RF transmission, but shall occur without unit restart.

Note: This parameter is optional for support. If the O-RU does not support this value, the O-RU uses the default value. If the O-RU does not support manual compensation, it ignores the parameter setting.

Note: Granularity of the applied value depends on the architecture and implementation of the system clock, and therefore, may vary across vendors.

10.3.1 G.8275.1 specific parameters

multicast-mac -address

The parameter defines the destination MAC address, used by the O-RU in the egress PTP messages.

Allowed values:

- FORWARDABLE (means that PTP shall use 01-1B-19-00-00-00 destination MAC address)
- NONFORWARDABLE (means that PTP shall use 01-80-C2-00-00-0E destination MAC address)

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Default value: FORWARDABLE.

10.3.2 G.8275.2 specific parameters

This section contains G.8275.2 specific parameters

local-ip-port

The parameter defines local ip address which will be used as a port for receiving ptp signal

master-ip-configuration

The parameter defines list of ip configuration of devices acting as ptp signal source.

local-priority

The parameter defines local priority or underlying master IP address.

ip-address

the parameter defines master IP address.

log-inter-sync-period

The parameter defines number of sync message during 1 second

Allowed values: 0 ~ -7 (this represents the value from 1 message per second to 128 messages per second)

log-inter-announce-period

The parameter defines number of announce message during 1 second

Allowed values: 0 ~ -3 (this represents the value from 1 message per second to 8 messages per second)

10.4 PTP Status

The PTP Status container is used to collect operational status information of the PTP ordinary clock, controlled by the O-RU. The object may be used to display operational information, which facilitates troubleshooting, to the operator. The information in the object shall not be used by the O-DU to autonomously alter its operation. If the O-RU Controller is interested in PTP status, it may configure a subscription to the **ptp-state-change** notification in the O-RU. Notifications will only indicate changes to the lock-state. Before requesting or subscribing to PTP status information, the O-RU Controller shall ensure that PTP is supported by the O-RU by requesting the **supported-timing-reference-types**, as defined in this chapter (Section 10.1). The following list includes the related parameters of this container.

reporting-period

This parameter defines minimum period in seconds between reports, sent by the O-RU, for parameters in this container.

default: 10

lock-state

This parameter indicates whether the integrated ordinary clock is synchronizing to the reference, recovered from PTP flow. The exact definition when to indicate locked or unlocked is up to specific implementation.

- **LOCKED:** The integrated ordinary clock is synchronizing to the reference, recovered from PTP flow.
- **UNLOCKED:** The integrated ordinary clock is not synchronizing to the reference, recovered from PTP flow.

clock-class

This parameter contains the clock class of the clock, controlled by the O-RU.

sources

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

This parameter contains characteristics of PTP sources of the clock, controlled by the O-RU.

state

This parameter indicates status of the PTP source:

- **PARENT:** Indicates that the PTP signal from this source is currently used as a synchronization reference.
- **OK:** Indicates that the PTP signal from this source can be potentially used as a synchronization reference, i.e. Announce messages, received from this source, contain acceptable content (domain number, clockclass, flags, etc).
- **NOK:** Indicates that the PTP signal from this source cannot be used as a synchronization reference, i.e. Announce messages, received from this source, contain unacceptable content (domain number, clockclass, flags, etc).
- **DISABLED:** Indicates that PTP connection is not available from this PTP source.

See the related o-ran-sync YANG Model for the full details.

10.5 SyncE Configuration

This container defines the configuration of SyncE

acceptance-list-of-ssm

The parameter contains the list of SyncE acceptable Synchronization Status Messages (SSM).

Allowed values:

- PRC (Primary Reference Clock)
- PRS (Primary Reference Source-Stratum 1)
- SSU_A (Synchronisation Supply Unit A)
- SSU_B (Synchronisation Supply Unit B)
- ST2 (Stratum 2)
- ST3 (Stratum 3)
- ST3E (Stratum 3E)
- EEC1 (Ethernet Equipment Clock 1)
- EEC2 (Ethernet Equipment Clock 2)
- DNU (Do Not Use)
- NONE

ssm-timeout

The parameter contains the value of maximum duration in seconds for which the actual SSM value may be different than configured values.

10.6 SyncE Status

The SyncE Status container is used to collect operational status information of SyncE reference on a node, controlled by O-RU. If the O-RU Controller is interested in SyncE status, it may configure a subscription to the **synce-state-change** notification in the O-RU. Notifications will only indicate changes to the lock-state. Before requesting or subscribing to SyncE status information, the O-RU Controller shall ensure that SyncE is supported at the O-RU by requesting the supported timing reference types, as defined earlier in this section 10.1. The following list summarizes the related parameters of this container.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

reporting-period

This parameter defines minimum period in seconds between reports, sent by the O-RU, for parameters in this container.

default: 10

lock-state

This parameter indicates, whether the integrated ordinary clock is synchronizing to the reference, recovered from the SyncE signal. The exact definition when to indicate locked or unlocked is up to specific implementation.

- **LOCKED:** The integrated ordinary clock is synchronizing to the reference, recovered from the SyncE signal.
- **UNLOCKED:** The integrated ordinary clock is not synchronizing to the reference, recovered from the SyncE signal.

sources

This parameter contains characteristics of SyncE sources of the clock, controlled by the NETCONF Server

state

This parameter indicates status of the SyncE source:

- **PARENT:** Indicates that the SyncE signal from this source is currently used as a synchronization reference.
- **OK:** Indicates that the SyncE signal from this source can be potentially used as a synchronization reference, i.e. SSM messages, received from this source, contain acceptable clock quality level.
- **NOK:** Indicates that the SyncE signal from this source cannot be used as a synchronization reference, i.e. SSM messages, received from this source, contain unacceptable clock quality level.
- **DISABLED:** Indicates that SSMs are not received from this SyncE source.

quality-level

This parameter contains value of the SSM clock quality level, received in SSM messages from the SyncE source.

See the related o-ran-sync YANG Model for the full details.

10.7 GNSS Configuration

This container defines the configuration of Global Navigation Satellite System (GNSS).

enable

This parameter defines if GNSS receiver shall be enabled or not. Allowed values: true/false;

Default values: false.

satellite-constellation-list

This parameter defines list of constellations to be used to acquire synchronization.

Allowed values:

- GPS
- GLONASS
- GALILEO
- BEIDOU

polarity

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

This parameter defines pulse polarity

Allowed values:

- POSITIVE
- NEGATIVE

Default value: POSITIVE.

cable-delay

This parameter is used to compensate cable delay. Allowed-values: 0 ~ 1000

Default value: 5

Note: This value is given in ns (nanoseconds) it is recommended to compensate 5ns per each meter of the cable.

anti-jam-enable {if feature GNSS-ANTI-JAM}

This parameter is used to enable or disable anti-jamming. Allowed values: true/false

Default value: false.

10.8 GNSS Status

An O-RU supporting GNSS capability uses the **gnss-state** container to report the state of its GNSS receiver. If the O-RU Controller is interested in GNSS status, it may configure a subscription to the **gnss-state-change** notification in the O-RU before requesting or subscribing the GNSS status information. Notifications will only provide changes to the gnss-status. The O-RU Controller shall ensure that GNSS is supported by the O-RU by requesting supported timing reference types, as defined in this chapter (Section 10.1). The following list summarizes the related parameters of this container.

gnss-status

This parameter indicates the status of the GNSS receiver:

- **SYNCHRONIZED:** Indicates that the GNSS receiver is synchronized.
- **ACQUIRING-SYNC**: Indicates the GNSS receiver is functioning correctly, but has not acquired synchronization
- **ANTENNA-DISCONNECTED:** Indicates the GNSS receiver is reporting that its antenna is disconnected.
- **INITIALIZING:** Indicates that the GNSS receiver is initializing.
- **ANTENNA-SHORT-CIRCUIT**: Indicates that the GNSS receiver is reporting that its antenna is short circuited.

Additionally, when the GNSS receiver is synchronized, the O-RU can report the following additional information:

satellites-tracked

The number of satellites being tracked by the O-RU receiver

altitude, **latitude** and **longitude**

The geospatial location reported by the GNSS receiver

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Chapter 11 Operations Use Cases

11.1 Supervision Failure

Section 3.6 describes the procedure by which an O-RU uses the expiry of a supervision watchdog timer to trigger “supervision failure” condition.

Note, an O-RU operating in supervision failure may still have NETCONF sessions established with one or more NETCONF clients with non-“sudo” privileges.

An O-RU that detects a supervision failure shall immediately cease RF transmission. The O-RU shall perform an autonomous recovery reset procedure. This procedure will trigger the re-initialization of the transport layer and re-commencement of the start-up installation procedures defined in section 3.

11.2 Log management

There are two type of log managements, troubleshooting log and trace log. They are independent each other.

Troubleshooting log file contains the logs continuously collected before <start-troubleshooting-logs> rpc. Any logs collected after <start-troubleshooting-logs> rpc are not contained.

Trace log file contains the logs continuously collected after <start-trace-logs> rpc. Any logs collected before <start-trace-logs> rpc are not contained.

11.2.1 Troubleshooting

By requesting technical logs an O-RU controller is able to get collected log data files that can be used for troubleshooting purposes.

The O-RU provides all possible troubleshooting log files. The contents and log formats are dependent on O-RU implementation.

The number and size of files provided by O-RU is not restricted but the O-RU may keep the number and size of files reasonably small to allow completion of the whole “Troubleshooting data upload” scenario (all files) within 15 minutes (with target to complete within 3 minutes). It is also recommended to provide more useful files first.

Note, the O-RU controller is free to continue the scenario till completion past the allowed time or skip requesting further files.

The files should be compressed with compression method indicated by file name extension:

- .gz (DEFLATE),
- .lz4 (LZ4),
- .xz (LZMA2 - xz utils),
- .zip (DEFLATE - zlib library).

O-RU shall start collecting the troubleshooting logs.

The rpc <start-troubleshooting-logs> triggers the O-RU to start creating files containing troubleshooting logs. Completed generation of files is indicated by NETCONF server to NETCONF client in form of a notification.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

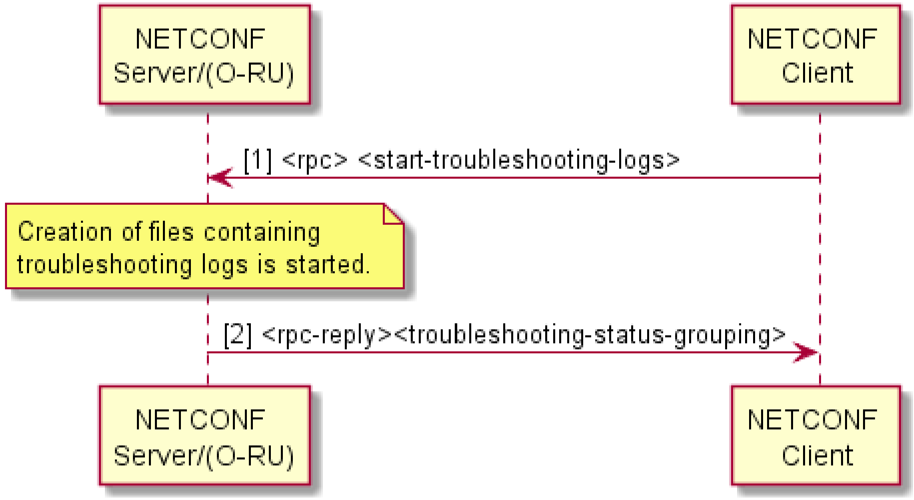


Figure 39: Start Creating Troubleshooting Log

The rpc <stop-troubleshooting-logs> informs O-RU that NETCONF client cancels creating troubleshooting logs files and is no longer interested in receiving <troubleshooting-log-generated> notification.

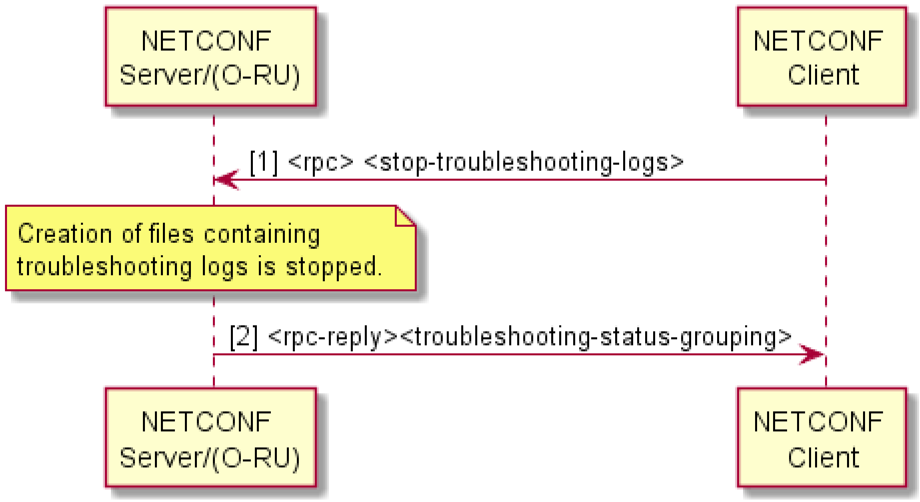


Figure 40: Stop Troubleshooting Log

The notification <troubleshooting-logs-generated> is signalled to the configured subscriber after the O-RU has finished creating all troubleshooting logs, indicating to the subscriber that the logs are ready to be uploaded. The O-RU shall include URLs for all troubleshooting log files in the notification, corresponding to the troubleshooting logs collected before rpc <start-troubleshooting-logs>.

After notification, the O-RU shall stop sending the further notifications <troubleshooting-logs-generated> for troubleshooting logs collected after rpc <start-troubleshooting-logs> without the rpc <stop-troubleshooting-logs>.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

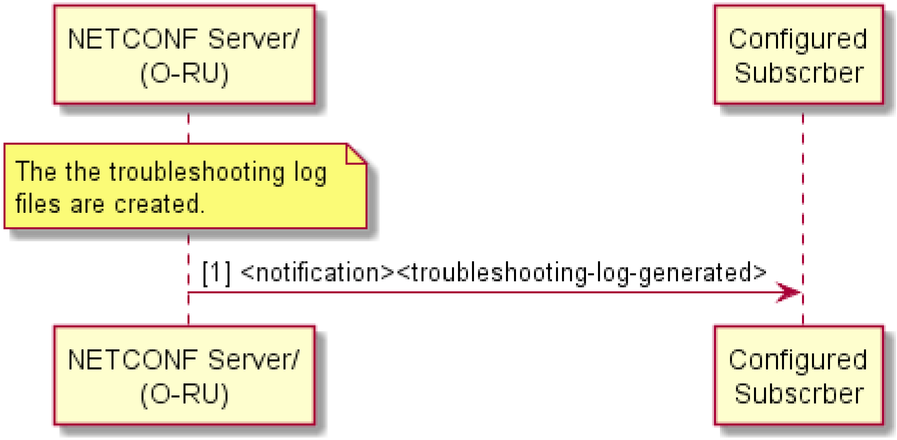


Figure 41: Generate Troubleshooting Log

The file transfer mechanism for the created troubleshooting log files shall be handled by the file management in chapter 9.

The overall troubleshooting log behaviour is illustrated in the following figure. It contains 2 cases, successful notification case and no notification as abnormal case. In the successful notification case, the notification provides URLs of files containing all troubleshooting logs collected before rpc <start-troubleshooting-logs>. After notification, NETCONF client doesn't need to signal rpc <stop-troubleshooting-logs> since RU stops creating troubleshooting logs. In the no notification case as one of abnormal scenarios, NETCONF client can make O-RU cancel creating the log files because O-RU doesn't send a notification for created log files for a long period.

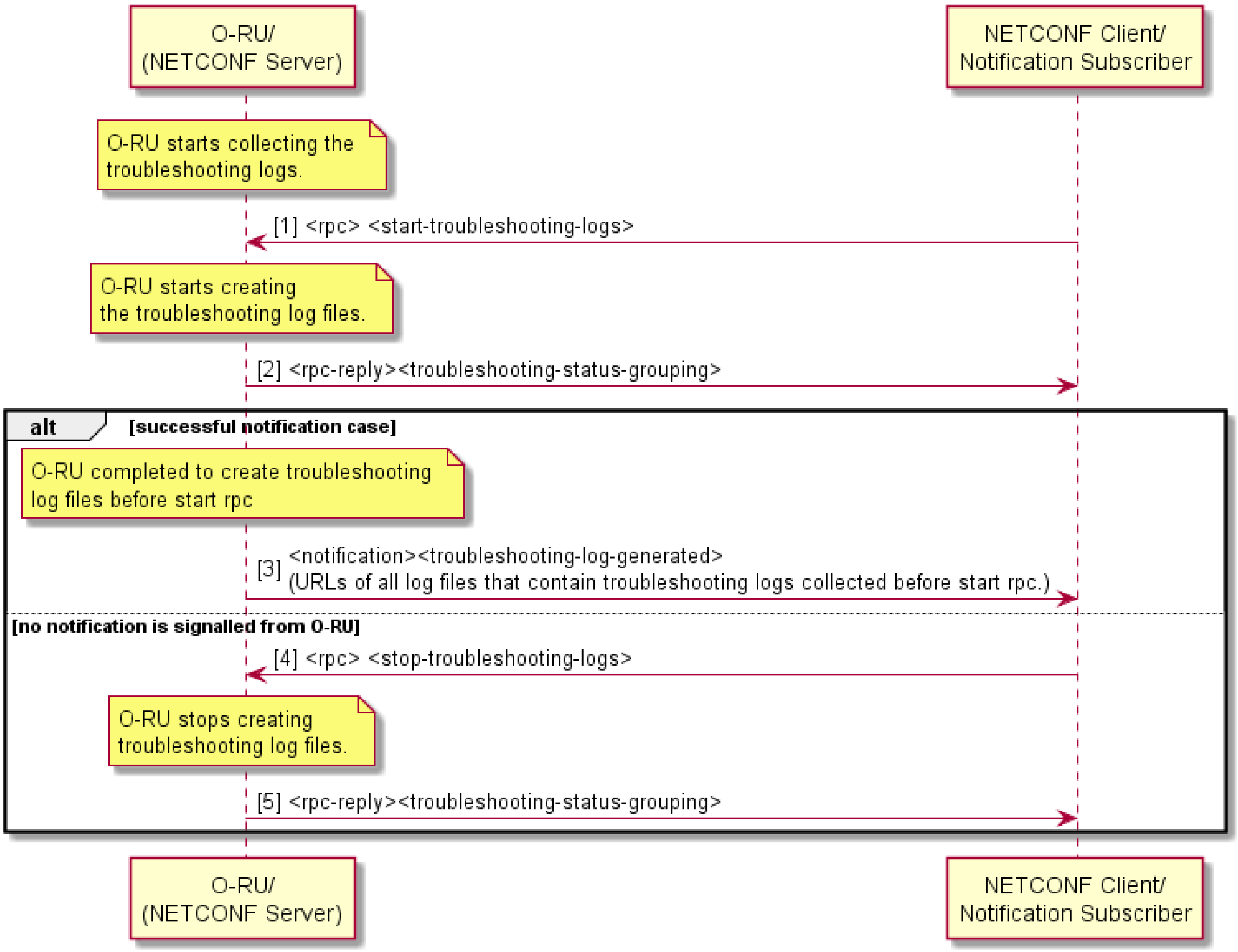


Figure 42: Overall Troubleshooting Log behaviour

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

11.2.2 Trace

By requesting trace logs an O-RU controller is able to get collected log data files that can be used for trace purposes.

The O-RU provides all possible trace log files. The contents and log formats are dependent on O-RU implementation.

The files should be compressed with compression method indicated by file name extension.

O-RU shall start collecting the trace logs at the moment of receiving <start-trace-logs> rpc. Notification <trace-log-generated> shall be periodically send to O-RU controller once collected logs stored in file(s) is(are) ready. URL of newly created log file(s) shall be included in notification. The number and size of files provided by O-RU in a single <trace-log-generated> notification is not restricted but the O-RU may keep the number and size of files reasonably small to allow completion of the whole "Trace data upload" scenario (all files from notification) within 15 minutes (with target to complete within 3 minutes).

Note: Timing of creating trace log files is up to O-RU implementation.

After <stop-trace-logs> rpc received from controller, O-RU is mandated to stop collecting trace logs and to send last <trace-log-generated> notification with the is-notification-last:: TRUE and URL(s) of file(s) which contain log data collected between previous <trace-log-generated> notification and <stop-trace-logs> rpc.

The file transfer mechanism for the created trace log files shall be handled by the file management described in Chapter 9 File Management.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

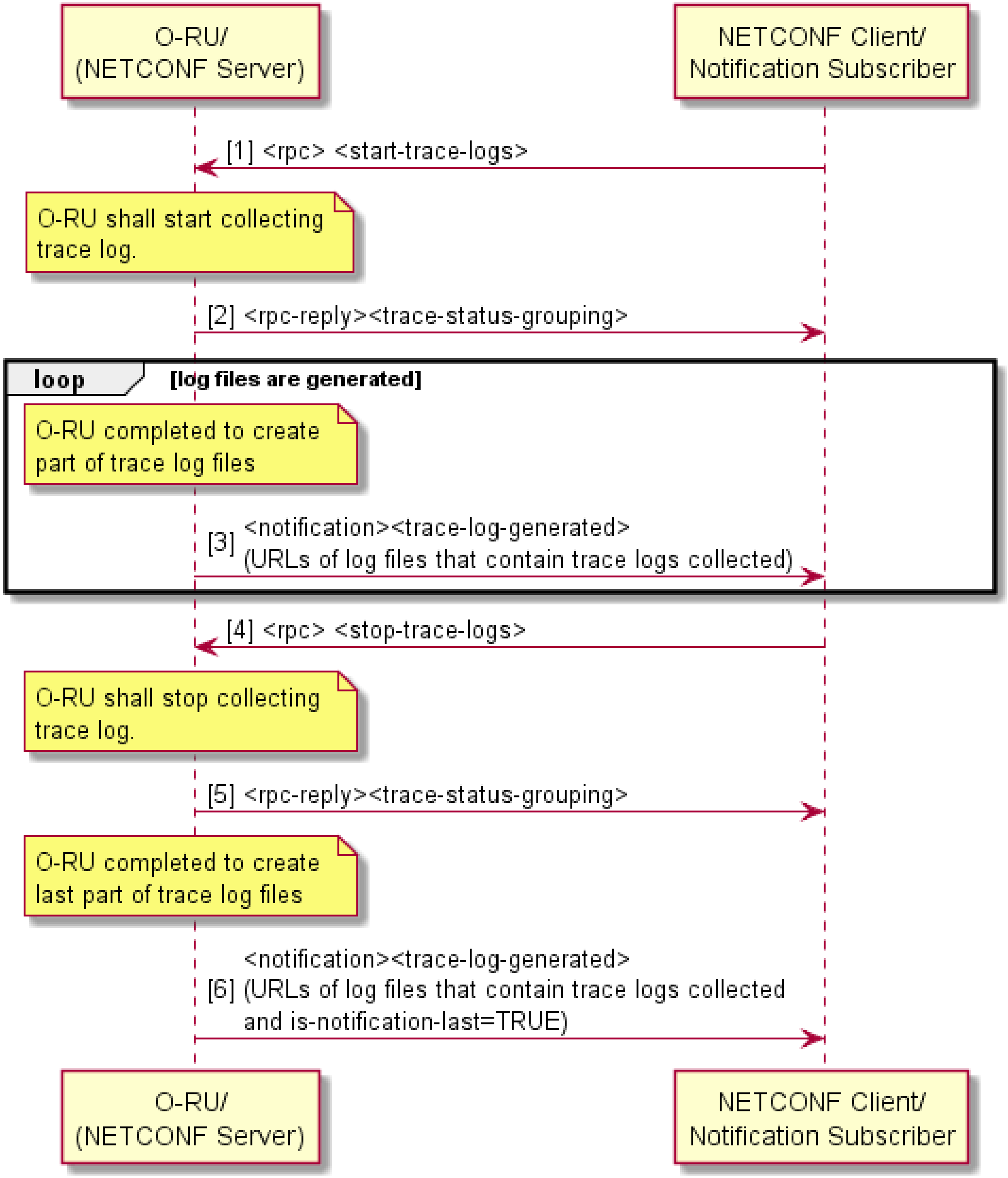


Figure 43: Overall Trace Log behaviour

11.3 Operational aspects of Antenna Line Devices

An O-RU can connect to one or more external equipment such as a RET, MultiRET, MHA, RAE and etc.

For the communication with the external equipment, AISG 2.0 protocol [26] as Layer 7 application and HDLC protocol as Layer 2 data link are used.

- HDLC protocol is standardized by ISO/IEC 13239 (https://www.iso.org/standard/37010.html). Detailed information can also be found in TS 25.462 [27].

- AISG 2.0 protocol is standardized by "Control interface for antenna line devices Standard No. AISG v2.0" [26] which is an adaptation of Iuant interface application layer defined in TS 25.466 [28] and formerly in TS 25.463 [29].

An O-RU may provide one or more ALD ports supporting connection with Antenna Line Devices. Each ALD port shall be able to support more than one ALD (i.e., a chained ALD configuration).

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

This section describes the communication mechanisms based on AISG 2.0 protocol [26]. For communication with external equipment, AISG 2.0 uses Application Part protocols (RETAP, TMAAP etc.) at Layer 7 and HDLC as a Layer 2 datalink protocol.

Note, further definitions of O-RU support for Antenna Line Devices will be included in a future version of this specification, and will target support of the AISG 3.0 protocol.

11.3.1 HDLC Interworking

HDLC protocol is standardized by ISO/IEC 13239. Detailed information can also be found in TS 25.462 [27]. The AISG 2.0 protocol is standardized by "Control interface for antenna line devices Standard No. AISG v2.0" which is an adaptation of Iuant interface application layer defined in TS 25.466 [28] and formerly in TS 25.463 [29].

Note: the assumed HDLC communication speed is 9600 bits per second.

In order to handle collision detection in the HDLC branch, an O-RU supporting the ALD functionality shall support the following running counters reported using the corresponding YANG model:

- Frames with wrong FCS
- Frames without stop flag
- Number of received octets

For running counters served by the O-RU, both the O-RU and NETCONF Client shall handle wrap-over mechanism in a way, that wrap over zero is not considered as erroneous situation.

A NETCONF client can recover these counters. From the changes observed in above counters, a NETCONF Client can deduce the presence of a collision on the HDLC bus. Additional diagnostic information may be derived from how these counters are incrementing.

Additionally, the O-RU implements "RPC Status" to indicate status of last "ald-communication" RPC to requestor.

- Status - flow control indicator of last requested operation (Status of RPC).

Prior to any communication with ALD(s), the O-RU shall provide ALDs with DC power. The way of how DC power is managed is out of scope of this interface specification.

In order to support collision detection and flow control, the following reference architecture with functional split is defined:

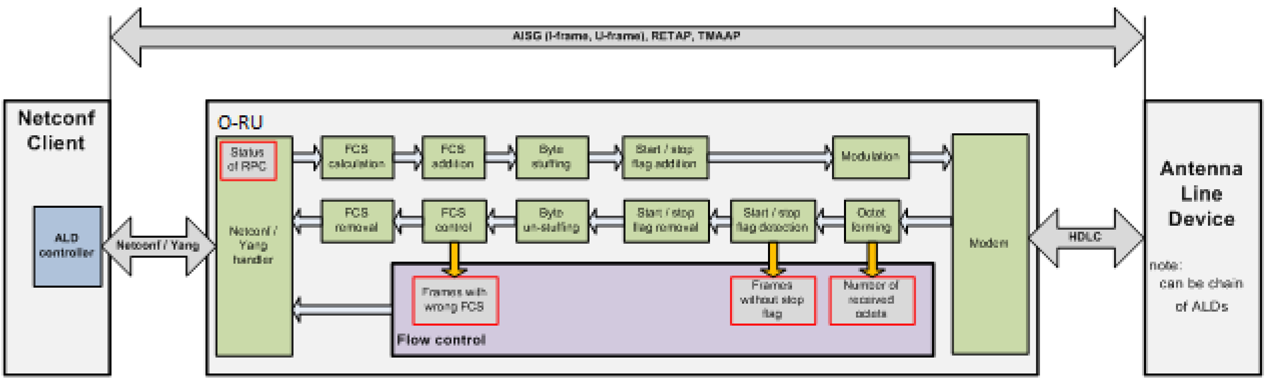


Figure 44: ALD Reference Architecture

The result of the above architecture is, that below mentioned parts of HDLC message are processed by entities as illustrated in Figure 45:

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

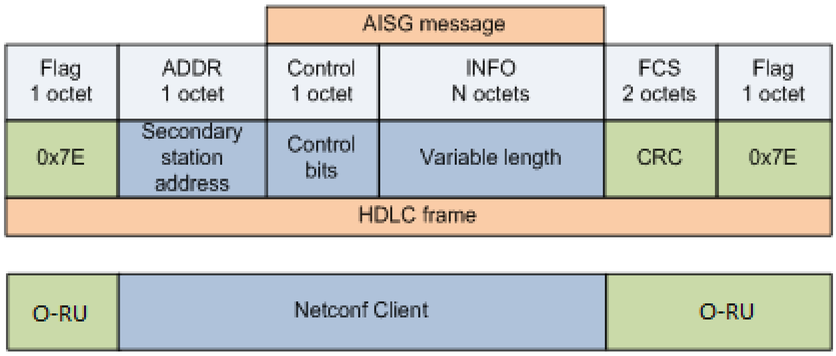


Figure 45: Component's responsibility split.

11.3.2 ALD Operations

The NETCONF Client sends RPC <ald-communication> to the O-RU. The RPC has following input parameters:

- leaf: **ald-port-id** (uint8) - contains the identity of the ALD port. The O-RU shall output the data to (corresponds to O-RU resources provided to NETCONF Client as inventory information)
- leaf: **ald-req-msg** (up to 1200 bytes) - may contain HDLC address, control bits and payload (see: TS 25.462 for details)

The O-RU performs HDLC communication with the ALD as follows: immediately after the requested payload is sent to the ALD over the desired ALD port, the O-RU switches the ALD port into reception mode.

Note, for details of HDLC transmission and reception algorithm, please see TS 25.462, chapter 4.5 "Message timing". Bits received within reception window are formed to octets and inserted as payload into ald-resp-msg.

The O-RU responds to the NETCONF Client using the <rpc-reply> message containing following parameters:

- leaf: **ald-port-id** (uint8)
- leaf: **status**
- leaf: **ald-resp-msg** (up to 1200 bytes)
- leaf: **frames-with-wrong-crc** (4 bytes)
- leaf: **frames-without-stop-flag** (4 bytes)
- leaf: **number-of-received-octets** (4 bytes)

Note: In case there is no response from the ALD received within the reception window, the record "**ald-resp-msg**" in <rpc-reply> sent by the O-RU shall be empty.

After reception, the O-RU shall wait an additional 3ms before the next transmission towards HDLC bus is initiated. See TS 25.462, chapter 4.5 "Message timing" for details.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

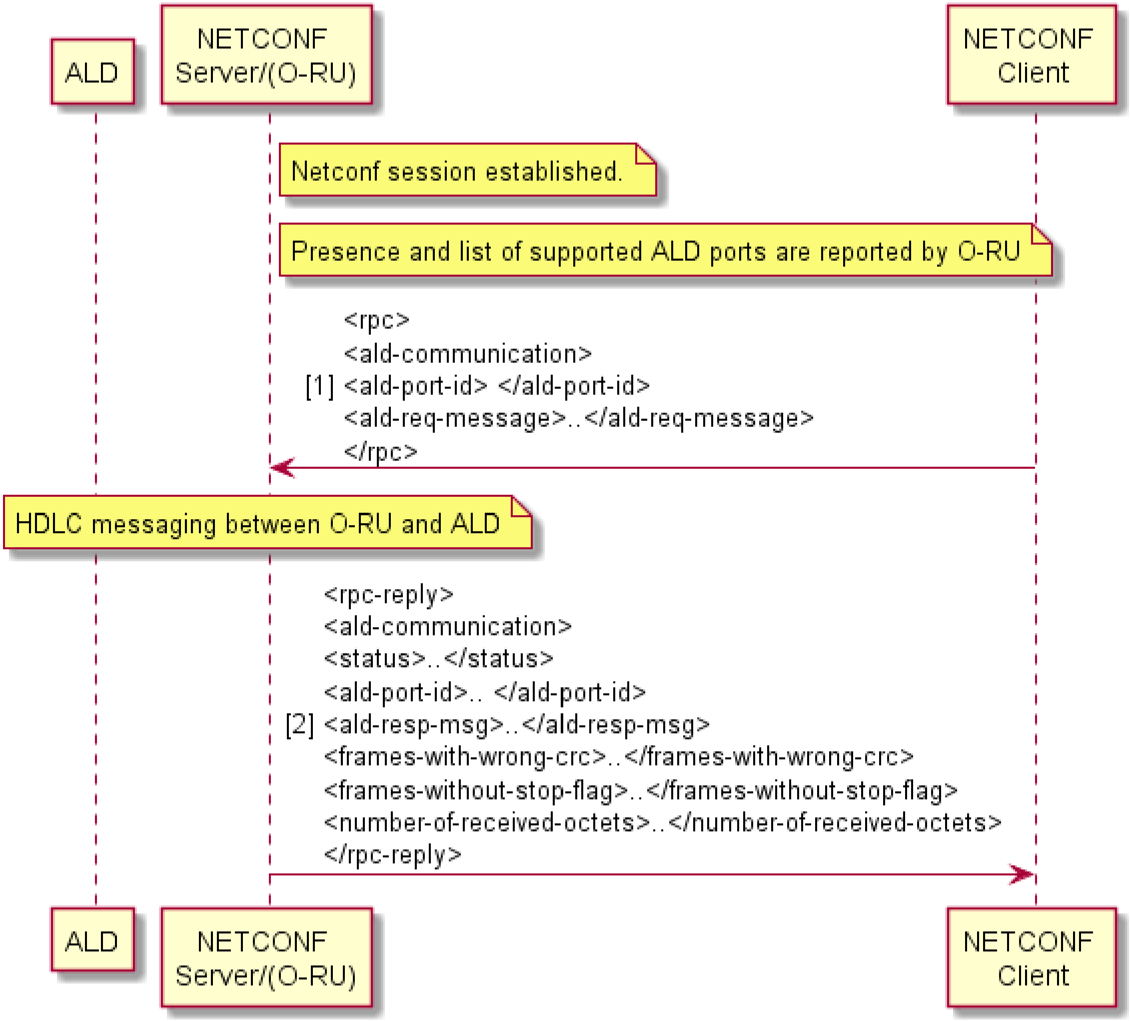


Figure 46: ALD Message Transfer

General scenario

<u>Precondition:</u>

M-Plane connectivity between NETCONF Client and NETCONF Server is successfully established. NETCONF Server reports presence of the supported HDLC Primary Devices.

1) NETCONF Client triggers DC voltage on desired ALD ports using NETCONF <edit-config> RPC. After DC is turned on - NETCONF Client waits 3s.

2) NETCONF Client performs HDLC link speed alignment to assure that all ALDs connected to a particular port have switched themselves to the correct baud rate used by this port.

3) NETCONF Client performs HDLC bus scan using desired HDLC Primary Device offered by O-RU.

3) NETCONF Client determines presence of HDLC Secondary Devices.

4) NETCONF Client assigns HDLC addresses to desired HDLC Secondary Devices.

5) NETCONF Client initiates HDLC layer for secondary devices by sending SNRM command.

6) NETCONF Client starts polling procedure for every HDLC-addressed Secondary Device.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Postcondition:

Detected and addressed HDLC Secondary Devices are available for configuration.

11.4 Operational aspects of external IO

An O-RU can connect to one or more input and output ports for external device supervision and control.

The External IO has the following functions

- INPUT: Supervising external devices
- OUTPUT: Controlling external devices

Also, external IO function include signalling to get the O-RU and O-RU controller in sync, enables port monitoring on the O-RU and provides notification from the O-RU to an O-RU controller, and provides control from an O-RU controller to the O-RU and enables output port controlling on the O-RU.

The O-RU only implements external IO yang module if the O-RU supports External IO aspect.

A change in condition of the external IO shall not affect other O-RU services such as RF transmission / reception behaviour.

11.4.1 External input

This section explains single external input line case. For multiple external inputs case, same behaviour for each input shall be processed individually.

For input, the O-RU and O-RU controller shall support two scenarios.

[1] To retrieve input state from O-RU controller and respond the input state from the O-RU to O-RU controller.

[2] To send notification from the O-RU to O-RU controller when input state is changed.

The value shall be

TRUE: Circuit is open.

FALSE: Circuit is closed

When nothing is connected to the line the value shall be TRUE.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

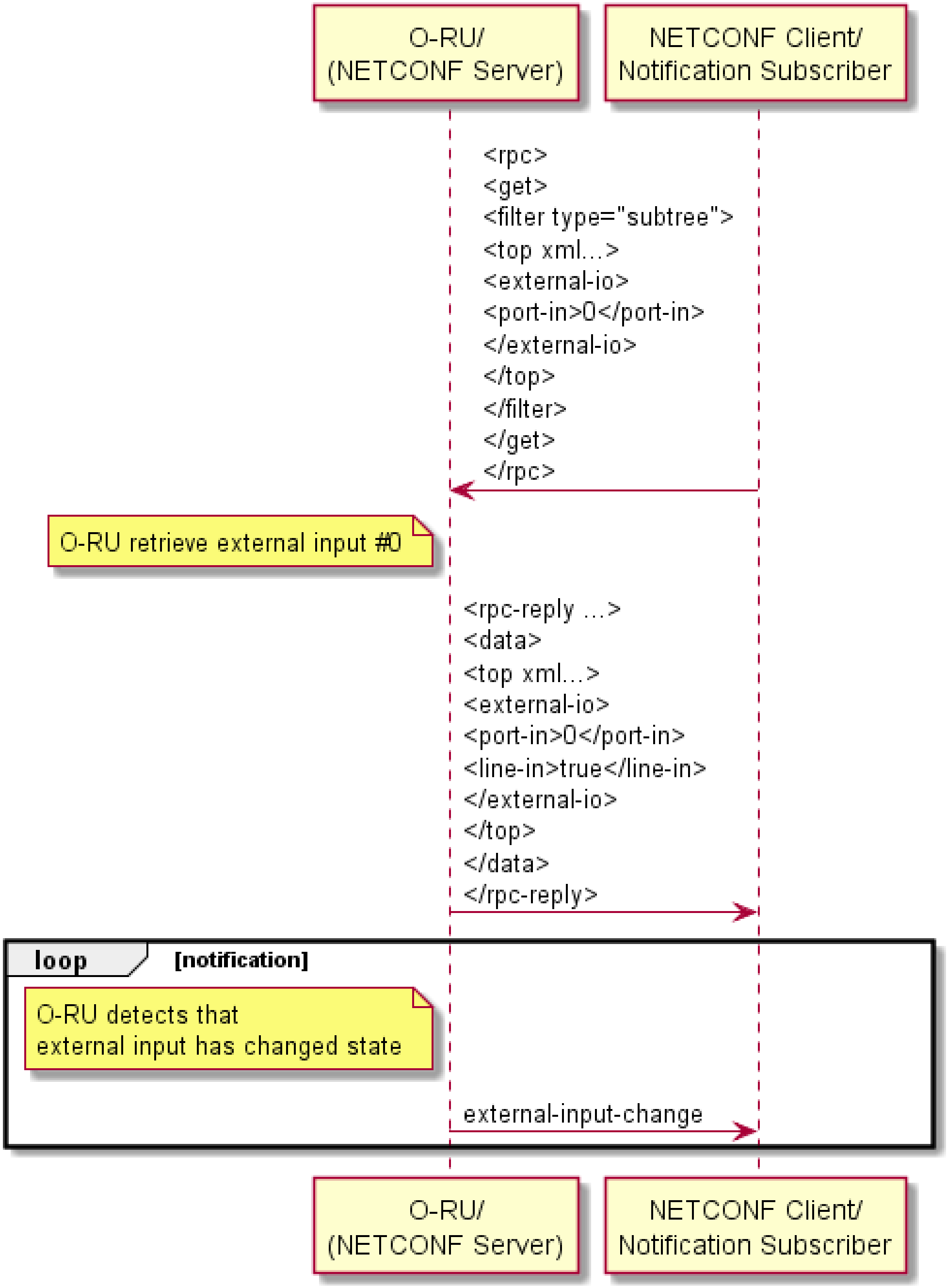


Figure 47: Retrieve external line-in

11.4.2 External output

This section explains single external output line case. For multiple external outputs case, same behaviour for each output shall be processed individually.

For output, the O-RU and O-RU controller shall support two scenarios.

[1] To retrieve output state from O-RU controller and respond the output state from the O-RU to O-RU controller.

[2] To send edit-config from the O-RU to O-RU controller when output state change is required.

The value shall be

TRUE: Circuit is open.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

FALSE: Circuit is closed

The default values shall be TRUE.

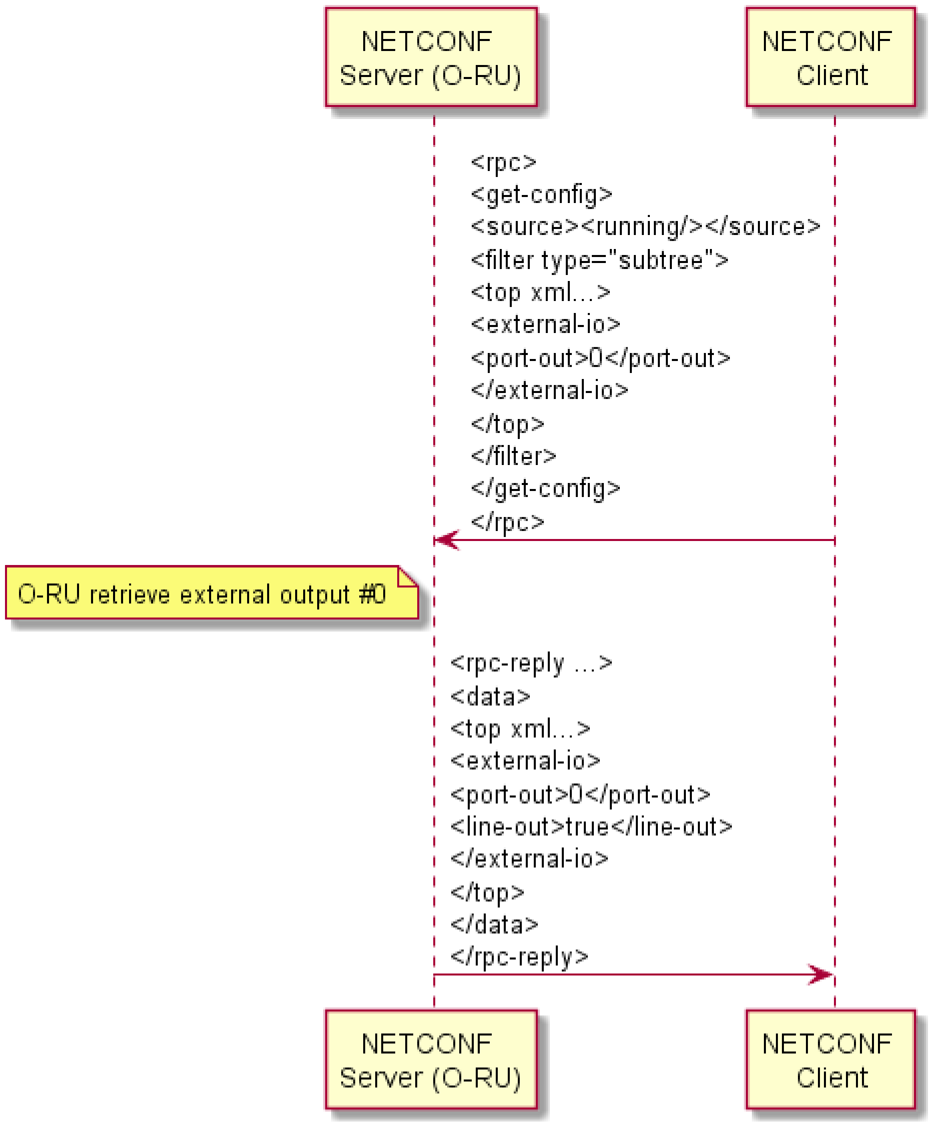


Figure 48: Retrieve external line-out

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

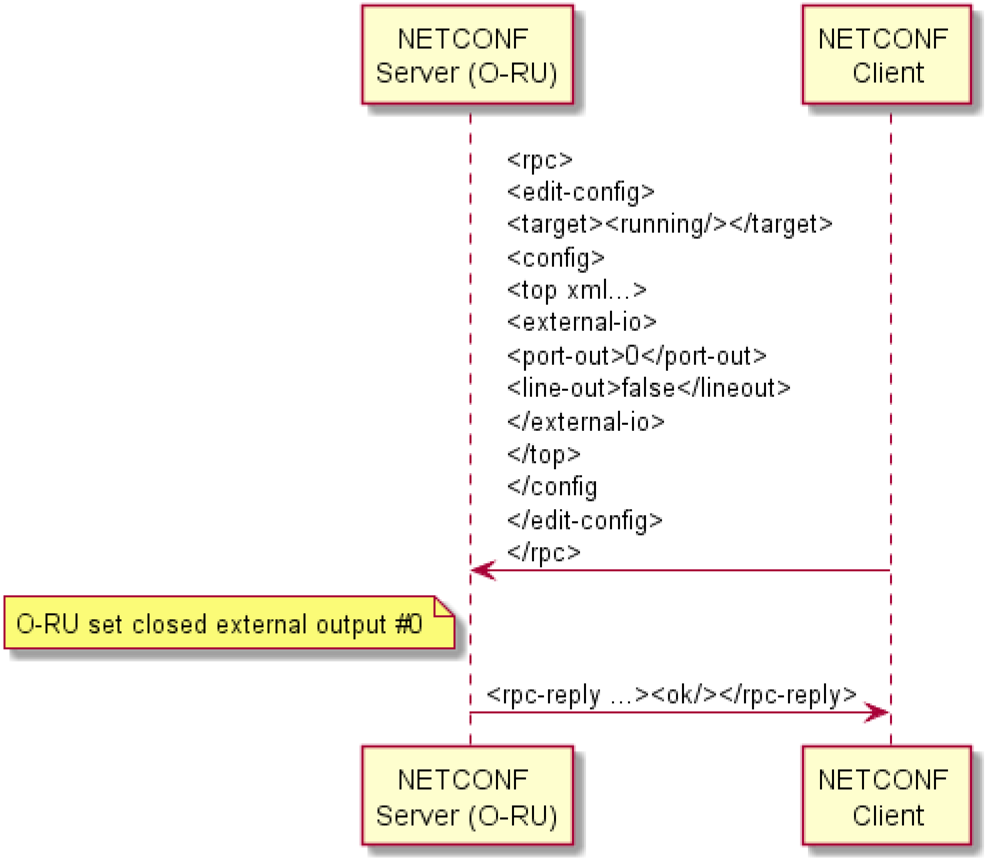


Figure 49: Control external line-out

Chapter 12 Details of O-RU Operations

12.1 Retrieval of O-RU Information

This sub-section provides handling for O-RU controller(s) to retrieve O-RU information from O-RU. The further actions such as SW Management, U-plane configuration and Performance Management will use these retrieved O-RU information.

The following information, for example, can be retrieved from the O-RU:

- hw/hardware/component
  - retrieve **mfg-name** – the name of the O-RU manufacturer
  - retrieve **serial-num** – the serial number of the O-RU
  - retrieve **software-rev** – the version of the O-RU software build
- o-ran-hardware/hardware/component/
  - retrieve **product-code** – the O-RAN defined product code
- o-ran-operations/operational-info/declarations
  - retrieve supported-mplane-version – the version of the O-RAN M-Plane interface
  - retrieve **supported-cusplane-version** – the version of the O-RAN CUS-Plane interface

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

retrieve **supported-header-mechanism** – the type of C/U plane headers supported by the O-RU

o-ran-operations/operational-state

retrieve **restart-cause** – the reason for the last restart

o-ran-sync/sync

retrieve **sync-state** – the synchronization state of the O-RU

The detail of O-RU information, please see corresponding YANG modules in Annex D.

12.2 User plane message routing

The purpose of U-Plane configuration is to define the relationship between U-Plane application endpoints in the O-DU and those in the O-RU. After such relationships are defined, the application endpoints are able to exchange IQ data using the U-Plane application protocol defined in [2].

Precondition:

- M-Plane connectivity is established between NETCONF Client and NETCONF Server

12.2.1 Configurable format for eAxC_ID

The eAxC_ID is used by C/U-plane application to manage eCPRI communication between desired C/U-Plane application components in O-DU and O-RU.

As defined in [2], the eAxC_ID consists of four parameters: DU-PORT, RU-PORT, CC-ID and BAND-SECTOR-ID. Order of parameters in eAxC_ID must follow definitions in CUS-Plane spec. In this version of the O-RAN WG4 specification, the length of eAxC_ID is constant and equal to 16 bits. To enable optimal sharing of the 16 bits between these four parameters, the assignment of eAxC_ID bits to parameters is not fixed. As a consequence, there is a need for NETCONF client to configure the bit assignment to parameters mappings using the M-Plane interface.

To handle flexible bit assignment, configurable bitmasks are defined for each parameter.

Note: flexible configuration means, that bits of eAxC_ID can be assigned to parameters in runtime.

Rules to be followed by NETCONF Client when configuring bit assignments:

- notation used for parameters forming eAxC_ID is from the LSB.

- each parameter uses consecutive bits

- each parameter can occupy 0-16 bits

- single bit of eAxC_ID cannot be assigned to more than one parameter

RPC **edit-config** shall be used to configure bit assignments to O-RU.

Bit assignment change for parameters related to an existing carrier is not allowed. (Impacted carriers need to be deactivated and deleted prior to any change in eAxC_ID configuration, and then be subsequently created and activated.)

An example of bit assignment usage where 3 bits are assigned to the BAND-SECTOR-ID, 3 bits for CC-ID, 7 bit for DU-PORT-ID and 3 bits for RU-PORT-ID is shown below:

<**du-port-bitmask**> 1111111000000000 </**du-port-bitmask**>

<**band-sector-bitmask**> 0000000111000000 </**band-sector-bitmask**>

<**ccid-bitmask**> 0000000000111000 </**ccid-bitmask**>

<**ru-port-bitmask**> 0000000000000111 </**ru-port-bitmask**>

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

12.2.2 U-Plane endpoint addressing

Parameter "eaxc-id" for low-level-tx-endpoint and low-level-rx-endpoint, defined using an unsigned 16 bit integer, shall follow the eaxc-id addressing schema defined in section 12.2.1. Please refer to the eaxc-id parameter description in CUS plane specification [2].

The NETCONF Client is responsible for assigning unique values to the "eaxc_id" addresses to all low-level-rx-endpoint elements and low-level-tx-endpoint elements, within the O-RU when operating in the same direction (Tx or Rx), even when these operate across different named interfaces of the O-RU.

More precisely, the same eaxc-id cannot be simultaneously assigned to multiple -low-level-rx-endpoints or to multiple and low-level-tx-endpoints. Clarifying eaxc-id assignment by example, and unless otherwise specified, within the same O-RU (considering a 2 × 2 MIMO):

Case 1: Allowed eaxc-id assignment (same antenna ports used for Tx and Rx)

- low-level-rx-endpoint (name 1) – eaxc-id=1
- low-level-tx-endpoint (name 1) – eaxc-id=1
- low-level-rx-endpoint (name 2) – eaxc-id=2
- low-level-tx-endpoint (name 2) – eaxc-id=2

Case 2: Allowed eaxc-id assignment (separate antenna ports used for Tx and Rx)

- low-level-rx-endpoint (name 1) – eaxc-id=1
- low-level-tx-endpoint (name 1) – eaxc-id=2
- low-level-rx-endpoint (name 2) – eaxc-id=3
- low-level-tx-endpoint (name 2) – eaxc-id=4

Case 3: Prohibited eaxc-id assignment (separate antenna ports used for Tx and Rx)

- low-level-rx-endpoint (name 1) – eaxc-id=1
- low-level-tx-endpoint (name 1) – eaxc-id=2
- low-level-rx-endpoint (name 2) – eaxc-id=1
- low-level-tx-endpoint (name 2) – eaxc-id=2

12.2.3 General configuration scenario

Below is described the general scenario to be followed by a NETCONF Client in order to properly configure communication between C/U-Plane endpoints in the O-DU and O-RU.

All operations can be performed in any order (including combining some of them in one request) provided assumed result (overall configuration) of each request sent by NETCONF Client is valid. Note selected highlighted rules below:

- eaxc-id is unique for all endpoints within the O-RU in the same direction (Tx or Rx) and linked with any low-level-rx-link or low-level-tx-link element
- at the moment of creation, every low-level-rx-link must be linked to an existing rx-array-carrier element and existing processing-element element
- at the moment of creation, every low-level-tx-link must be linked to an existing tx-array-carrier element and existing processing-element element

1) NETCONF Client determines the presence of following elements offered by NETCONF Server:

- tx-arrays - by fetching the list of **tx-arrays** in o-ran-uplane-conf.yang
- rx-arrays - by fetching the list of **rx-arrays** in o-ran-uplane-conf.yang
- low-level-tx-endpoint elements - by fetching the list **static-low-level-tx-endpoints** in o-ran-uplane-conf.yang
- low-level-rx-endpoint elements - by fetching the list **static-low-level-rx-endpoints** in o-ran-uplane-conf.yang
- interface elements - by fetching list of **interfaces** in o-ran-interfaces.yang

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

2) NETCONF Client determines capabilities exposed by static-low-level-tx-endpoints and static-low-level-rx-endpoints. Additionally. NETCONF Client determines capabilities exposed by "endpoint-types" and "endpoint-capacity-sharing-groups" and specific parameters proprietary to [tr]x-array(s). Obtained information must be respected when NETCONF Client configures low-level-[tr]x-endpoints referenced to static-low-level[tr]x-endpoints by parameter "name".

3) For elements determined in step 1) NETCONF Client examines relationship between

- static-low-level-tx-endpoint elements and tx-array elements in o-ran-uplane-conf.yang
- static-low-level-rx-endpoint elements and rx-array elements in o-ran-uplane-conf.yang
- static-low-level-tx-endpoint elements and interface elements
- static-low-level-rx-endpoint elements and interface elements
- tx-arrays, rx-arrays and their elements in o-ran-uplane-conf.yang.

Note: Netconf Client retrieves the content of o-ran-beamforming,yang module to obtain knowledge regarding beamforming-related parameters that apply for particular Netconf Server. This step is optional, as o-ran-beamforming.yang module exists only in case Netconf Server supports beamforming. Obtained parameters are needed by Netconf Client to perform beamforming control.

4) For every static-low-level-rx-endpoint NETCONF Client determines endpoint's ability to support non-time managed and/or time managed traffic. Information about delayed traffic type supported by endpoints is exposed through parameter **managed-delay-support** (enumeration) under endpoint-types and it indicates whether the endpoint can support time managed traffic (MANAGED), non-time managed traffic (NON_MANAGED), or both (BOTH). It is required that the desired type of supported traffic to be configured to the endpoint. Configuration is assumed to be static for run-time. Configuration is applicable with "**non-time-managed-delay-enabled**" (Boolean) parameter of low-level-rx-endpoint related by "name" to static-low-level-rx-endpoint exposing endpoints ability to support non-time managed traffic. Default value of this parameter is FALSE, meaning endpoint supports time managed traffic by default. For details see: Note 2.

5) NETCONF Client determines accessible low-level-rx-endpoint elements and low-level-tx-endpoint elements, including optional interface restrictions, that are suitable for the desired cell configuration (i.e. are linked with specific antenna arrays and are able to support desired type of traffic).

6) NETCONF Client performs C/U-Plane transport configuration between O-DU and O-RU. NETCONF Client configures interfaces and creates **processing-elements** related to the **interfaces** offering access to desired endpoints (suitable in terms of capabilities and able to process signals related with desired [tr]x-array). Details of configuring **interfaces** and **processing-elements** are described in Chapter 4.

7) Once transport layer is configured, O-DU may perform initial verification of C/U Plane Transport Connectivity as described in chapter 4.6 – with respect to content of list “restricted-interfaces” every desired endpoint is reachable through.

8) NETCONF Client assigns unique values to the eaxc-id(s) to the endpoints determined in step 1.

Note: uniqueness of eaxc-id is mandatory within the O-RU in the same direction (Tx or Rx) even across interface elements and having relationship to low-level-rx-endpoint elements or low-level-tx-endpoint elements.

Note 2: In case NETCONF Client wants particular value of eAxC_IDto be used for non-time managed traffic, NETCONF Client shall assign this eAxC_ID to parameter "eaxc-id" belonging to low-lever-rx-endpoint, that is capable to support non-time managed traffic (as per reference to capabilities exposed by corresponding static-low-level-rx-endpoint corresponding to low-level-rx-endpoint by name). When assigning eAxC_ID to convenient low-level-rx-endpoint, NETCONF Client must also configure the low-level-rx-endpoint to work in non-time-managed mode (when applicable - see: 4)). Change between types of traffic configured to low-level-rx-endpoint shall not be requested by NETCONF Client in case there is traffic served by endpoint that is a subject of reconfiguration.

9) NETCONF Client creates **tx-array-carrier**(s) and **rx-array-carrier**(s). The **tx-array-carriers** and **rx-array-carriers** can be configured with **type** set to **LTE**, **NR** or **DSS-LTE-NR**. The configuration of array carrier with type **DSS-LTE-NR** is only allowed when the O-RU supports Dynamic Spectrum Sharing (DSS) feature as indicated by feature **DSS_LTE_NR** in o-ran-module-cap.yang. If the O-RU indicates it supports the feature DSS_LTE_NR but does not support Section Extension 9, then instead of configuring a carrier as DSS-LTE-NR, the O-DU must configure DSS by using different eAxC ids (i.e. different endpoints) as described in CUS plane specification [2].

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Type	N_{RB}	Center of channel bandwidth (same as F_{REF} as defined in 3GPP TS38.104 Section 5.4.2.1)
LTE or DSS	N_{RB}mod2=1	Between (k-1) RE and k RE of n_{PRB} RB
	N_{RB}mod2=0	Between the highest RE of (n_{PRB}-1) RB and k RE of n_{PRB} RB
NR	N_{RB}mod2=1	Center of k^{th} RE of n_{PRB} RB
	N_{RB}mod2=0	

Table 5: Center Bandwidth Calculation

The parameters k, n_{PRB} and N_{RB} referenced here are defined in Table 5.4.2.2-1 in 3GPP TS 38.104.

10) NETCONF Client performs C/U-Plane transport configuration between O-DU and O-RU. NETCONF Client configures interfaces and creates **processing-elements** related to the **interfaces** offering access to desired endpoints (suitable in terms of capabilities and able to process signals related with desired [tr]x-array). Details of configuring **interfaces** and **processing-elements** are described in Chapter 4.

11) NETCONF Client creates low-level-[tr]x-link(s) to make relationship between low-level-[tr]x-endpoint(s), [tr]x-array-carriers and processing elements belonging to transport. Respective TX path and RX path linkage must be followed.

Note: C/U-Plane traffic can be prioritized by reference "**user-plane-uplink-marking**" indicated by low-level-rx-link in o-ran-uplane-conf.yang. The reference is to o-ran-processing-element.yang, where it is linked to "**up-marking-name**". Further the **up-marking-name** points to o-ran-interfaces.yang, where it ends up pointing to priority depending on actually used u-plane transport (either PCP for Ethernet or DSCP for IP). For details regarding priorities see: ref [2], section 3.3 "Quality of Service".

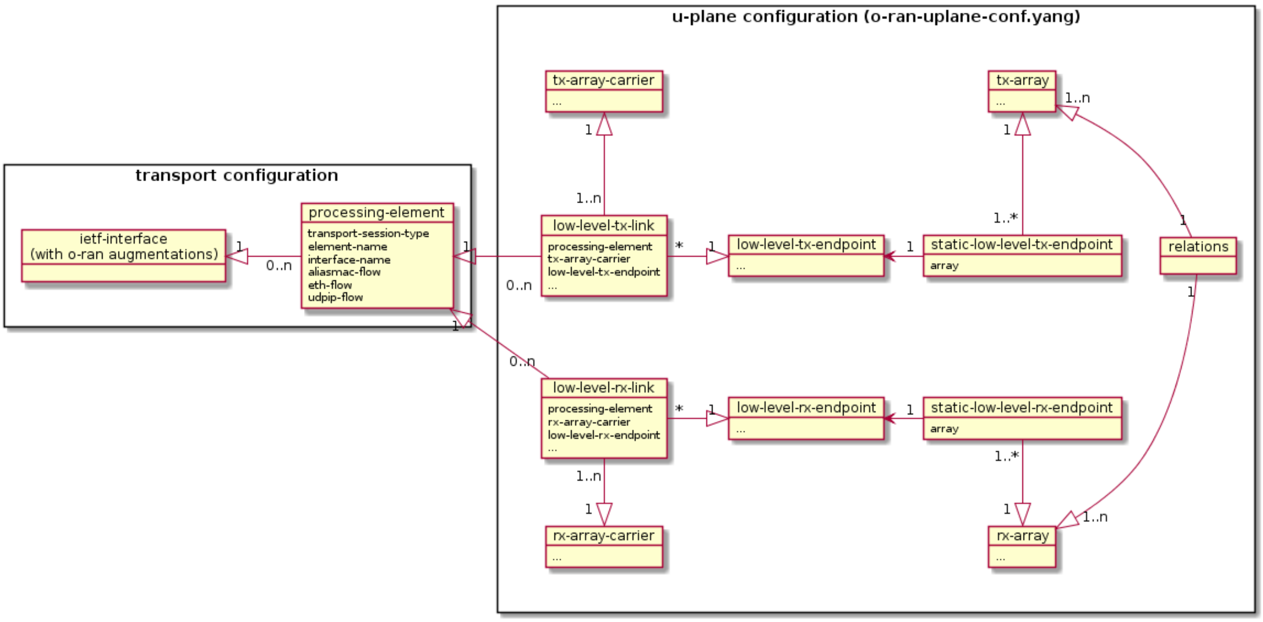


Figure 50: Diagram showing relations between CU-Plane and Carrier configuration elements

For detailed content of objects shown in "u-plane configuration" box on above diagram, please examine o-ran-uplane-conf.yang module.

After steps above carrier configuration scenario can be started. This is described in subsection 12.3.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

12.3 Carrier Configuration

12.3.1 Carrier creation

This section provides basic scenario for carrier creation procedure. Precondition for below steps is to fulfil steps from subsection 12.2.3

1) NETCONF Client creates the **tx-array-carriers** in relation to the desired **tx-arrays**. Note that generally number of **tx-array-carriers** will be the same as multiple of the desired number of **tx-arrays** and the number of component carriers.

2) NETCONF Client creates the **rx-array-carriers** in relation to the desired **rx-arrays**. Note that generally number of **rx-array-carriers** will be the same as multiple of the desired number of **rx-arrays** and the number of component carriers.

3) NETCONF Client creates the **processing-elements** related to **interfaces** offering access to endpoints.

4) NETCONF Client creates the **low-level-tx-links** containing relationship to the existing **tx-array-carriers**, **low-level-tx-endpoints** and existing **processing-elements**.

5) NETCONF Client creates the **low-level-rx-links** containing the relationship to existing **rx-array-carriers**, **low-level-rx-endpoints** and existing **processing-elements**.

With the above steps successfully performed, the relationship between C/U-Plane application endpoints at O-DU and O-RU is configured.

12.3.2 Activation, deactivation and sleep

The NETCONF Client performs activation by setting the value of the parameter “active” at tx-array-carrier element / rx-array-carrier element to “ACTIVE”.

The NETCONF Client performs deactivation by setting the value of the parameter “active” at tx-array-carrier element / rx-array-carrier element to “INACTIVE”

Communication between related U-Plane endpoints is enabled under condition, that for corresponding tx-array-carrier or rx-array-carrier value of parameter “active” is “ACTIVE” and value of parameter “state” is “READY”. Otherwise communication is disabled.

The NETCONF Client can put the tx-array-carrier / rx-array-carrier to sleep by setting value of parameter “active” in the corresponding tx-array-carrier element / rx-array-carrier element to “SLEEP”.

A particular tx-array-carrier / rx-array-carrier is in sleep mode when value of its parameter “active” is “SLEEP” and value of its parameter “state” is “READY”.

For detailed description of tx-array-carriers and rx-array-carriers please refer to *description* substatement in YANG models.

Below diagram shows possible transitions and values combination to be followed by "active" and "state". Combination or transitions outside of below diagram is not allowed.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

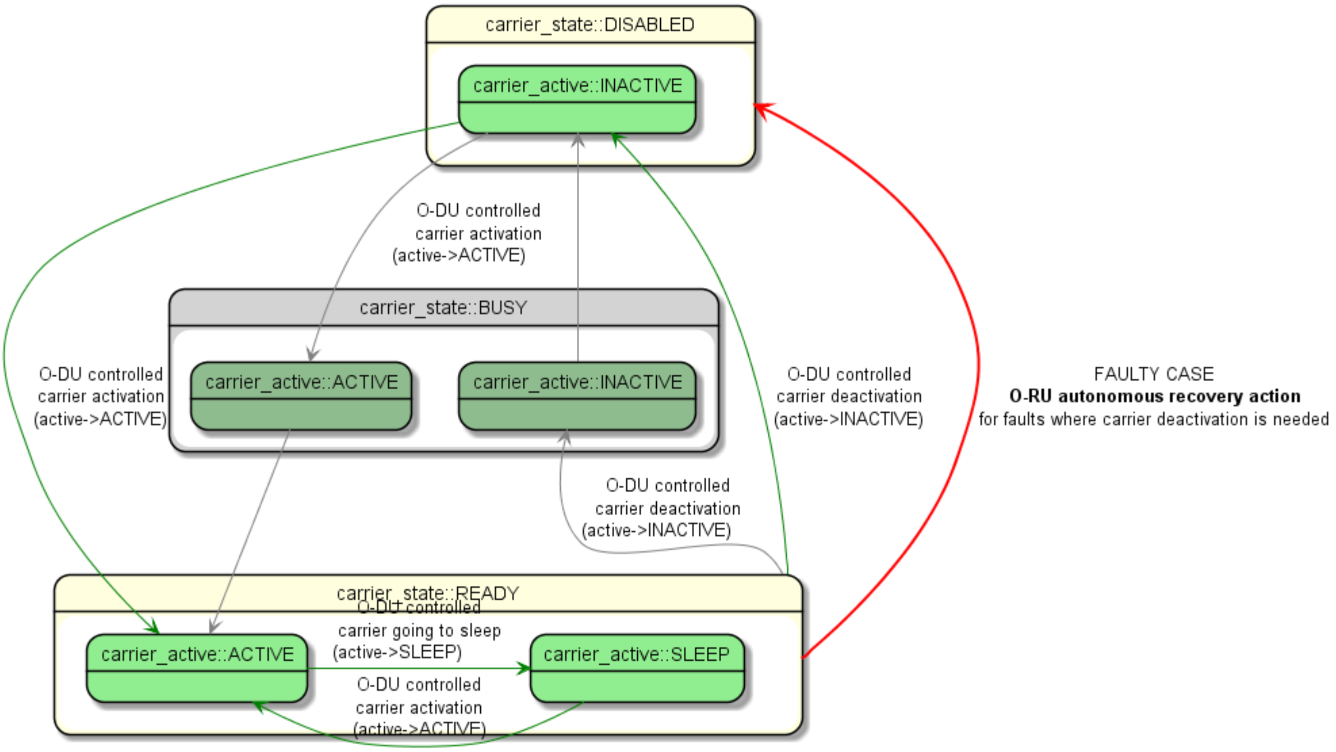


Figure 51: Diagram showing all possible transitions and combination of "active" and "state" parameters

Note: BUSY state is only available during transition and existence of this state depends on internal O-RU design.

12.3.3 Carriers relation to sync

tx-array-carrier and rx-array-carrier possible states and transitions are the same for sync LOCKED and HOLDOVER mode. When O-RU is working in FREERUN mode the only possible tx-array-carrier and rx-array-carrier state is DISABLE/INACTIVE. Below diagram shows possible transitions and according parameters values combination.

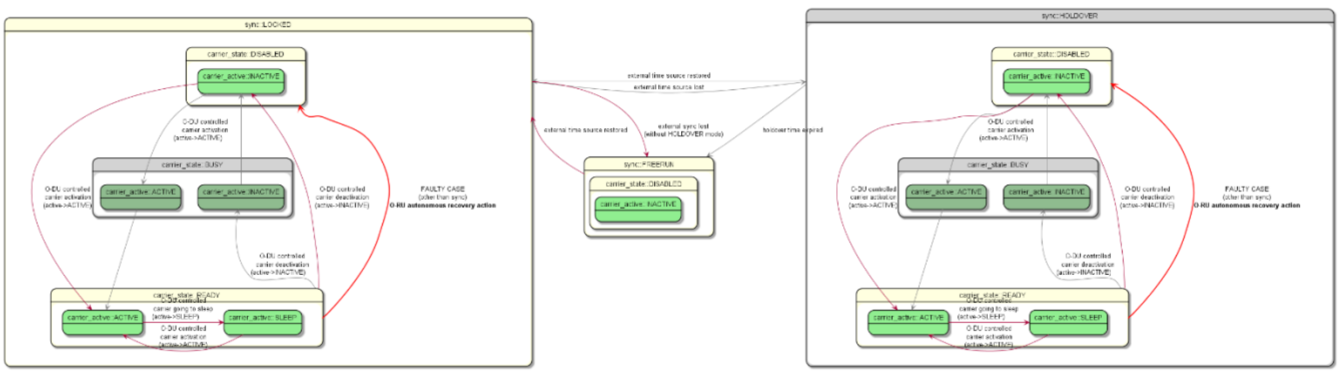


Figure 52: Combination of tx-array-carrier/rx-array-carrier transitions and allowed states compared to sync state

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

12.3.3.1 Synchronization lost and HOLDOVER mode expired

This section explains tx-array-carrier and rx-array-carrier behaviour when O-RU loose synchronisation and enters to FREERUN mode.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

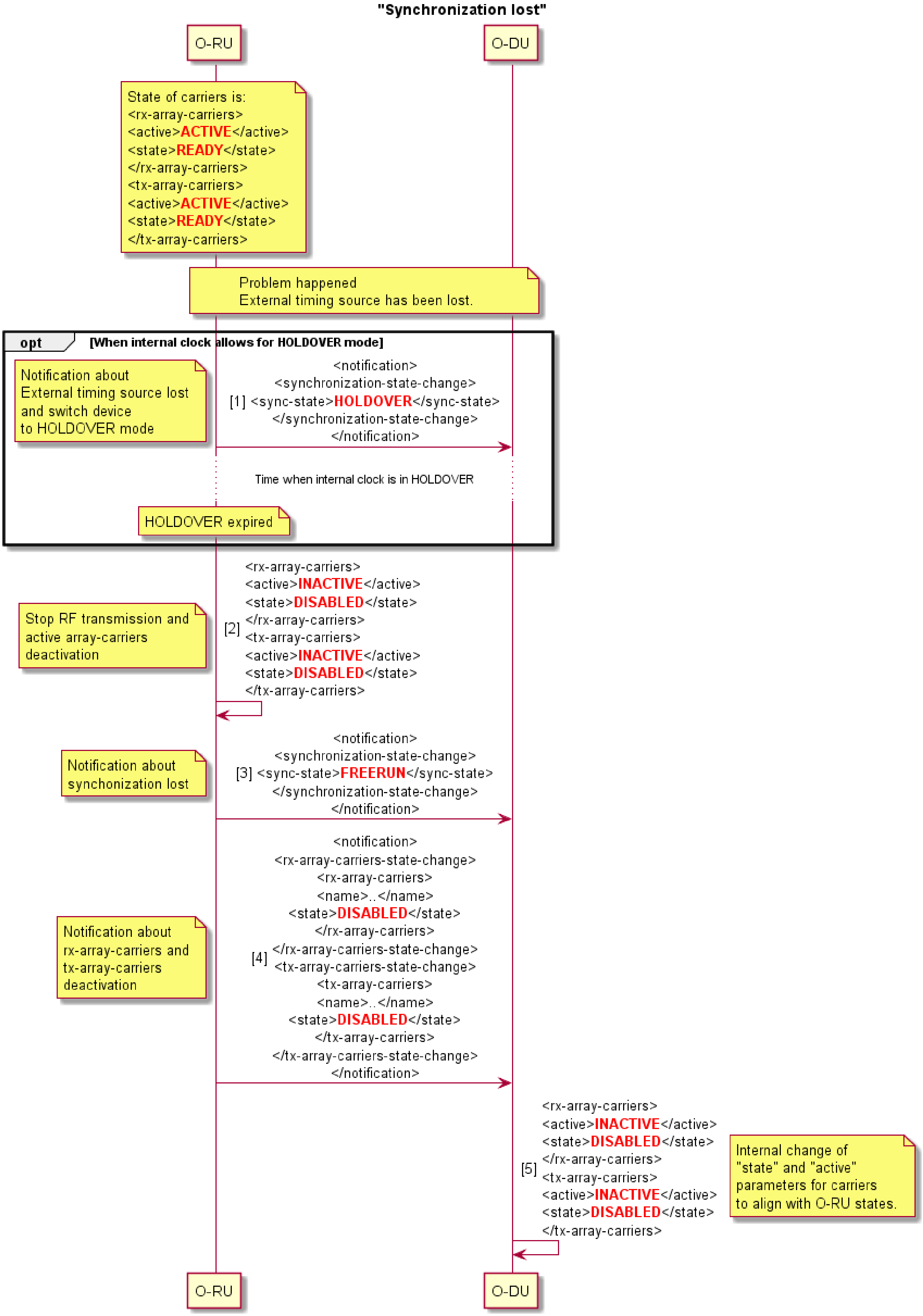


Figure 53: Synchronization lost scenario

12.3.3.2 External timing source restored

This section explains how tx-array-carrier and rx-array-carrier can be reactivated when external timing source is restored.

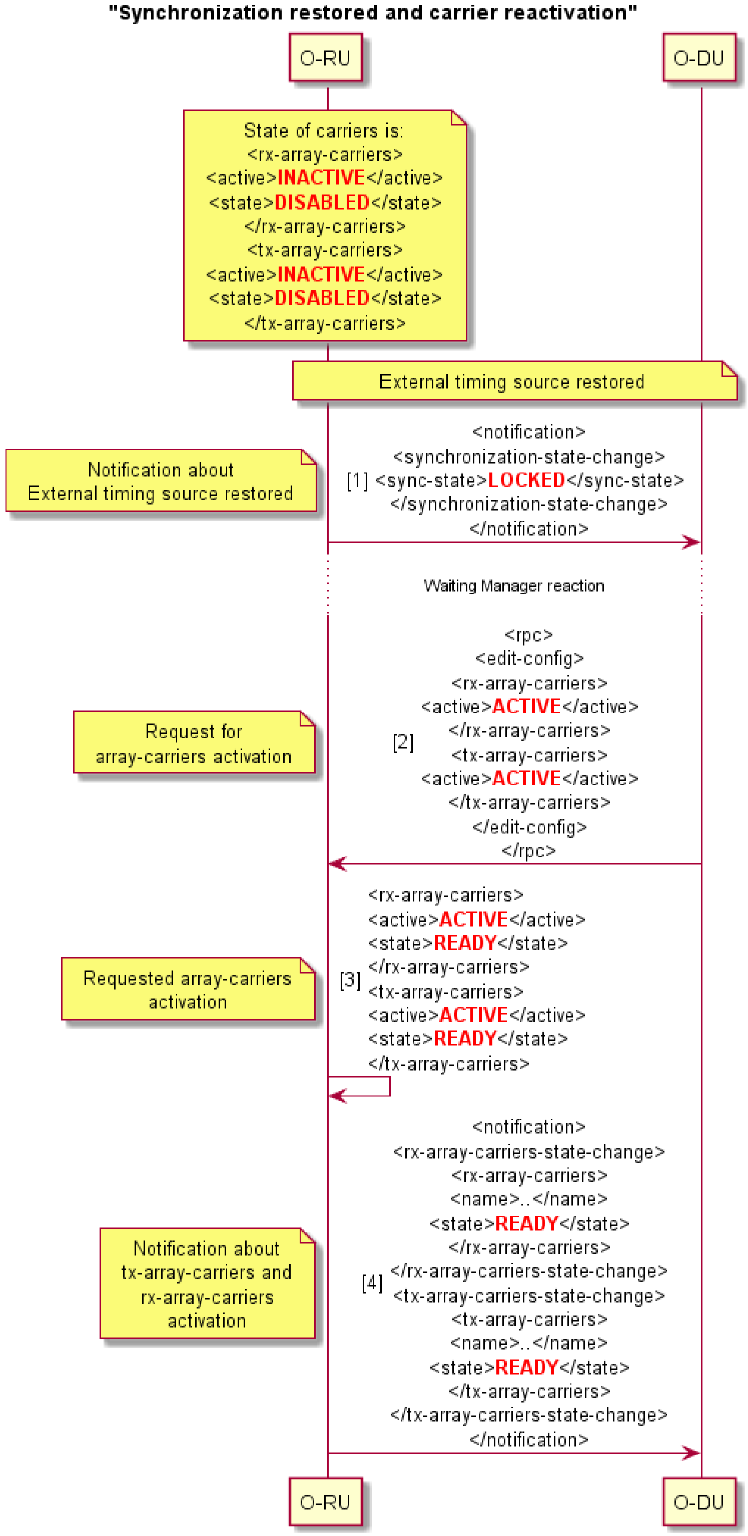


Figure 54: Synchronization restored and carrier reactivation scenario

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

12.4 Beamforming Configuration

The beamforming functionality allows the O-RU to influence the angle of the main lobe of the signal which is radiated from/received by the O-RU. Beamforming support is optional and an O-RU shall indicate that it supports such functionality by indicating that it supports the “urn:o-ran:beamforming:x.y” namespace.

A multi-band capable O-RU shall be able to support independent beamforming configuration on each of its supported **tx-arrays** and/or **rx-arrays** depends on O-RU antenna configuration.

12.4.1 Pre-Defined Beamforming Configuration

In case the O-RU supports beamforming, the o-ran-beamforming.yang and o-ran-uplane-conf,yang modules are used to report the pre-defined relationship between supported beams to a NETCONF client. A **band-number** and/or **capabilities-group** is used to uniquely identify separate **tx-arrays** and/or **rx-arrays** supported by an O-RU with the beamforming configuration referencing the set of **tx-arrays** and **rx-arrays** that are associated with this band and/or **capabilities-group**.

A default service area of the O-RU is determined as the grid of pre-defined beams. When O-RU updates beamforming configuration as described in 12.4.2, the grid of pre-defined beams can be newly defined. In this case, the default service area is changed accordingly.

O-RU may support new service area by applying tilt-offset to the given default service area in elevation and/or azimuth domains as described in 12.4.3.

12.4.2 Beamforming Configuration Update

This section provides the method to modify and to apply the beamforming configuration (weights, attributes and/or beam properties). The modification of the beamforming information is allowed only if O-RU supports the feature “MODIFY-BF-CONFIG” used for defining the modification of beamforming configuration.

The beamforming configuration is stored in the O-RU and comes from the O-RU’s software, treated in Chapter 5. The O-RU shall locate the beamforming configuration file in the generic folder, i.e., O-RAN/beamforming/.

To modify the beamforming configuration, the following steps are applied.

1. NETCONF client can retrieve the file list of the O-RU’s folder: O-RAN/beamforming/.

2. NETCONF client can trigger the upload of the beamforming configuration file from the O-RU’s folder.

3. Operator can recover the uploaded file and edit the beamforming configuration file offline.

4. NETCONF client can download the file to the original folder.

The modified beamforming configuration file shall not have the same name as any other file in the folder. Its file name is the matter of implementation.

The beam properties in o-ran-beamforming YANG module contain **coarse-fine**, **coarse-fine-beam-relation** and **neighbor-beam** for each **beam-id**. This information is received from the O-RU as O-RU's capability at O-RU start-up and typically are used by the scheduler in O-DU. A NETCONF client (O-RU Controller) can modify the beamforming information via file described in this section. When the beamforming configuration (weight, attribute and beam properties) is modified via file, the configuration of the beam properties list in the o-ran-beamforming YANG module should be modified together via the same file if affected by the modified weight and/or attribute.

An O-RU supporting the modification of beamforming configuration shall support the storage of at least two beamforming files per simultaneous **band-number** and/or **capabilities-group** supported. For each band within a multi-band O-RU or each **capabilities-group**, one file corresponds to the pre-defined (factory, read-only) beamforming configuration and at least one file corresponds to a modified (read-write) beamforming file. The O-RU has the responsibility to remove existing file and prepares space for new file when the NETCONF client **file-download** rpc is issued. When the O-RU only supports the storage of a single modified (read-write) beamforming file per band of operation, i.e., **number-of-writable-beamforming-files** = 1 the **file-download** operation for the modified beamforming configuration needs to be done while

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

neither **tx-array-carriers** nor **rx-array-carriers** are configured in the O-RU to avoid the removal of the modified beamforming configuration file for the current active software.

If the O-RU supports the capability to store two or more modified beamforming configuration files per band of operation in the O-RU, i.e., **number-of-writable-beamforming-files** > 1, the NETCONF **file-download** operation can be performed without any timing limitation. That's because the modified beamforming configuration file for the current beamforming configuration can be kept during the **file-download** operation. To apply the new modified beamforming configuration, the following steps are applied:

1. The NETCONF client can download the file to the beamforming folder if the O-RU supports the capability **number-of-writable-beamforming-files** > 1.
2. The NETCONF client shall deactivate **tx-array-carriers** and **rx-array-carriers** in the U-Plane configuration by setting "INACTIVE" for the **active** parameters, if they are ACTIVE.
3. Optionally, the NETCONF client shall delete **tx-array-carriers** and **rx-array-carriers**, if O-RU doesn't support the capability **update-bf-non-delete**.
4. Alternatively, the NETCONF client can trigger the download of the modified beamforming configuration file to the folder if the O-RU's capability is **number-of-writable-beamforming-files** =1.
5. The NETCONF client shall activate the modified beamforming configuration by using:
    - **activate-beamforming-config** rpc and selecting the modified beamforming configuration file and the **band-number** for which this modified configuration applies,
    - **activate-beamforming-config-by-capability-group** rpc and selecting the modified beamforming configuration file and the **capabilities-group** for which this modified configuration applies.
6. If a NETCONF client subscribes to the notification **beamforming-information-update** and/or **capability-group-beamforming-information-update** in advance, the O-RU sends such notification to the notification subscriber. Then the NETCONF client can subsequently retrieve beam properties in o-ran-beamforming YANG module via NETCONF <**get**> operation.
7. Optionally, the NETCONF client shall create **tx-array-carriers** and **rx-array-carriers** again, if the O-RU doesn't support the capability **update-bf-non-delete**.
8. The NETCONF Client shall activate **tx-array-carriers** and **rx-array-carriers** in the U-Plane configuration by setting "ACTIVE" for **active** parameters.

Then the new edited beamforming information is applied to the new **tx-array-carriers** and **rx-array-carriers** in the U-Plane configuration.

[Abnormal handling] If the O-RU fails to activate the edited beamforming configuration file correctly, i.e., rpc error for rpc **activate-beamforming-config** or **activate-beamforming-config-by-capability-group**, the O-RU shall revert back to the pre-defined/factory beamforming configuration file and report this to the NETCONF client.

At the **reset** rpc, the beamforming configuration information is switched to the pre-defined beamforming configuration. Even though the reset operation is issued, the O-RU may store the modified beamforming configuration file in the folder, which is not used, if O-RU supports the capability **persistent-bf-files** to store them in the reset-persistent memory.

The file format of the beamforming configuration is O-RU implementation specific.

The diagrams below show two methods to modify the file of beamforming configuration information plus the method how to apply the modified file for beamforming configuration conformation to use.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

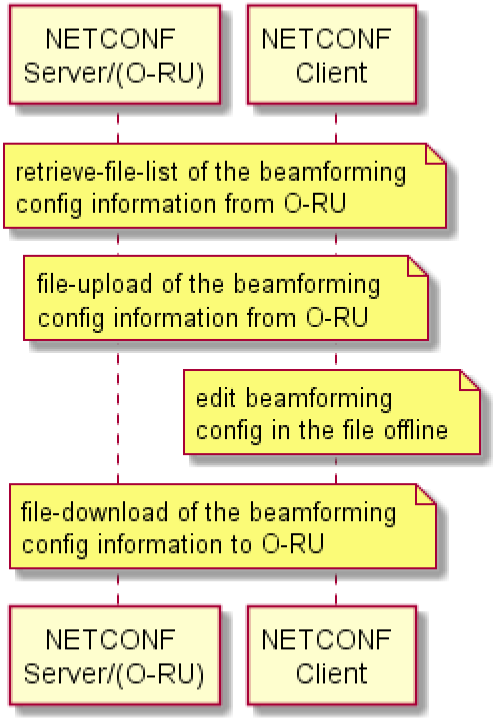


Figure 55: Method to Modify the File of Beamforming Configuration Information

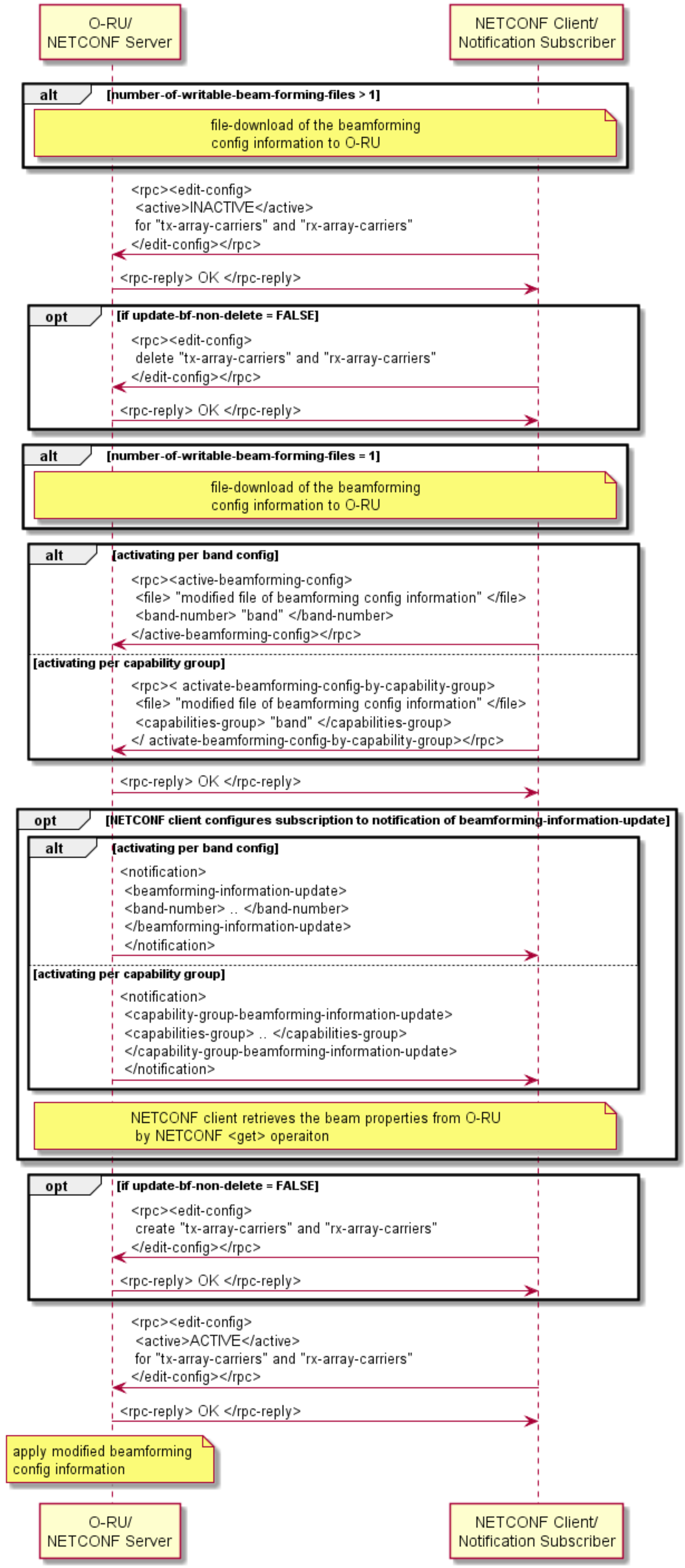


Figure 56: Method to Apply the modified file for Beamforming Configuration Information

12.4.3 Tilting Pre-defined Beams

This section describes the optional capability by which the O-RU's pre-defined beams may be tilted by using the "**BEAM-TILT**" feature defined in the o-ran-beamforming YANG model. This capability is an O-RU specific functionality, enabling adaptation of the service area associated with an O-RU without the need for operation of additional ALDs described in Chapter 11, or modifying the beamforming configuration using the "**MODIFY-BF-CONFIG"** feature described in sub-section 12.4.2.

> Note, the operation of the feature "**BEAM-TILT**" is independent to the operation of the "**MODIFY-BF-CONFIG**". When the "**MODIFY-BF-CONFIG**" feature is used to define a new default service area, the "**BEAM-TILT**" feature may be used to apply tilt-offsets to the newly defined service area.

O-RU can change the service area by applying a tilt-offset to the elevation and/or azimuth pointing angles for the pre-defined beams. This feature allows to shift beam characteristic of all predefined-beams in elevation and/or azimuth direction (i.e. changing the service area or sector coverage) while preserving the beam adjacency among the beams within grid of beams.

> Note, **offset-elevation-tilt-angle** values smaller than 0 represents an up-shift of the default service area towards the zenith (i.e., corresponding to a decrease in zenith angle) and values larger than 0 represent a down-shift of the default service area away from the zenith (i.e., corresponding to an increase in zenith angle).

To shift service area of the O-RU in a different direction, O-RU controller shall check whether the O-RU supports feature BEAM-TILT feature during capabilities negotiation of the NETCONF session. In the case that O-RU supports the **BEAM-TILT** feature, at least one of the **elevation-tilt-granularity** and **azimuth-tilt-granularity** will be read as greater than zero value from O-RU. Tilting is a per band operation and hence the parameters are defined per band. If O-RU supports **BEAM-TILT** feature, O-RU controller can configure the values of **offset-elevation-tilt-angle** and/or **offset-azimuth-tilt-angle** the configuration values should meet the ranges and granularity information retrieved from **predefined-beam-tilt-offset-information**.

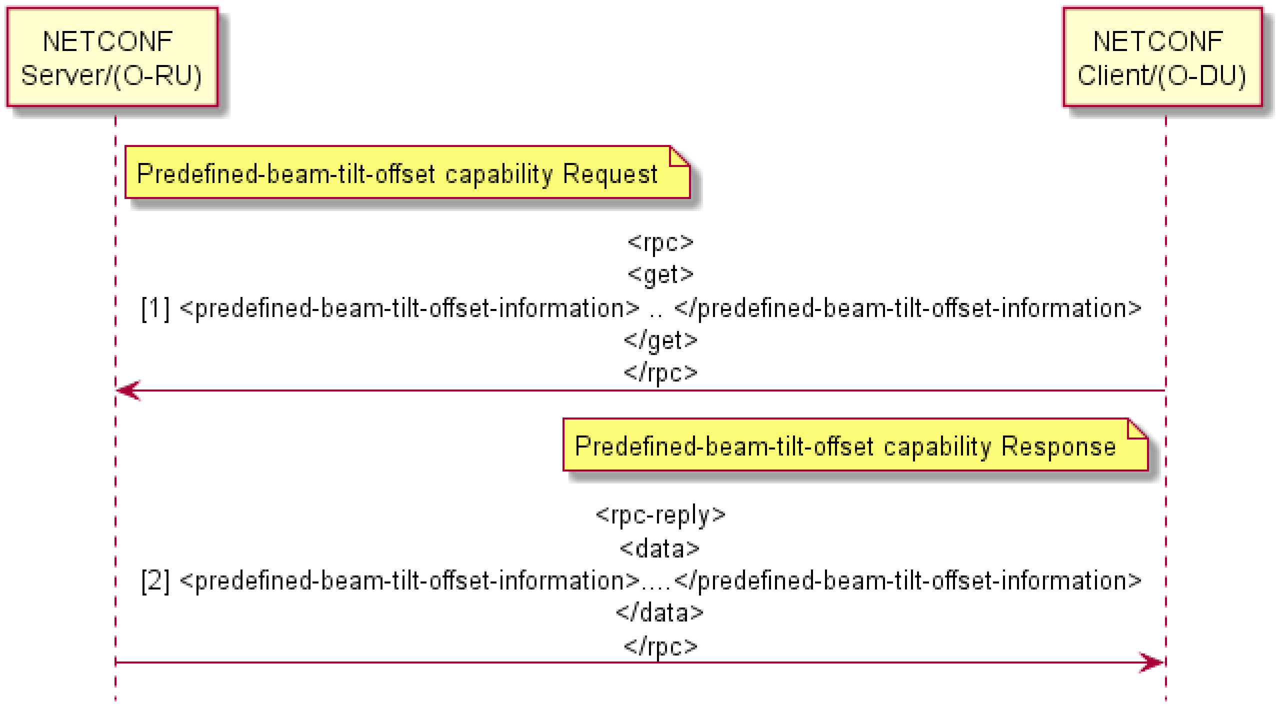


Figure 57: Sequence diagram for predefined-beam-tilt-offset-information

Depending on O-RU's implementation, the O-RU may need some time to complete the change of service area according to the updated **offset-elevation-tilt-angle** and/or **offset-azimuth-tilt-angle** for a particular **band-number**. The O-RU shall report its capability via the parameter, **run-time-tilt-offset-supported**. For O-RU with **run-time-tilt-offset-supported** = *FALSE*, changing the values in **offset-elevation-tilt-angle** and/or ***offset-azimuth-tilt-angle*** for a specific

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

band shall be allowed only if all **tx-array-carriers/rx-array-carriers** corresponding to the band is *INACTIVE*. When the service area change is completed in O-RU, the O-RU delivers the notification **predefined-beam-tilt-offset-complete** to inform the O-RU Controller which then may request to activate **tx-array-carriers/rx-array-carriers** in O-RU. For O-RU with **run-time-tilt-offset-supported** = TRUE, neither changing the state of **tx-array-carriers/rx-array-carriers** nor delivering notification **predefined-beam-tilt-offset-complete** is not required.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

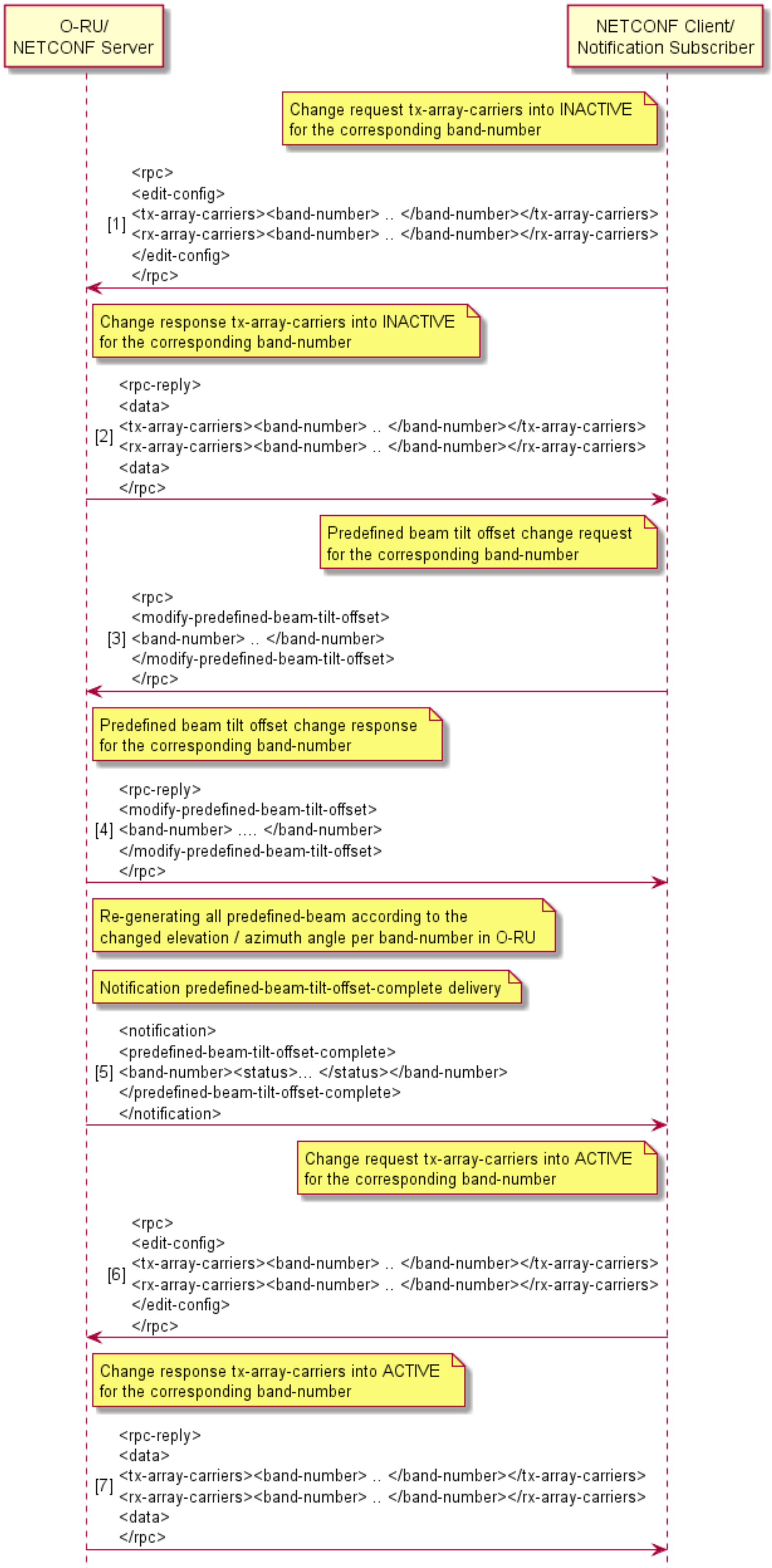


Figure 58: Procedure for the predefined-beam-tilt-offset

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

12.4.4 Dynamic Beamforming Control option

As option, O-RU may support dynamic beamforming control mode. Support for this type of beamforming control can be recognized—in the case of weights-based dynamic beamforming—from value of parameter rt-bf-weights-update-support (TRUE), and—in the case of attributes-based dynamic beamforming—from the value of parameter rt-bf-attributes-update-support (TRUE), in o-ran-beamforming.yang module

In dynamic beamforming control mode DU updates content of lookup table in O-RU using eCPRI C-Plane messages. For details of eCPRI messaging please see **Error! Unknown document property name.** spec, chapter 5.4.2 "Scheduling and Beamforming Commands".

Dynamically updated content of lookup table is further addressed by DU in the same way as it is done for static beamforming - by requesting particular Beam ID to be applied.

In case dynamic beamforming control is supported, O-RU indicates following supplementary information using parent leaf "static-properties" in o-ran-beamforming.yang module.

- beamforming type (frequency domain, time domain, hybrid)

- beamforming weight compression format (optional)

- available range of Beam IDs, that can be dynamically updated by DU.

- supported time and frequency granularity for time domain and hybrid beamforming control.

Note: Neighborhood relations between beams produced by beam IDs controlled by DU are unknown to O-RU, hence are not exposed.

In the case of weights-based dynamic beamforming, to properly calculate beamforming weights DU needs to know antenna array geometry. This information DU obtains by reading the content of o-ran-uplane-conf.yang (list of tx-arrays and rx-arrays with their child parameters). Details of beamforming weight calculations are not a subject for M-Plane activity and as such are intentionally not covered in this specification.

12.5 Antenna Calibration

Some antennas need to be calibrated to ensure their intended performance. Antenna calibration operation is an optional capability whose operation is dependent on O-RU design, i.e. different O-RUs may support different types of calibration – periodic vs on-demand, different calibration duration – short/medium/long, etc. In this section, a common framework is defined which can accommodate various types of antenna calibration implementations.

In this framework, the NETCONF client (O-DU) retrieves resource requirements for antenna calibration operation, e.g., timing and number of iterations/steps, from O-RU by getting the **antenna-calibration-capabilities** container defined in the o-ran-antenna-calibration YANG model. The O-DU can subscribe to **antenna-calibration-required** notifications to receive indications from the O-RU that calibration is required. When the O-RU indicates antenna calibration is required, or when the NETCONF Client decides to calibrate the O-RU, the NETCONF Client allocates time resources for antenna calibration and configures them in the O-RU using the **start-antenna-calibration** RPC request. The NETCONF client shall allocate the time resources for the calibration operation ensuring that these meet the minimum time necessary as reported by the O-RU using the **antenna-calibration-capabilities**. When available, the NETCONF client (O-DU) shall ensure that the frequency resources indicated in the **dl-calibration-frequency-chunk** and **ul-calibration-frequency-chunk** lists in the **antenna-calibration-required** notification are reserved for calibration operation, otherwise the NETCONF client shall consider that the full bandwidth of the carrier is being reserved for calibration operation.

The O-RU shall perform antenna calibration operation using the time resources allocated in the **antenna-calibration-start** RPC and frequency resources declared in **antenna-calibration-needed** notification and shall notify the completion of the antenna calibration operation to the notification subscriber. The O-DU should be configured to not schedule user data using the time frequency resources identified for antenna calibration operation. The O-DU may schedule data during calibration operation using time frequency resources not identified for calibration operation. When the O-DU is scheduling user data during calibration process using resources not used for calibration, it shall only schedule DL user data in DL calibration symbols and UL user data in UL calibration symbols.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

12.5.1 Overall Operation

12.5.1.1 General

During the O-RU "start-up" procedure, the NETCONF client (O-DU) retrieves the O-RU's antenna calibration capability information including antenna calibration capability related parameters defined in o-ran-antenna-calibration.yang model. These parameters describe the O-RU's time resource requirements for calibration and the O-RU's capability of performing "self-calibration". The O-RU time resource requirements are described using the parameters **number-of-calibration-symbols-per-block-dl** and **number-of-calibration-symbols-per-block-ul**. One symbol block corresponds to a set of consecutive symbols in time required for the calibration operation, and it is the basic time unit of calibration. Sets of symbol blocks are grouped into one calibration step and the O-RU shall indicate how many symbol blocks constitute one calibration step using the **number-of-calibration-blocks-per-step-dl** and **number-of-calibration-blocks-per-step-ul** parameters. The O-RU indicates these parameters separately for downlink and uplink calibration. The O-RU shall also indicate the minimum time gap required between consecutive symbol block allocations (**interval-between-calibration-block**), number of calibration steps needed (**number-of-calibration-steps)** and the minimum required time gap between consecutive calibration step allocations (**interval-between-calibration-step**). Based on these parameters, the O-DU shall be able to allocate the time resources required for antenna calibration operation meeting the necessary time resources indicated by the O-RU. If the O-RU supports mixed numerology, the highest possible numerology supported by the O-RU shall be used as the common reference per component carrier according to the CUS plane definition for slot indexing with mixed numerologies.

12.5.1.2 Initiation

Either the O-RU or O-DU may initiate calibration operation. The trigger condition for the O-DU and/or O-RU to initiate calibration is out of scope of this specification. The NETCONF client is assumed to have subscribed to the notifications defined in the o-ran-antenna-calibration YANG model. When an O-RU determines that it needs to perform antenna calibration operation, it notifies the notification subscriber using the notification **antenna-calibration-required**, including a list of frequency ranges corresponding to the minimum frequency resources required for calibration. Upon reception of the notification, the O-DU can allocate time frequency resources for calibration and can send the **start-antenna-calibration** RPC request, including the time resource allocation information for the antenna calibration. This operation is referred as 'O-RU initiated antenna calibration' operation. The O-DU may also autonomously initiate calibration operation, using the same **start-antenna-calibration** RPC request, i.e., without receiving the **antenna-calibration-required** notification message from O-RU. This operation is referred as 'O-DU initiated antenna calibration' operation. If the O-RU has indicated the need for the calibration through sending the **antenna-calibration-required** notification", the O-DU shall consider that the use of frequency resources indicated using frequency range list within the notification as being affected during the calibration operation and if no frequency list is available, consider the full bandwidth of all configured carriers reserved is affected during calibration.

After receiving "**start-antenna-calibration** RPC request" (antenna calibration start command), the O-RU shall send an RPC reply (antenna calibration start response) including ACCEPTED status to the NETCONF client, if the O-RU is able to start the calibration operation according to the time resources allocation information in the RPC request. Otherwise the O-RU shall include a REJECTED status in the RPC reply, with a suitable error reason such as "resource mask mismatch with O-RU antenna calibration capability", "overlapped DL and UL masks", "insufficient memory", "O-RU internal reason" (if no other error reason matches the error condition) etc. If the O-RU does not receive a **start-antenna-calibration** RPC request within 60 seconds after triggering the sending of the first **antenna-calibration-required** notification, the O-RU shall raise a major alarm "Triggering failure of antenna calibration" (see Annex A for fault details). After the alarm is raised, the O-RU may resend the **antenna-calibration-required** notification multiple times. The O-RU shall not re-send the **antenna-calibration-required** notification in periods shorter than 60 seconds.

12.5.1.3 Self-Calibration Operation

When the alarm "triggering failure of antenna calibration" alarm remains uncancelled, if self-calibration is supported and permitted, i.e., **self-calibration-support** is true and **self-calibration-allowed** is true, the O-RU may perform a self-calibration procedure. The O-RU shall wait a minimum 60 seconds after raising a major alarm and receiving **no start-antenna-calibration** RPC request from the NETCONF client before initiating its self-calibrate procedure. When self-calibration is not supported or not permitted, i.e., **self-calibration-support** is false or **self-calibration-allowed** is false, the O-RU may upgrade the severity of the alarm to critical according to the sub-section 8.3.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Normal operation of self-calibration, i.e., when the O-RU self-calibration-support set to true, requires no coordination of time-frequency resources by the O-DU, i.e., the O-DU is permitted to continue to schedule user data during calibration process using the resources identified in antenna-notification-required notification without impacting the operation of the calibration procedure.

An O-RU may also support the optional O-RU-COORDINATED-ANT-CAL feature enabling it to set the coordinated-calibration-support leaf to true. When set to true, this indicates that the O-RU is able to determine a priori the time-frequency resources required for self-calibration and indicate those to the O-DU in the antenna-calibration-coordination notification instead of antenna-notification-required notification. When the coordinated-calibration is supported and permitted, i.e., coordinated-calibration-support is true and coordinated-calibration-allowed is true, the O-RU may perform a coordinated self-calibration procedure. With such a procedure, the O-RU shall indicate that time-frequency resources will be sent to a subscribed O-DU at least 60 seconds before the operation of the coordinated antenna calibration procedure.

An O-DU receiving an antenna-calibration-coordination notification can beneficially use the indicated time-frequency resources to adapt its operation during the antenna calibration operation, e.g., consider the time-frequency resources as reserved for calibration. If no UL and/or DL frequency-chunk lists are provided in the notification, the O-DU may consider the full bandwidth of all configured UL and/or DL carriers reserved for calibration operation. If such U-Plane resources are scheduled by the O-DU, the operation of the O-RU may be degraded, including performance of the calibration procedure and handling of DL and UL U-plane traffic and any associated performance counters.

12.5.1.4 Calibration Completion

The O-RU shall indicate completion of all types of calibration procedures (i.e., rpc triggered, self-calibration and co-ordinated self-calibration) using the **antenna-calibration-result** notification (Calibration results) to the notification subscriber. If a self-calibration or co-ordinated self-calibration procedure completes but with **status** set to **FAILURE**, the O-RU may upgrade the severity of the alarm to critical.

In some situations, SFN wrap around may happen causing O-DU and O-RU to interpret the ‘start-SFN’ parameter to point to different GPS seconds elapsed since GPS epoch. To avoid this situation, the O-DU to may decide not to schedule any user-plane data on the calibration time-frequency resources in all SFN cycles until the O-DU receives an **antenna-calibration-result** notification message from the O-RU. Once the calibration is complete, the O-DU schedules user data and sends C/U-Plane message as in normal operation state.

12.5.1.5 Antenna Calibration Procedure

The following figure shows the overall operation for antenna calibration.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

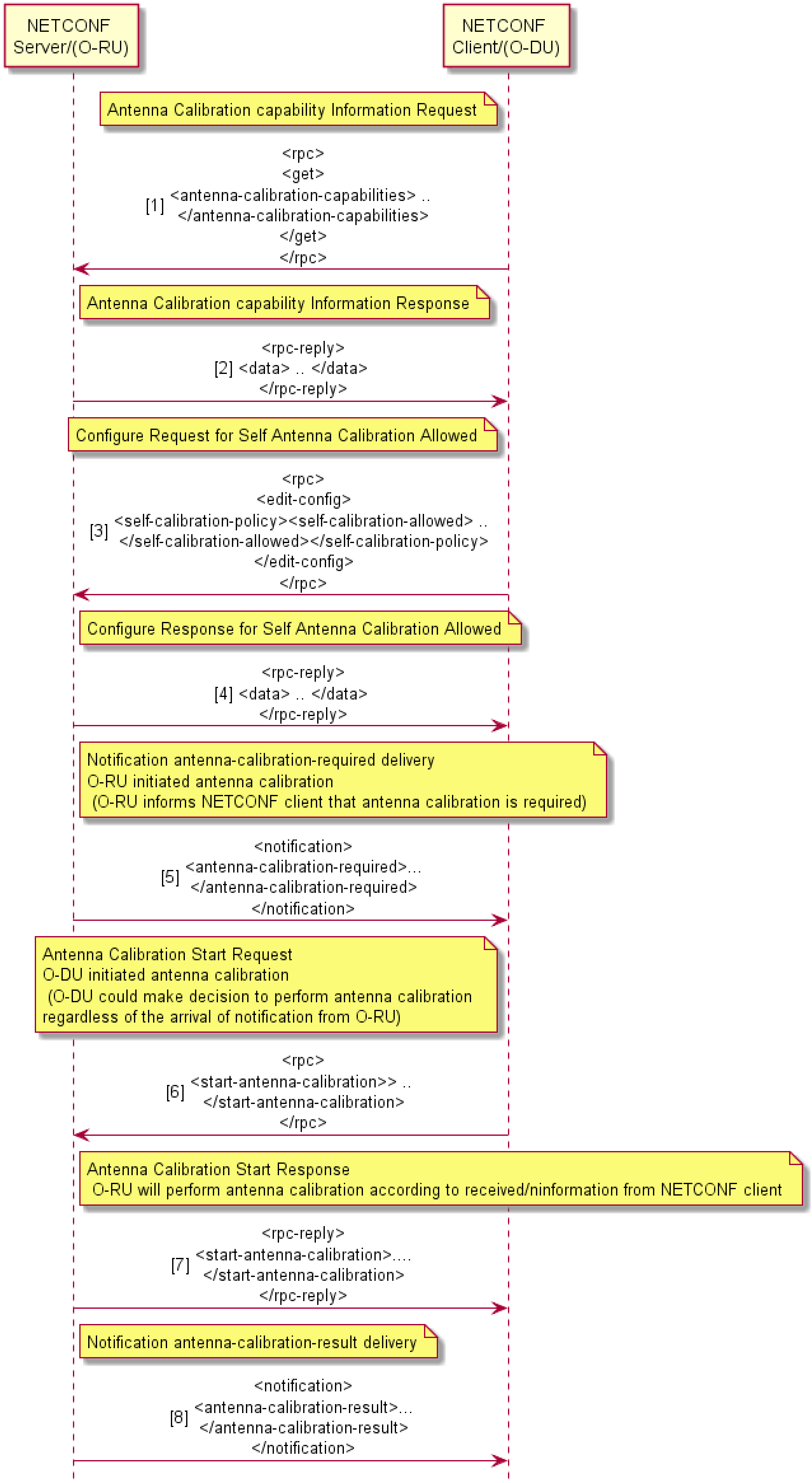


Figure 59: Overall of antenna calibration operation

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

12.5.2 O-RU Antenna Calibration Capability Parameter Configuration

The antenna calibration framework is a generic framework designed to support various vendor specific implementations of O-RU calibration. Therefore, the framework does not describe the details of how the O-RU calibrates its antenna, rather it defines a generic framework with necessary message flows and parameters for ensuring that the time and frequency resources required for calibration are coordinated between the O-DU and O-RU. The following parameters describe the O-RU's time resource needed for calibration.

- **self-calibration-support**: boolean value indicates whether O-RU is capable of supporting self-calibration.

- **number-of-calibration-symbols-per-block-dl**: indicates how many consecutive symbols are required for DL antenna calibration operation, i.e. the size of DL Symbol-block.

- **number-of-calibration-symbols-per-block-ul**: indicates how many consecutive symbols are required for UL antenna calibration operation, i.e. the size of UL Symbol-block.

- **interval-between-calibration-blocks**: if a time interval is required between consecutive antenna calibration operation, this indicates the required time value as unit of symbols. A common value is used here for the intervals between DL-DL blocks, UL-UL blocks, DL-UL blocks and UL-DL blocks, which is the largest minimum interval required between any two adjacent calibration blocks. It shall be any value that O-RU implementation requires within this parameter range.

- **number-of-calibration-blocks-per-step-dl**: indicates how many blocks are required for one step of DL antenna calibration operation.

- **number-of-calibration-blocks-per-step-ul**: indicates how many blocks are required for one step of UL antenna calibration operation.

- **interval-between-calibration-steps**: if a time interval is required between consecutive steps of antenna calibration operation, define indicates the required time value as unit of radio frames. It can be any value that the O-RU implementation requires within the defined parameter range.

- **number-of-calibration-steps**: shows how many steps is required for whole DL/UL antenna calibration operation.

The following figure shows the relationship between the various antenna calibration capabilities parameters described above.

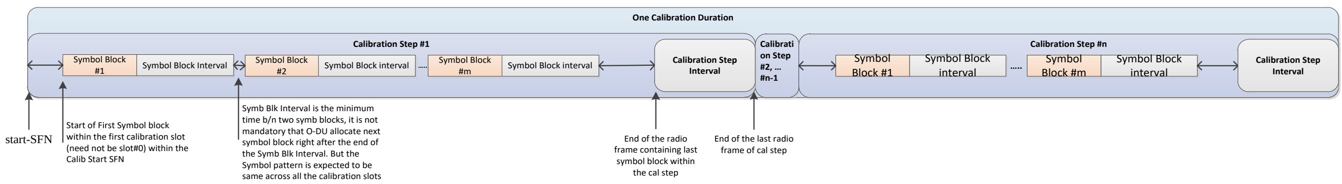


Figure 60: Relationship among Antenna Calibration Capability parameters

12.5.3 antenna-calibration-required Notification Parameters

If the O-RU initiates the calibration operation, the O-RU notifies the notification subscriber (O-DU) using the **antenna-calibration-required** notification message, including the O-RU's frequency resource requirements. The O-RU is able to indicate non-contiguous frequency "chunks" necessary for calibration using the **dl-calibration-frequency-chunk** and **ul-calibration-frequency-chunk** lists. These lists use the parameters below to describe the frequency resources required for calibration:

- **start-calibration-frequency-dl**: indicates the lowest frequency value in Hz of the frequency range is required for DL antenna calibration operation.

- **end-calibration-frequency-dl**: indicates the highest frequency value in Hz of the frequency range is required for DL antenna calibration operation.

- **start-calibration-frequency-ul**: indicates the lowest frequency value in Hz of the frequency range is required for UL antenna calibration operation.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

- **end-calibration-frequency-ul**: indicates the highest frequency value in Hz of the frequency range is required for UL antenna calibration operation.

12.5.4 Start-antenna-calibration RPC Request Parameters

The NETCONF Client sends the "**start-antenna-calibration** RPC request" including the time resource allocation parameters. These parameters indicate the exact symbols, slots, and frames that can be used for calibration.

Note, because the NETCONF Client (O-DU) is responsible for allocating the time resources for calibration with the knowledge of UL and DL configuration, dynamic TDD operation is implicitly supported.

The resource allocation information about symbol, slot, and frame are indicated using bitmasks for downlink and uplink calibration separately. The start SFN of the first calibration step is sent to the O-RU to synchronize the calibration starting point at both O-DU and O-RU. When indicated, the O-RU shall use the frequency resources indicated using the frequency ranges in the "NETCONF **antenna-calibration-required** notification" message for calibration.

Following table lists the parameters configured in the O-RU using the "**start-antenna-calibration** RPC request"

Parameters	Type / Range	Descriptions
symbol-bitmask-dl	string	Bitmask indicating DL calibration symbol within a calibration slot. First character in the string indicate first symbol, next character in the string indicate second symbol and so on. Value 1 indicates that the symbol is allocated for calibration and 0 means the symbol shall not be used for calibration.
Symbol-bitmask-ul	string	Bitmask indicating UL calibration symbol within a calibration slot. First character in the string indicate first symbol, next character in the string indicate second symbol and so on. Value 1 indicates that the symbol is allocated for calibration and 0 means the symbol shall not be used for calibration.
Slot-bitmask-dl	string	Bitmask indicating DL calibration slot within a calibration frame. First character in the string indicate first slot, next character in the string indicate second slot and so on. Value 1 indicates that the slot is allocated for calibration and 0 means the slot shall not be used for calibration.
Slot-bitmask-ul	string	Bitmask indicating UL calibration slot within a calibration frame. First character in the string indicate first slot, next character in the string indicate second slot and so on. Value 1 indicates that the slot is allocated for calibration and 0 means the slot shall not be used for calibration.
Frame-bitmask-dl	string	Bitmask indicating DL calibration frame within a calibration step. First character in the string indicate first radio frame equal to the start-SFN, next character in the string indicate the next frame and so on. Value 1 indicates that the frame is allocated for calibration and 0 means the frame shall not be used for calibration.
Frame-bitmask-ul	string	Bitmask indicating UL calibration frame within a calibration step. First character in the string indicate first radio frame equal to the start-SFN, next character in the string indicate the next frame and so on. Value 1 indicates that the frame is allocated for calibration and 0 means the frame shall not be used for calibration.
Calibration-step-size	uint8	Number of frames within a calibration step
calibration-step-number	uint8	Number of calibration steps
start-SFN	unt16	SFN number of the first calibration step

Table 6: Antenna Calibration Parameters

12.5.5 Example Antenna Calibration Operation

This subsection illustrates an example of antenna calibration operation. For simplicity, the O-RU is the initiator in this example, but either O-RU or O-DU could initiate antenna calibration operation. In this example, the TDD configuration is assumed as shown in Figure 61.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

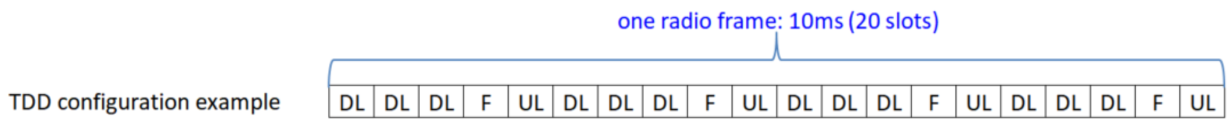


Figure 61: Example of TDD configuration

DL = Downlink slot
UL = Uplink slot
F = Flexible slot

This example illustrates calibration operation where an O-RU requires DL and UL antenna calibration operation in **o-ran-antenna-calibration.yang** with 2 calibration steps; within each step, 64 DL calibration blocks with 4 continuous DL symbols in each calibration block and 32 UL calibration blocks with 1 continuous UL symbol in each calibration block are required. Between each calibration block, a length of minimum 3 symbols interval is required, and a length of minimum 5 frames interval is required between consecutive calibration steps.

Once antenna calibration operation is required by the O-RU, an **antenna-calibration-required** notification is sent to the notification subscriber (O-DU), including the O-RU's frequency resources requirement in a list of frequency ranges in Hz, which in this example uses a single chunk of frequencies from 1.8GHz to 1.82GHz. The O-DU considers that the frequency range indicated in the **antenna-calibration-required** notification will be subsequently used during antenna calibration. The O-DU will allocate time resources for antenna calibration based on the TDD configuration together with the O-RU DL and UL antenna calibration capability, then configure the antenna calibration using the **start-antenna-calibration** RPC request. In this example, 64 DL calibration blocks in each calibration step are allocated in 4 frames, within each frame, 8 DL slots are allocated and within each DL slot, 2 calibration blocks are allocated for DL calibration. In parallel, 32 UL calibration blocks in each calibration step are allocated in 4 frames, within each frame, 4 UL slots are allocated and within each UL slot, 2 calibration blocks are allocated for UL calibration. To guarantee the interval between 2 calibration steps, the size of each calibration step is set to 10 frames. At least 3 symbols interval between each calibration block is also guaranteed in symbol bitmasks. The O-DU may allocate larger intervals than O-RU requires as shown in this example where a 9 symbols interval is allocated instead of the minimum of 3 symbols after second UL symbol block in all UL calibration slots.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

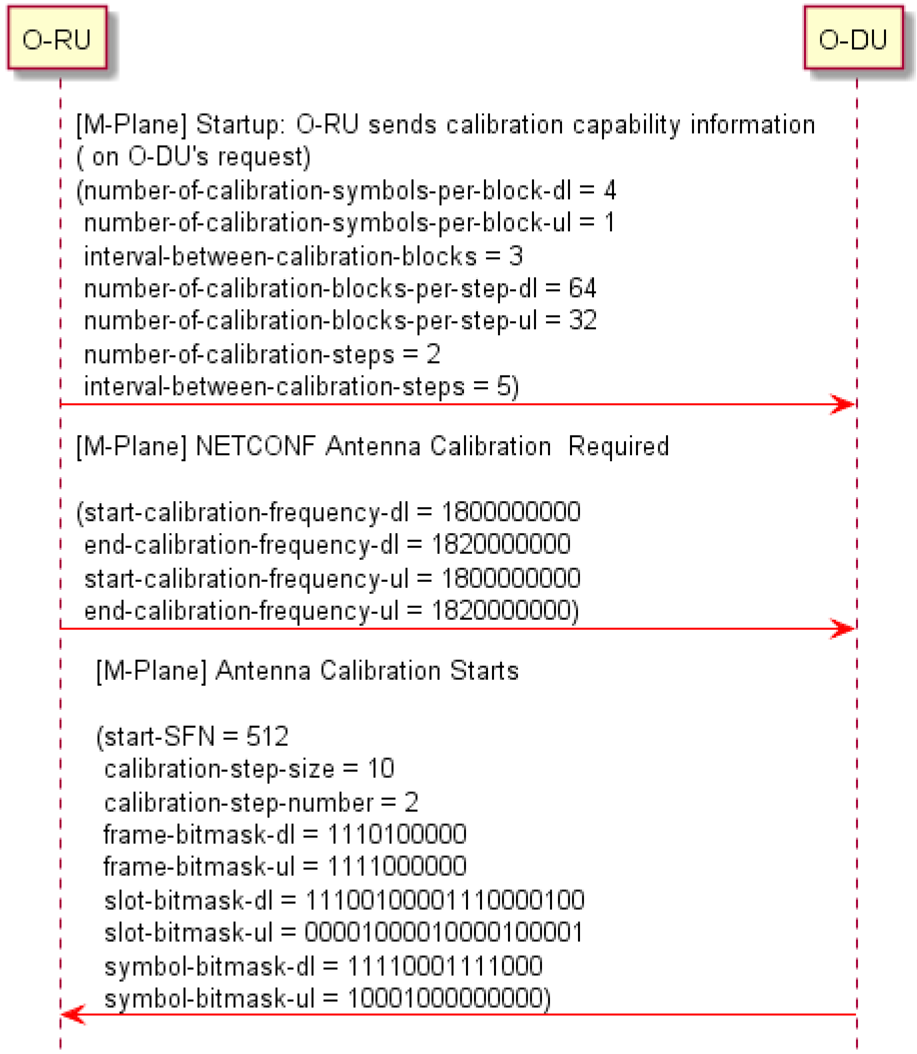


Figure 62: Example of message exchange

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

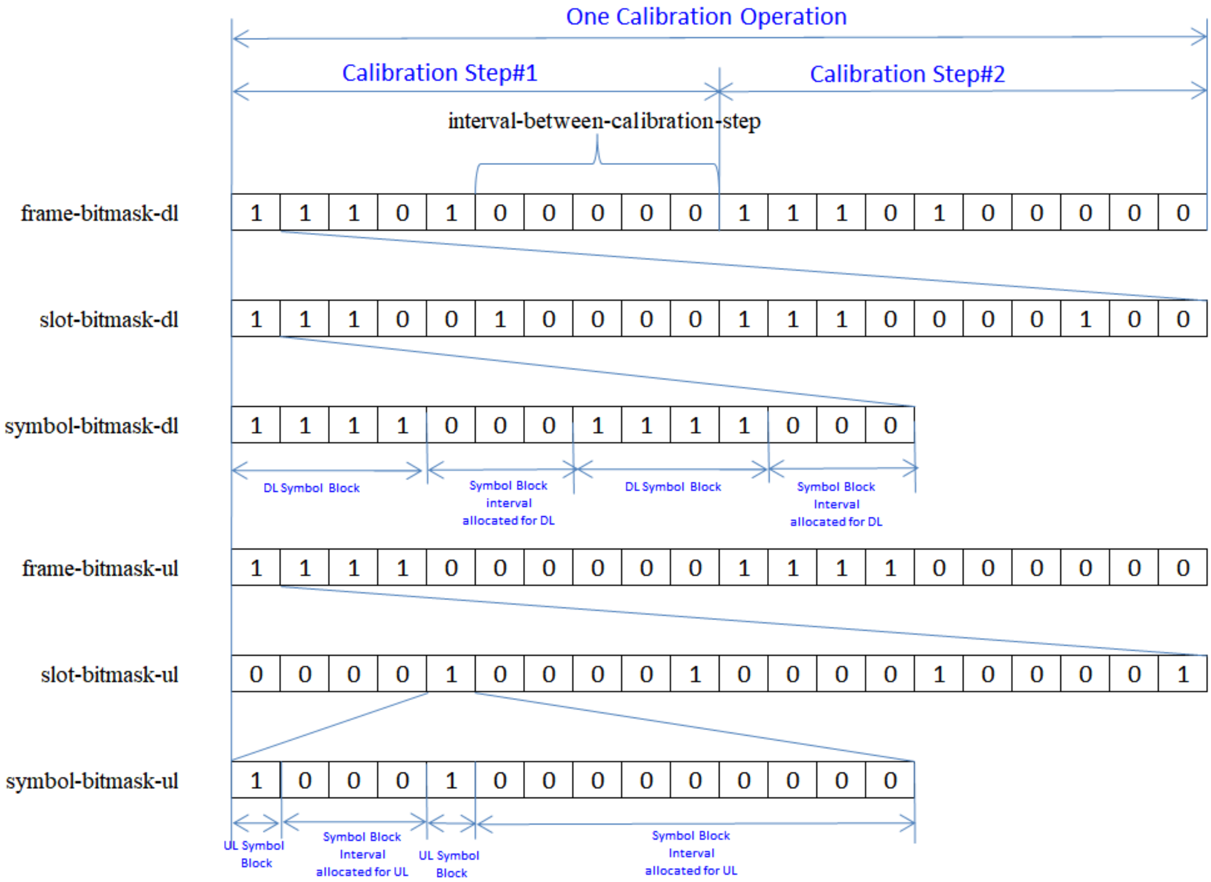


Figure 63: Time domain bitmask information from O-DU

12.6 Static configuration for PRACH and SRS

PRACH and raw SRS are periodic. Their location in time and frequency resources is constant for all periods. This makes it feasible to configure PRACH and raw SRS with M-Plane in a sense that handling PRACH and / or raw SRS processing by assigned low-level-rx-endpoints does not require real-time control through C-Plane messages.

Static configuration of PRACH and SRS with M-Plane needs to cover following aspects:

- Configuration of frequency resources assigned to PRACH / SRS
- Configuration of time resources assigned to PRACH / SRS (including PRACH / SRS periodicity)
- Configuration of compression, iFFT and SCS
- Assignment of HW resources (low-level-rx-endpoints) for processing of PRACH / SRS

Static configurations shall be provided to the O-RU as part of carrier configuration - before the configured carrier is activated. Static PRACH / SRS configuration provided for already active carrier shall be rejected by the O-RU.

Note: In case a static-low-level-rx-endpoint exposes parameter **static-config-supported** with value **NONE** – such endpoint does not offer support for static configuration of PRACH nor SRS reception.

Note: In case the configuration provided to O-RU contains records for TDD pattern(s), PRACH patterns and/or SRS patterns, the O-RU validates consistency between patterns. Configuration where there is collision between patterns detected, shall be rejected by the O-RU.

12.6.1 Static configuration for PRACH processing

The O-RU exposes its ability to support static PRACH configuration by support of the feature **PRACH-STATIC-CONFIGURATION-SUPPORTED** in o-ran-module-cap.yang module. Presence of this feature means, that at least one of static-low-level-rx-endpoints offered by the O-RU supports static configuration for PRACH. From the model perspective, static PRACH configuration is supported by static-low-level-rx-endpoints having the parameter **static-**

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

config-supported exposed as **PRACH**. Such static-low-level-rx-endpoint can be referenced by low-level-rx-endpoint designated for reception of PRACH. Specific PRACH configuration may be utilised by the low-level-rx-endpoint according to the optional parameter **static-prach-configuration**.

Note: a single low-level-rx-endpoint can only reference to single instance of static-prach-configuration. However, a single static-prach-configuration may be referenced by many low-level-rx-endpoints.

If parameters related to static PRACH configuration are set by NETCONF Client – real-time C-Plane control for PRACH opportunities shall not be provided to the O-RU, allowing for static configuration to be utilised.

12.6.1.1 Frequency domain configuration

The meaning of frequency-related parameters is illustrated using the example figure below. Note: Parameter offset-to-absolute-frequency-center belongs to low-level-rx-endpoint.

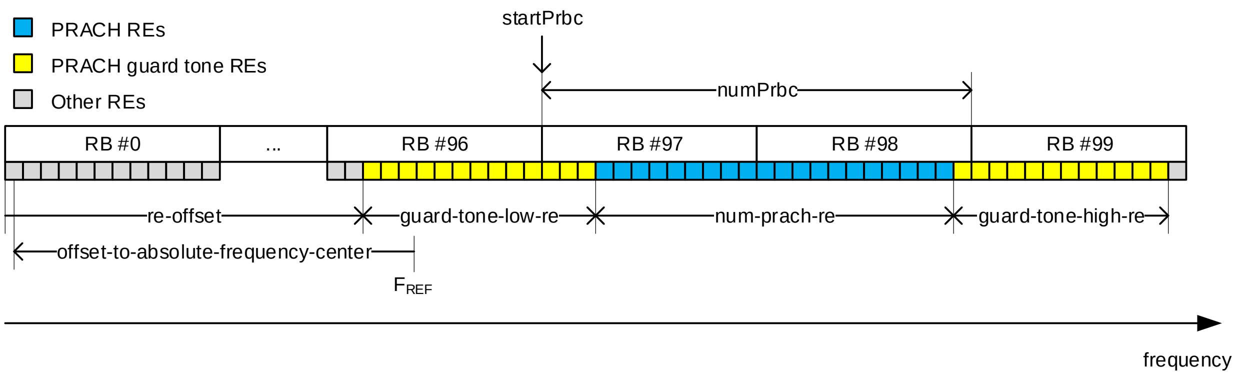


Figure 64: Relation between frequency-related parameters of the PRACH occasion

Relations between parameters allow to calculate startPrbc and numPrbc. For details of startPrbc and numPrbc please see: O-RAN Fronthaul Working Group; Control, User and Synchronization Plane Specification [2].

12.6.1.2 Time domain configuration

Meaning of parameters is illustrated using the example figure below.

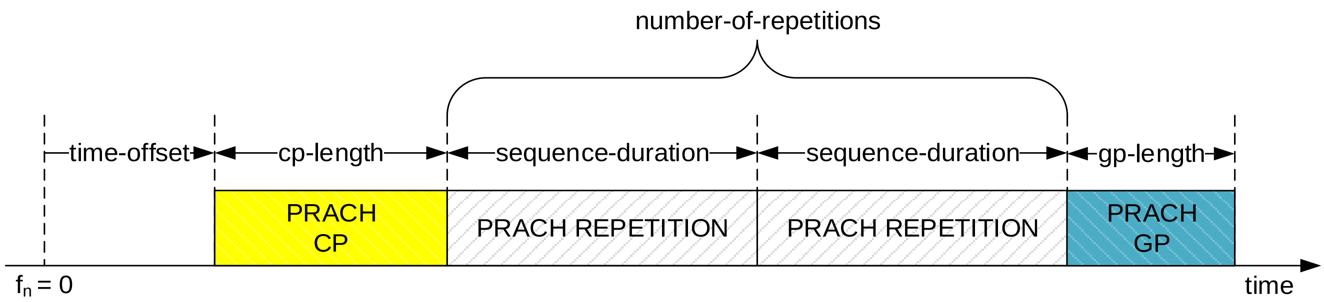


Figure 65: Timing-related parameters of single PRACH occasion

Figure 65 shows a single PRACH occasion containing 2 repetitions.

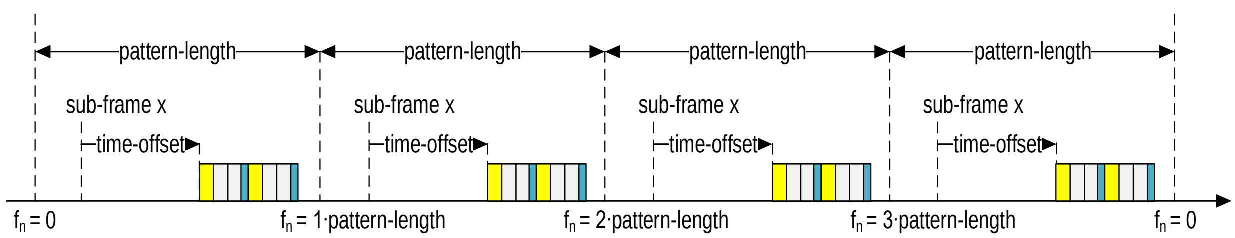


Figure 66: Timing-related parameters of one PRACH pattern

Figure 66 shows a single PRACH pattern containing two occasions of 2 repetitions (reuse of occasion shown on Figure 65 for simplified view).

The corresponding parameters for above diagram are: (“number-of-prach-occassions” = 2, “number-of-repetitions” = 2)

> Note: **time-offset** is defined with reference to parameters **frame-number** and **sub-frame-id** under **static-prach-configuration**. This parameter applies for the first occasion of a PRACH pattern. For subsequent occasions of the same PRACH pattern, the O-RU utilizes the parameters **cp-length**, **gp-length** and **beam-id** to determine the time boundaries. The parameters are taken from the list **occasion-parameters** such that, the first occasion uses the first set of elements from the list. Subsequent occasions use consecutive sets of parameters. The number of sets of parameters in this list is equal to value of the parameter **number-of-occasions**.

One **static-prach-configuration** instance allows to configure a set of PRACH patterns. For a single PRACH configuration, all corresponding PRACH patterns repeat over the period defined by the **pattern-period** parameter for such PRACH configuration. The PRACH patterns of single PRACH configuration shall not overlap in terms of time and frequency.

At most one PRACH pattern shall start in a subframe (subframes in different frames are distinguished). PRACH pattern shall not cross boundary between subframes except PRACH pattern for long PRACH format with one occasion that spans boundary between subframes.

An O-RU shall reject any configuration where the number of patterns in single static PRACH configuration exceeds the number exposed by capability parameter **max-prach-patterns** in o-ran-uplane-conf.yang module.

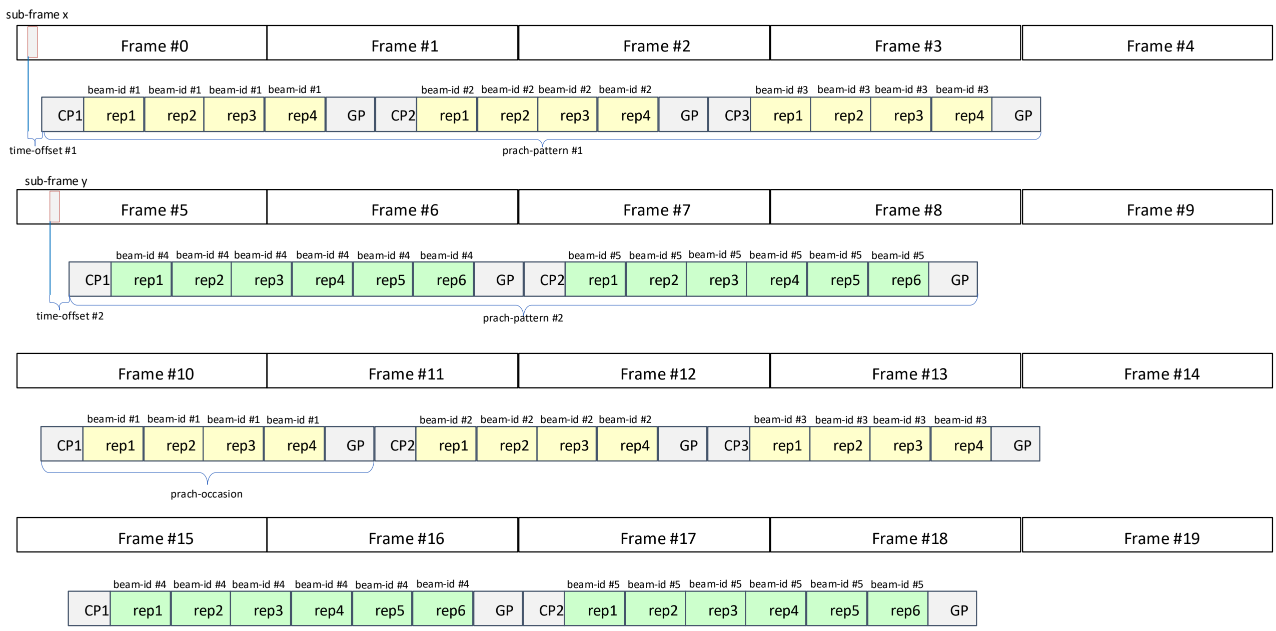


Figure 67: Example PRACH configuration formed of two PRACH patterns having different number of repetitions (Note: PRACH occasions are expanded for visual clarity).

> Note: Figure 67 shows a theoretical configuration – not necessarily standardized by 3GPP. This is to demonstrate the flexibility of the solution of static PRACH configuration offered by M-Plane configuration.

The above configuration uses 2 prach-patterns:

- Pattern #1 is having: number-of-repetitions = 4, number-of-occasions = 3
- Pattern #2 is having: number-of-repetitions = 6, number-of-occasions = 2

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

For the PRACH configuration shown on Figure X-4, the parameter **pattern-period** = 10 as this is the number of frames after which PRACH pattern repeats.

> Note: Such a static PRACH configuration can be supported by static-low-level-rx-endpoints having parameter **max-prach-pattern**s ≥ 2 as this configuration consists of 2 patterns.

12.6.1.3 Operation

Static PRACH configuration shall be set and rx-endpoints shall be linked to it before rx-array-carrier activation. On carrier activation, the O-RU starts receiving RF signals corresponding to the configured **prach-patterns** list. Specifically, the O-RU receives RF signals corresponding to the prach-pattern p when

mod(n_f, pattern-period) = frame-number$_p$ and
n_{sf} = sub-frame-id$_p$ and
t = time-offset$_p$

where

n_f is the system frame number,
mod(x, y) is remainder of division of x by y,
n_{sf} is the subframe number within system frame n_f,
t is the time since start of subframe n_{sf},
frame-number, sub-frame-id$_p$ and time-offset$_p$ are parameters of prach-pattern p,
pattern-period is a parameter of PRACH configuration.

Once the RF signal corresponding to the PRACH repetition n_r in PRACH occasion n_o is received and processed, the O-RU sends the corresponding IQ values in a U-plane message or messages with header fields set as follows:

frameId = mod(floor(n_f / pattern-period) • pattern-period + frame-number$_p$, 256)
(note: this corresponds to n_f value captured when prach-pattern p started)
subframeId = sub-frame$_p$,
slotId = zero-based PRACH occasion number within PRACH pattern,
symbolId = zero-based PRACH repetition number within PRACH occasion,
sectionId = 4095,
startPrbu = floor((re-offset$_p$ + guard-tone-low-re) / 12),
numPrbu = ceil((re-offset$_p$ + guard-tone-low-re + num-prach-re) / 12) - startPrbu.

where

n_f is the system frame number,
mod(x, y) is remainder of division of x by y,
floor(x) is largest integer smaller than or equal to x,
ceil(x) is smallest integer greater than or equal to x,
frame-number$_p$, sub-frame-number$_p$ and re-offset$_p$ are parameters of prach-pattern p.
pattern-period, guard-tone-low-re and num-prach-re are parameters of PRACH configuration.

If data section is subdivided due to application level fragmentation, resulting values of startPrbu and numPrbu shall be calculated as per general rules. If multiple PRACH repetitions are scheduled at the same time at different re-offset frequencies, the O-RU shall send corresponding data sections in one U-Plane message following message size restrictions.

12.6.2 Static configuration for raw SRS processing

The O-RU exposes its ability to support static raw SRS configuration by support of the feature **SRS-STATIC-CONFIGURATION-SUPPORTED** in o-ran-module-cap.yang module. Presence of this feature means, that at least one of static-low-level-rx-endpoints offered by the O-RU supports static configuration for raw SRS reception. From the model perspective, static SRS configuration is supported by static-low-level-rx-endpoints having the parameter **static-config-supported** exposed as **SRS**. Such static-low-level-rx-endpoint can be referenced by a low-level-rx-endpoint designated

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

for reception of SRS. Specific SRS configuration may be utilised by the low-level-rx-endpoint according to the optional parameter **static-srs-configuration**.

Note: a single low-level-rx-endpoint can only reference to single instance of static-srs-configuration. However, a single static-srs-configuration may be referenced by many low-level-rx-endpoints.

If parameters related to static SRS configuration are set by NETCONF Client – real-time C-Plane control for SRS shall not be provided to the O-RU, allowing for static configuration to be utilised.

Static SRS configuration is used to configure NDM (Non-Delay Managed) raw SRS (Sounding Reference Signal) patterns in a static manner, such that raw SRS U-Plane traffic can be processed by the O-RU without receiving C-Plane messages conveying real-time raw SRS configuration. Raw SRS may capture non-beamformed (beam-id = 0) or beamformed (beam-id != 0) signals and uses non-delay managed U-Plane messages.

One **static-srs-configuration** instance allows to configure a set of SRS patterns. For a single SRS configuration, all SRS patterns repeat over the period defined by **pattern-period** parameter for such SRS configuration. SRS patterns corresponding to a single SRS configuration shall not overlap in terms of time and frequency.

An O-RU shall reject any configuration where the number of patterns in single static SRS configuration exceeds the number exposed by capability parameter **max-srs-patterns** in o-ran-uplane-conf.yang module.

12.6.2.1 Operation

Static SRS configuration shall be set and rx-endpoints shall be linked to it before rx-array-carrier activation. On carrier activation, the O-RU starts receiving RF signals corresponding to the configured **srs-patterns** list. Specifically, the O-RU receives RF signal and sends corresponding U-plane messages as if, for each configured srs-pattern p, each rx-endpoint linked with the SRS configuration received C-plane messages with fields:

dataDirection = 0 (RX),
payloadVersion = 0,
filterIndex = 0,
frameId = mod(n_f, 256),
subframeId = sub-frame-id$_p$,
slotId = slot-id$_p$,
startSymbolId = start-symbol-id$_p$,
numberOfSections = 1,
sectionId = 4095,
rb = 0,
symInc = 0,
startPrbc = start-prbc$_p$,
numPrbc = num-prbc$_p$,
reMask = 0xFFF,
numSymbol = num-symbol$_p$,
ef=0,
beamId = beam-id$_p$,

where

n_f is the system frame number,
mod(x, y) is remainder of division of x by y,
sub-frame-id$_p$ slot-id$_p$, start-symbol-id$_p$, beam-id$_p$, start-prbc$_p$ and num-prbc$_p$ are parameters of srs-pattern p,

12.7 TDD pattern configuration

The O-RU exposes its ability to support TDD pattern configuration by support of the feature **CONFIGURABLE-TDD-PATTERN-SUPPORTED** in o-ran-module-cap.yang module. Presence of this feature means, that at least one of static-low-level-[tr]x-endpoints offered by the O-RU supports configuration for TDD pattern, so that these static-low-level-

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

[tr]x-endpoints can be used (through low-level-[tr]x-endpoints) by [tr]x-array-carriers having configurable TDD pattern assigned.

Note: Configured TDD pattern shall not be violated by C-Plane and U-Plane messages.

Note: In case configuration provided to O-RU contains records for TDD pattern(s), PRACH patterns and/or SRS patterns, O-RU validates consistency between patterns. Configuration where there is collision between patterns detected, shall be rejected by O-RU.

From the model perspective, configuration for TDD pattern is supported by static-low-level-[tr]x-endpoints having parameter **configurable-tdd-pattern** exposed as **TRUE**. Such static-low-level-[tr]x-endpoint can be respectively referenced by low-level-[tr]x-endpoint designated to serve for [tr]x-array-carrier having preconfigured TDD pattern assigned. Specific configuration of the TDD pattern may be utilised by [tr]x-array-carrier according to the optional parameter **configurable-tdd-pattern**.

Absence of leaf **configurable-tdd-pattern** at [tr]x-array-carrier means, that such [tr]x-array-carrier has no configurable-tdd-pattern assigned.

A configurable TDD pattern can be assigned to a [tr]x-array-carrier under the condition, that all static-low-level-[tr]x-endpoints serving such an [tr]x-array-carrier expose value of capability **configurable-tdd-pattern-supported** as **TRUE**.

A single [tr]x-array-carrier can only reference to single instance of configurable-tdd-pattern. Whereas, a single **configurable-tdd-pattern** shall be referenced by all cooperating [tr]x-array-carriers serving for a specific [tr]x-array. Linkage between tx-array-carriers and rx-array-carriers configured to use the same configurable-tdd-pattern must be assured by the entity responsible for configuration provisioning to O-RU. For example, ensuring that all cooperating [tr]x-array-carriers use static-low-level-[tr]x-endpoints (through low-level-[tr]x-endpoints) having the same value of **tdd-group**. The practical implication of this is that static-low-level-[tr]x-endpoints exposing the same value of parameter **tdd-group** must be used by low-level-[tr]x-endpoints serving for [tr]x-array-carriers having the same TDD switching points and the same directions to the air interface granted by TDD patterns they are configured to use.

Note: M-Plane model allows an O-RU to be configured with more than one TDD patterns. This is capability can be used by O-RUs having more than one [tr]x-array.

A single TDD pattern configuration consists of list of records. Each single record contains details for frame-offset and direction of signal that must be applied at the moment a specific frame-offset occurs at air interface. Supported directions are UL (uplink), DL (downlink) and GP (neither uplink nor downlink).

Note: Assignment of **configurable-tdd-pattern** to a [tr]x-array-carrier is only possible in case all following conditionals are met:

- O-RU supports feature **CONFIGURABLE-TDD-PATTERN-SUPPORTED**.
- all static-low-level-[tr]x-endpoint configured to serve for a specific [tr]x-array-carrier have capability **configurable-tdd-pattern** set to **TRUE**

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Chapter 13 Licensed-Assisted Access

13.1 Introduction:

Licensed-assisted access (LAA) leverages the carrier-aggregation (CA) functionality. With LAA, CA is performed between licensed and unlicensed component carriers (CCs). This enables the LAA system to opportunistically benefit from using unlicensed spectrum (e.g., UNII bands in the 5 GHz spectrum) to enhance the aggregated capacity of the O-RU with the objective of enhancing the downlink throughput.

Several modifications in the RAN are needed to enable LAA in the O-DU and O-RU such as listen-before-talk (LBT), discontinuous transmission, carrier-selection, discovery reference signal (DRS) transmission, etc. This section is focused on the LAA-related messages and procedures needed in the M-plane. The C-plane related messages are defined in the CUS-plane spec [2].

This version of the M-plane spec supports only LAA based on Rel. 13 of the 3GPP specs, where transmission on the unlicensed spectrum can be done only in the downlink direction. The support of eLAA Rel. 14 (i.e., enabling UL transmission on the unlicensed spectrum) may be included in a later version of the M-plane spec.

The modifications at M-plane to Support LAA can be summarized as follows:

i) LAA-initiation process: O-DU learns about O-RU capabilities and configures it.

- O-RU LAA Support: The O-RU indicates it supports LAA by including support for the "urn:o-ran:laa:x.y" and "urn:o-ran:laa-operations:x.y" namespaces in its ietf-yang-library model (RFC 7895) [20].
- O-RU LAA Capability Information: When the LAA feature is enabled, leafs corresponding to LAA-related O-RU capabilities, such as the number of supported LAA SCarriers, maximum LAA buffer size, etc, are conveyed via the M-plane to the O-DU as part of the o-ran-module-cap.yang module.
- O-RU LAA Configuration: The NETCONF client configures the unlicensed LAA component carrier with the LAA-related parameters such as the energy-detection threshold, DRS measurement timing configuration (DMTC) period, etc. as part of the o-ran-uplane-config.yang module. The configuration of the number of LAA SCarriers, multi-carrier type, etc. is performed using the o-ran-laa.yang Module.
- For explanation of the LAA-initiation process, please refer to Figure 4 in Chapter 3, where the O-RU LAA capability info is conveyed within the “Retrieval of O-RU information” step, while the O-RU LAA configurations are conveyed in “Configuring the O-RU operational parameters” step in Figure 4.

ii) Carrier-selection: Selecting the best channel in the unlicensed band, both initially and dynamically over time.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

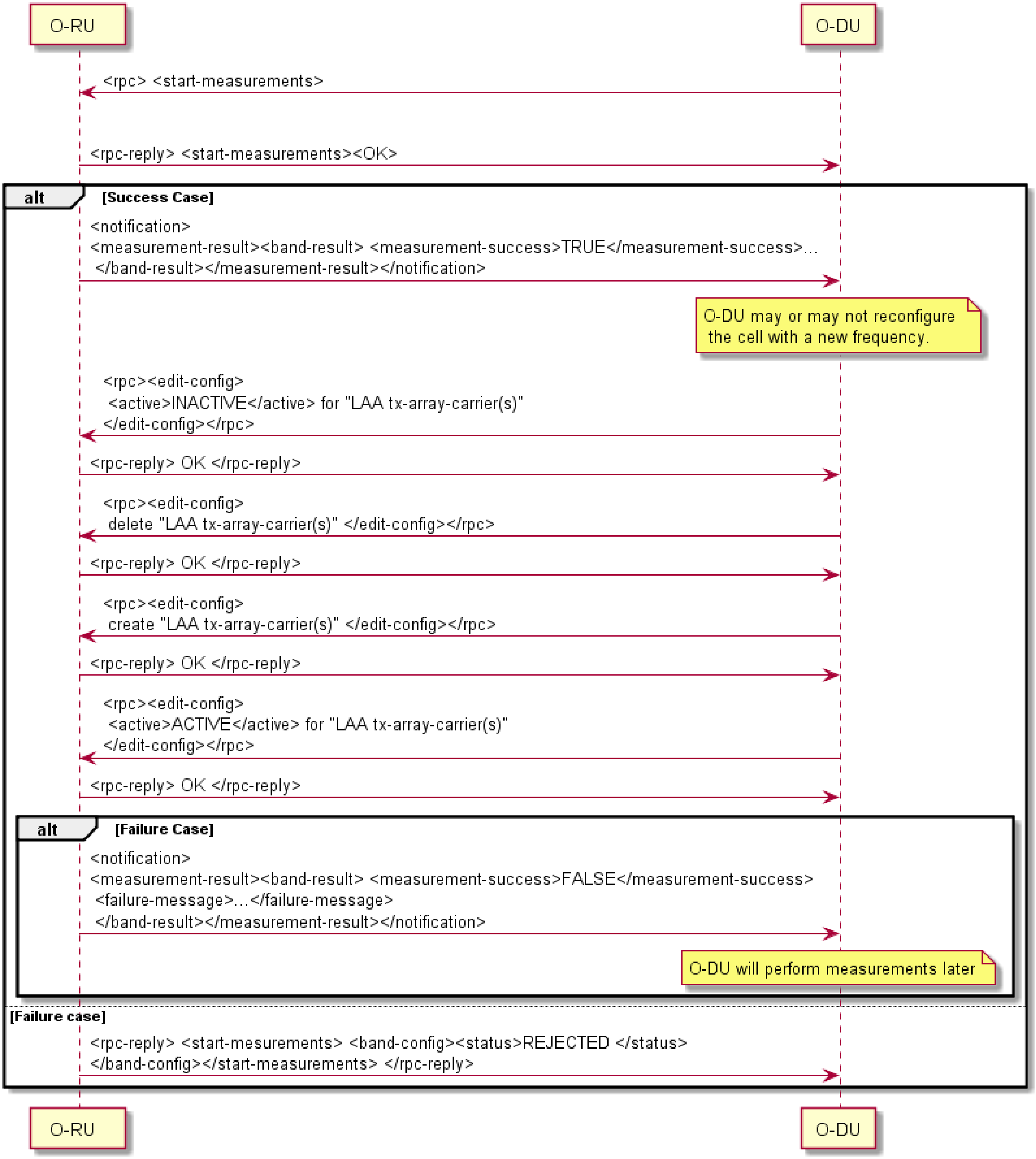


Figure 68: Carrier-Selection Call-flow

13.2 LAA-initiation Process

13.2.1 LAA Module Capabilities

During LAA-initiation, the O-RU reports its LAA capabilities to the NETCONF client. These capabilities are sent at the start up as part of the o-ran-module-cap.yang module. The attributes included are:

(1) **sub-band-frequency-ranges**: The unlicensed sub-bands (e.g., 46A, 46B, etc.) that are supported at the O-RU and their frequency ranges

(2) **number-of-laa-scarriers** (uint8): Number of LAA SCarriers that the O-RU can support.

(3) **maximum-laa-buffer-size** (uint16): Maximum O-RU buffer size in Kilobytes (KB) per CC. This parameter is needed at the O-DU to know how much data can be sent in advance and stored at the O-RU to address the LBT uncertainty.

(4) **maximum-processing-time** (uint16): Maximum O-RU Processing time in microseconds at the O-RU to handle the received/transmitted packets from/to the O-DU. This parameter is needed at the O-DU to determine the time where it needs to send the data to the O-RU.

(5) **self-configure** (Boolean): Capability to manage the contention window at the O-RU. Based on the CUS-spec, there are two modes of operation for LAA, 1) when the contention window is managed by the O-DU, and 2) when the contention window is managed by the O-RU. This field is set to True if the O-RU can manage the contention window locally.

13.2.2 LAA O-RU Parameter Configuration

The second stage of the LAA-initiation process is the configuration message (using RPC edit-config). In this message, the O-DU configures the O-RU with the required parameters in the downlink direction. LAA parameters can be configured by Netconf Client after capability exchange is finished. It can also be sent as needed, to reconfigure the O-RU with new parameters (e.g., **ed-threshold-pdsch**, etc.). The attributes of this message (o-ran-laa.yang Module) include:

(1) **number-of-laa-scarriers** (uint8): Number of LAA SCarriers to be used at the O-RU. This number should be less than or equal the number reported by the O-RU in its module capabilities.

(2) **multi-carrier-type** (Enumeration): This value indicates the list of multi carrier types (A1, A2, B1, B2) which as subsection 15.1.5 in [32].

(3) **multi-carrier-tx** (Boolean): This value indicates whether self-deferral is activated or not. “True” indicates transmission on channel access win (i.e., no self-deferral). “False” indicates mutual transmission on multiple carriers.

(4) **multi-carrier-freeze** (Boolean): This value indicates if the absence of other technology in the unlicensed band can be guaranteed. This attribute can only be used when the multi-carrier-type is A1. “False” indicates that absence of other technology is not guaranteed.

(5) **laa-ending-dwpts-supported** (Boolean): This value indicates whether LAA ending in Downlink Pilot Time Slot (DwPTS) is supported.

(6) **laa-starting-in-second-slot-supported** (Boolean): This value indicates LAA starting in second slot is supported.

LAA carrier configurations (o-ran-uplane-conf.yang Module) include:

(1) **ed-threshold-pdsch** (int8): This value indicates the energy detection (ED) threshold for LBT for PDSCH and for measurements in dBm.

(2) **ed-threshold-drs** (int8): This value indicates the ED threshold for LBT for DRS in dBm.

(4) **tx-antenna-ports** (uint8): This value indicates the Tx antenna ports for DRS.

(5) **transmission-power-for-drs** (int8): This value indicates the offset of CRS power to reference signal power (dB).

(6) **dmtc-period** (enumeration): This value indicates DMTC period in milliseconds.

(7) **dmtc-offset** (uint8): This value indicates DMTC offset in Subframes.

(8) **lbt-timer** (uint16): This value indicates LBT Timer in milliseconds.

If Self Configure capability is set to “true”, the following parameters are also needed to be configured. For every traffic priority class, the O-DU needs to configure maximum CW usage counter. This value indicates the maximum value of counter which shows how many max congestion window value is used for back off number of each priority class traffic. This value is defined at section 15.1.3 of [32] as K. Based on the 3GPP specification, this value is selected by O-RU from the set of values {1, 2, …, 8}

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

13.3 Carrier-Selection

13.3.1 LAA Measurements

The function of the message “**rpc start-measurements**” is to order the O-RU to start measurements. This message can be used for carrier selection initially or dynamically over time. O-RU sends RPC response where status==ACCEPTED (positive case) or REJECTED (negative case). O-RU performs measurement and delivers result with respect to max response time. If result is not ready on time - O-RU sends notification with "measurement-success" == FALSE and with appropriate failure reason. For every configured band, the O-RU informs the NETCONF client whether the measurement was successful or not. For bands with successful measurements, the O-RU reports the occupancy ratio and average RSSI for each channel. For bands with failure measurements, the O-RU includes the reason (e.g., TIMEOUT when the O-RU is not able to finish the measurement for this specific band).

The occupancy ratio of a given channel is defined as the percentage of the busy duration (i.e., measured signal power is larger than the energy-detection threshold) to the total measurement duration of this specific channel. The energy-detection threshold is specified in the o-ran-uplan-conf.yang module using the **ed-threshold-pdsch** leaf. Note that this threshold is the same as the energy-detection threshold used for LBT for PDSCH transmission. The total measurement duration per channel is specified in o-ran-laa-operation module using the **duration-per-channel** leaf. The range of the occupancy ratio is from 0% (no signal is detected over the total measurement duration per channel) to 100% (i.e., channel was always occupied during measurement).

The average RSSI of the measured channel is the measured power of this specific channel averaged over the total measurement duration per channel. This parameter is reported to the NETCONF client in dBm and takes a value from the range 0 dBm to -128 dBm.

13.3.2 LAA Carrier Frequency Configuration

After receiving the measurements, the NETCONF client configures the O-RU with the new channel(s), if needed. For every component carrier (CC) that needs to be configured with a new center frequency, the O-DU will need to first deactivate the TX carrier, delete it, then create a new TX carrier (with the new center carrier frequency as well as any other new configurations), and then activate the TX carrier again to start OTA operation. The procedure for deactivating/deleting/creating/activating the carrier is explained in Section 12.3: Carrier Configuration, in the M-plane specification and elaborated in Figure 50. Note that the creation of LAA carriers is identical to the creation of regular carriers but in the unlicensed bands. O-RU responds to the configuration request with success or failure.

Note also that carrier-selection algorithm at the O-DU (i.e., selecting the “best” channel based on the reported measurements from the O-RU) is an implementation issue and is out of the scope of this document.

Chapter 14 Shared Cell

This chapter specifies the support of the “Shared Cell” O-RU use case. The features of C/U-Plane aspects are described in chapter 11 of [2]. The M-Plane aspects necessary to support Shared Cell are described in this chapter.

14.1 Architecture

The NETCONF client (O-RU Controller) establishes M-Plane connection individually to each O-RU, where the O-RUs are operating in either cascade mode or FHM mode **Figure 69**. In this figure, solid lines indicate C/U-plane interface and dotted lines indicate M-plane interface. Therefore, from the M-Plane point of view, the same architecture model can be applied as specified in Chapter 2.1.2. New functionality which is required to be added, together with existing functionality which is required to be enhanced are specified in sub-sections 14.2 to 14.5. There are no changes to the functionality described in the following chapters and their associated YANG models:

- Chapter 5 : Software Management
- Chapter 8 : Fault Management
- Chapter 9 : File Management

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

A NETCONF client with suitable privileges is able to trigger a reset procedure for each O-RU. It is strongly recommended that when triggering the reset procedures for multiple O-RUs, a NETCONF client should order the procedures such that a reset of an individual O-RU does not affect the operation of other O-RUs operating in either cascade or FHM mode. This can be achieved by correct ordering of the triggering of the reset procedure between the different O-RUs. For example, when triggering a reset involving multiple O-RUs operating in cascade mode, the ordering of the reset trigger sent by the NETCONF client should be done beginning with the last (most-southern) O-RU to the first (most northern) O-RU in chain. In configuration where FHM is used - when FHM reset is needed, O-RUs connected through this FHM should be reset first, then FHM reset can be performed. It is strongly recommended to disable carriers affected by such a reset procedure prior to the triggering of the reset to minimize the impact on C/U-plane traffic as much as possible. In case an O-RU connected through FHM requires to be reset - the reset will not impact to other O-RUs.

Note: there is no difference between Hierarchical M-plane architecture and Hybrid M-plane architecture from the point of reset ordering in either cascade mode (chain topology), or FHM mode (star topology).

Using this approach, a NETCONF client can performs software management on multiple O-RUs. Since this procedure may require the reset of the O-RU to update the appropriate software file(s) for the O-RUs, the NETCONF client is recommended to order the software management procedures such that the reset procedure issued against one O-RU shall not impact the other O-RUs or FHM.

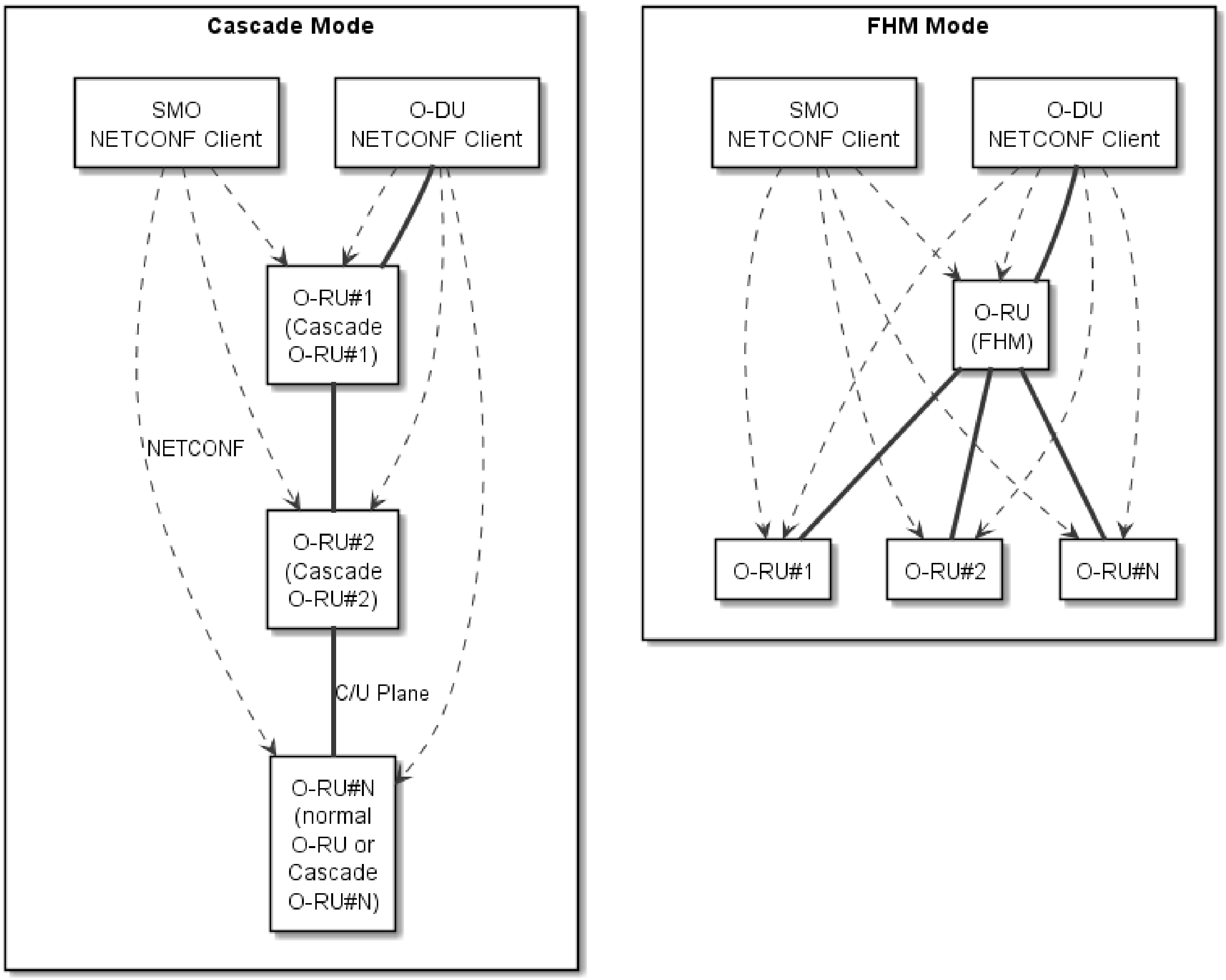


Figure 69: M-Plane Connection

14.2 Start-up and Installation

This sub-section provides the consideration regarding the shared cell specific additional mechanism for the overall start-up mechanism for "O-RUs with Copy and Combine function" and for "O-RUs without Copy and Combine function".

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Each O-RU establishes M-Plane connection individually to the NETCONF client in the O-RU Controller. The procedures through Transport Layer initialization (DHCP process and VLAN scanning) and supervision of NETCONF connection are the same as the Figure 4 in Chapter 3 for both "O-RUs with Copy and Combine function" and "O-RUs without Copy and Combine function".

For the transport layer initialization, the following assumptions are made:

- The order of each O-RU's M-plane establishment is not restricted because of the network transparency at O-RU (FHM and Cascade).
- For simplification, the network should be configured using a common management plane vlan-id or untagged interface for all O-RUs within one shared cell network managed by same NETCONF client. As a result, the same vlan-id is learned by VLAN scanning by all O-RUs within the shared cell network.

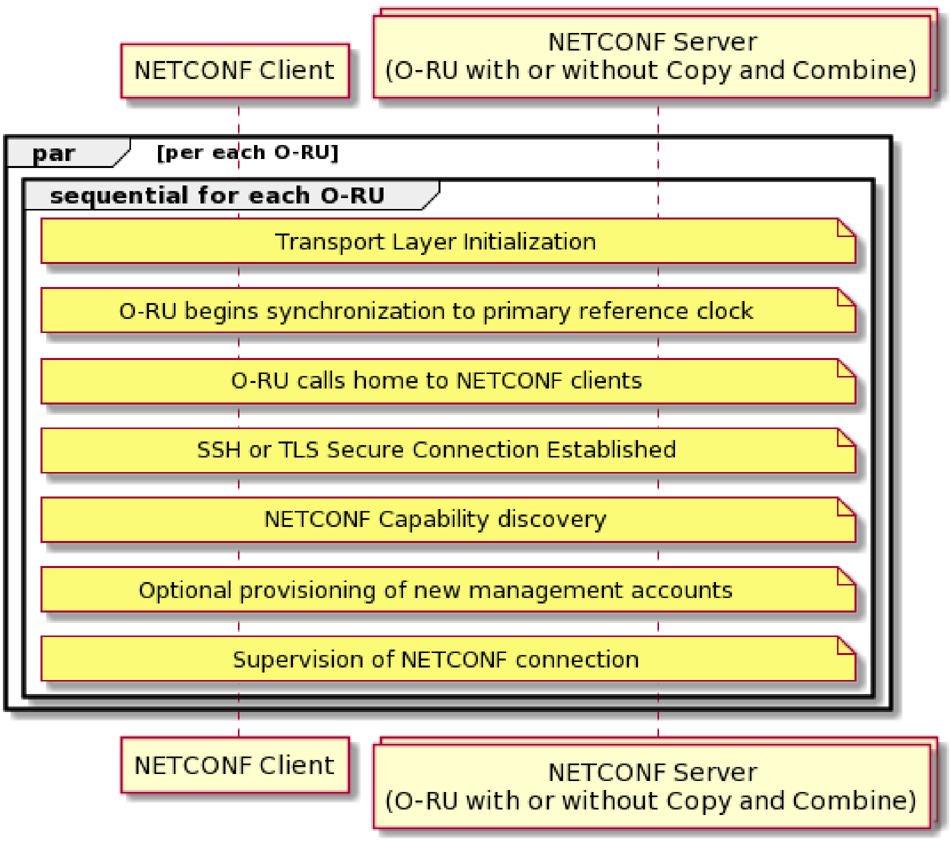


Figure 70: NETCONF establishment of the start-up

In the step "retrieval of O-RU information", the NETCONF client retrieves the O-RUs' capability from the NETCONF servers by using individual M-plane connection in parallel. The Copy and Combine related capability is defined in sub-section 14.5.1.

After the retrieval of O-RU information, the NETCONF client performs the topology discovery procedure in order to discover the topology of NETCONF servers within one shared cell network. (See sub-section 14.5.2)

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

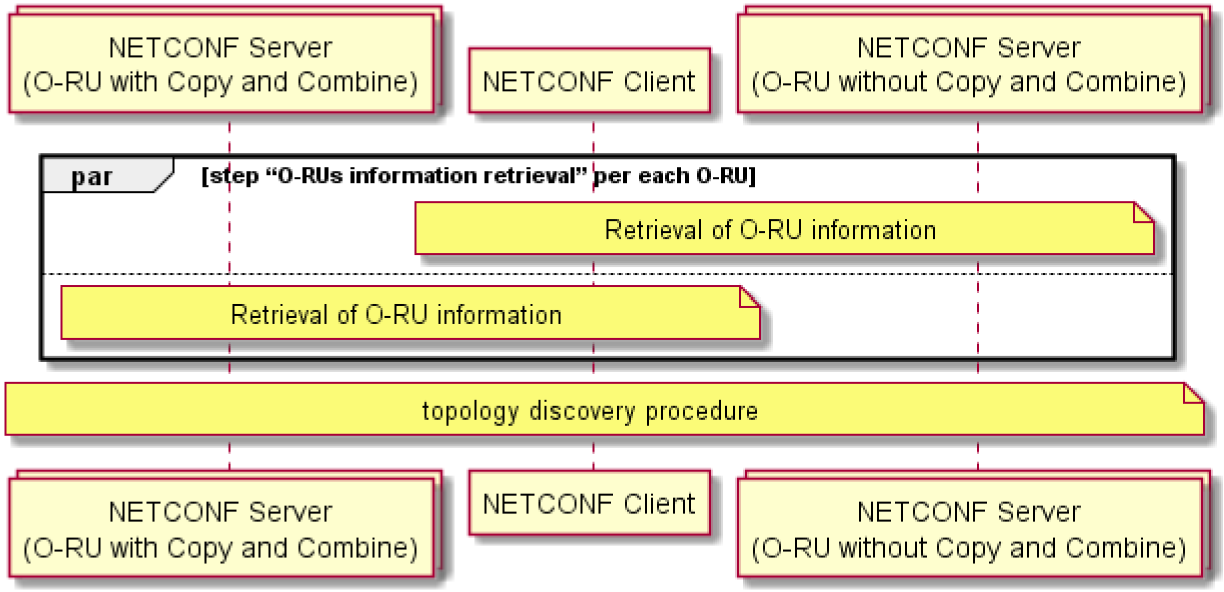


Figure 71: RU information retrieval and topology detection at the start-up

The NETCONF client performs software management per O-RU as described in Chapter 5.

As described in section 14.1, as the reset procedure is required during the software management procedure, it is recommended that the NETCONF client resets the O-RU while taking account of the topology of O-RUs and whether a reset of one O-RU will affect other O-RUs in the chain or star topology.

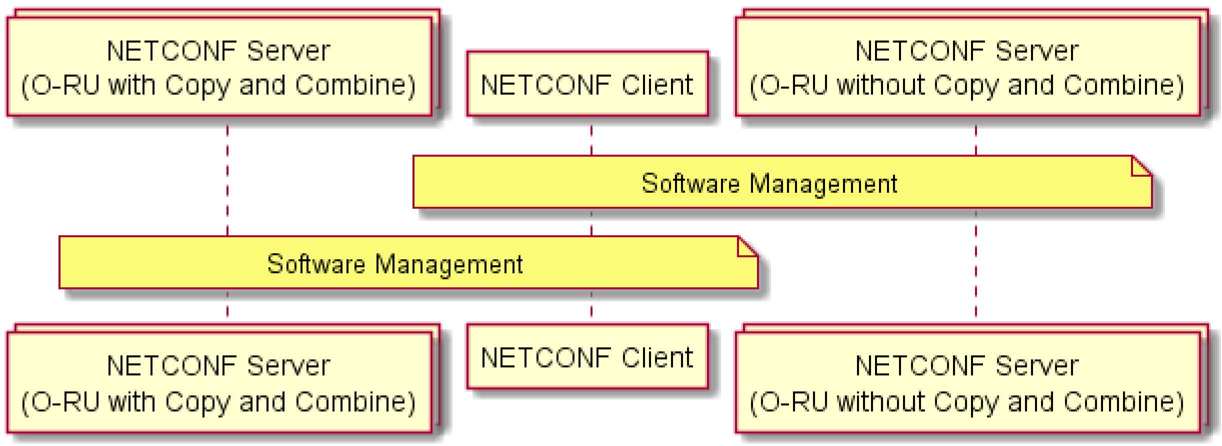


Figure 72: Software management per O-RU

After the software management steps are completed for all NETCONF servers, the NETCONF client performs shared cell configuration. (See sub-section 14.5.3)

The NETCONF client performs transport configuration, connectivity check configuration, C/U-plane transport connectivity check procedure, Retrieval of the O-RU Delay Profile and U-plane configuration procedures for all O-RUs.

In this version of the specification, the only processing element definition used for supporting shared cell is the Ethernet-type-flow which is a combination of VLAN identity and MAC address. The vlan-id(s) used for C/U-plane transport-flows is/are common to all O-RUs operating within one shared cell network.

The Ethernet bridging functionality in an O-RU with Copy and Combine function is able to bridge the Ethernet Loopback messages between the O-DU and other O-RUs configured as part of the shared cell operation. For more details, see the sub-section 14.5.2.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

The u-plane configuration in o-ran-uplane-conf module shall have identical configuration except config-false instances' names and low-level-tx(rx)-endpoints' names for all O-RUs operating within the one shared cell network. The value of gain in tx-array-carriers can be independently configured per O-RU (, i.e., a common value is not mandatory).

The u-plane configuration in o-ran-uplane-conf module is no longer required for O-RU (FHM). Instead, s**hared-cell-copy-uplane-config** and **shared-cell-combine-uplane-config** in o-ran-shared-cell.yang module are used. (See sub-section 14.5.4)

> Note: in this version of the specification, only eCPRI headers are supported for the C/U-Plane protocol (, i.e., support of the IEEE 1914.3 header is not defined)

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

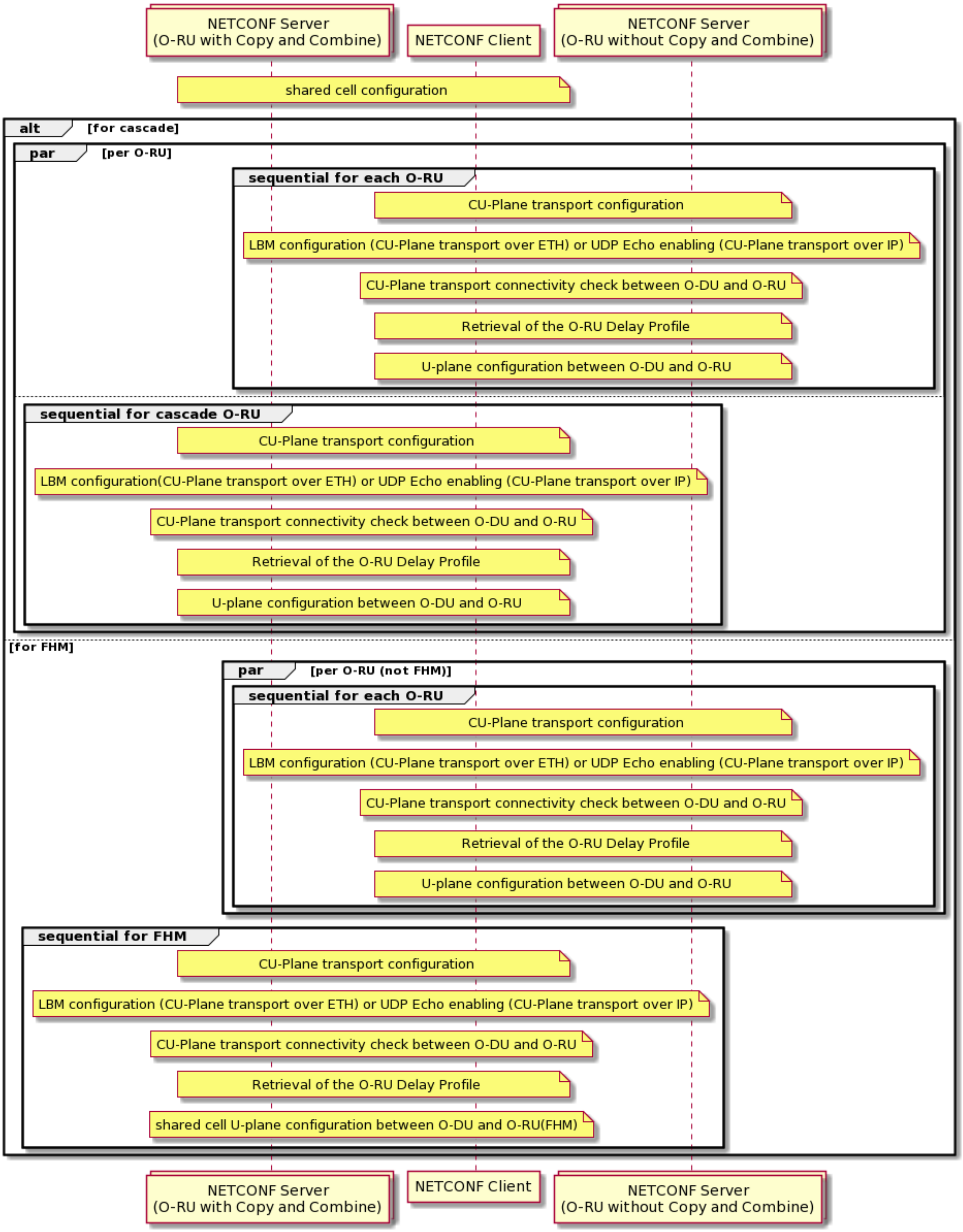


Figure 73: shared cell configuration and u-plane configuration

The NETCONF client performs further steps for the regular start-up procedure as described in chapter 3.

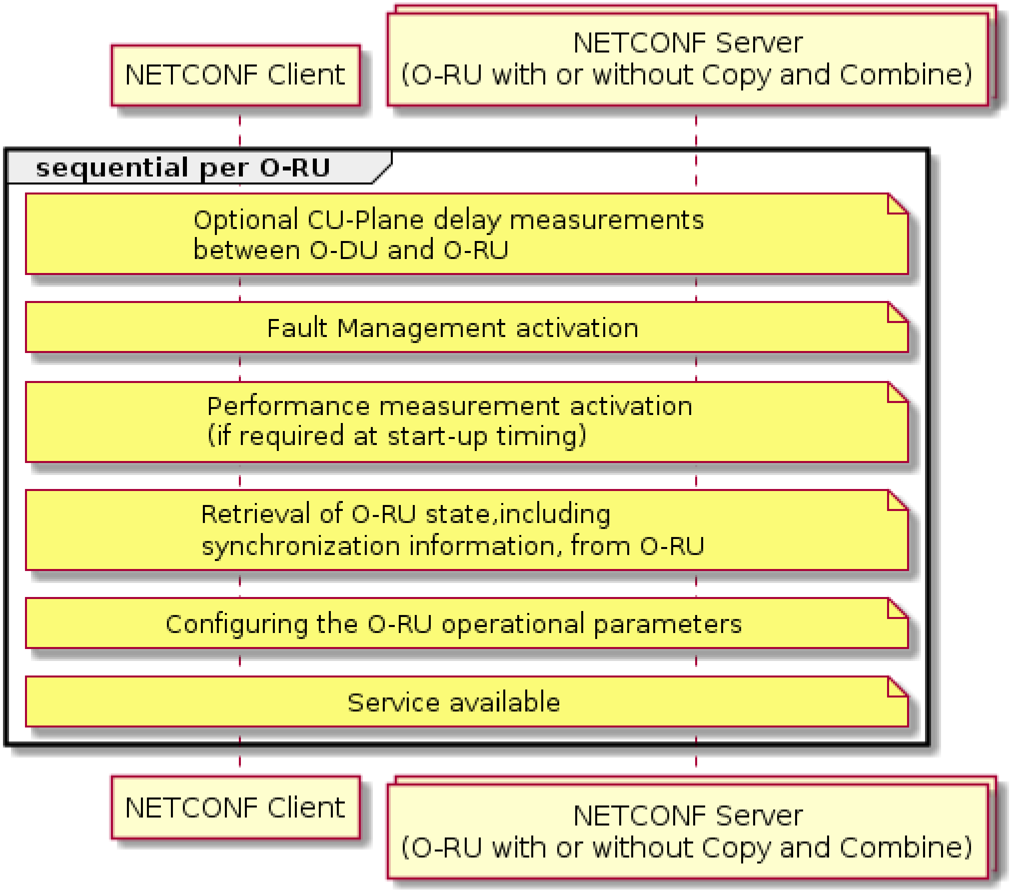


Figure 74: further steps for start-up procedure

14.3 Performance Management

This sub-section provides description of the specific part of Performance Management for shared cell.

14.3.1 Measurement-group

transceiver-stats:

O-RU (Cascade / FHM) has multiple connections to O-DU and O-RU (Cascade/Normal). The baseline O-RU models permit O-RUs to be defined with multiple ports and multiple transceiver modules. Transceiver module is defined by o-ran-transceiver which refers the port-number for these interfaces. Please refer to sub-section 14.5.1 O-RU Information for Shared Cell. In this case, the O-RU (Cascade / FHM) shall be able to report **transceiver-stats** per port-number.

Rx-window-stats:

O-RU (Cascade / FHM) shall monitor **rx-window-stats** per eaxc-id, per transport or per hardware component (O-RU) because it receives data flow from the north-node.

Note: This version of the specification does support **rx-window-stats** to monitor the downlink reception window and doesn't support monitoring by an O-RU of the uplink traffic from south-node.

Tx-stats:

O-RU (Cascade / FHM) shall monitor **tx-stats** per eaxc-id, per transport or per hardware component (O-RU) because it transmits data flow to the north-node.

Note: This version of the specification does support **tx-stats** to monitor uplink traffic and doesn't support monitoring by an O-RU of the downlink traffic to south-node.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Epe-stats:

O-RU (Cascade / FHM) shall monitor **epe-stats** per hardware component.

Measurement-group	measurement-units
transceiver-stats	port-number (multiple)
rx-window-stats	eaxc-id, transport or hardware component (O-RU)
tx-stats	eaxc-id, transport or hardware component (O-RU)
epe-stats	hardware component

Table 7: Measurement-group of O-RU (Cascade / FHM)

For more detail, please refer to Table 8 Counters definition in Annex B.

14.4 Delay Management

In the shared cell environment, the use of O-RU Adaptive Delay capability is not permitted. The O-DU and each O-RU have their own delay parameters and supported transmission window and reception window. Also, the topology of the O-RU configuration can be detected by the topology discovery procedure.

The O-DU can determine the delay budget between itself and the O-RUs considering the O-RUs' topology and delay parameters and its own transmission window and reception window. During this discovery, the required processing time of the O-RU is also considered. The O-RU processing time includes the copy operation for downlink operation, and the combine operation for uplink operation, and the delay periods for the copy and combine operations are defined as **t-copy** and **t-combine** in o-ran-shared-cell.yang module:

t-copy: Corresponding to the maximum FHM or cascade O-RU processing delay between receiving an IQ sample over the fronthaul interface from the north-node, coping it and transmitting it over the fronthaul interface to the south-node.

t-combine: Corresponding to the maximum FHM or cascade O-RU processing delay between receiving an IQ sample over the fronthaul interface from the south-node(s), combing them and transmitting it over the fronthaul interface to the north-node.

Therefore, based on the above information, the O-DU can determine how many O-RUs are configured to operate in a shared cell instance.

After the delay budget between the O-DU and the furthest (southern-most) O-RU in the chain is determined, multiple O-RUs can be configured to operate in between the O-DU and the furthest O-RU. The time budget between the O-DU and the furthest O-RU is constant and is shared for all O-RUs operating in cascade mode.

For combine function operation, the O-RU shall await the successful reception of the eCPRI frame(s) from the south-node(s). Once the FHM receives the eCPRI frame from all of the south-nodes, the O-RU (FHM) can perform the combine operation. Once the cascade O-RU receives the eCPRI frame from the south-node, the O-RU can perform the combine operation using the eCPRI frame and received radio information. The maximum time an O-RU is permitted to wait for the required eCPRI frames is set by the **ta3-prime-max** configured by NETCONF client. If the O-RU cannot commence the combination procedure until a time after the configured **ta3-prime-max** minus static **t-combine**, e.g., due to the delayed reception of eCPRI frame(s), the O-RU shall discard the delayed eCPRI frames if received and combine other received frames (for FHM mode) or radio information (for Cascade mode). The configurable **ta3-prime-max** shall be equal to or less than **ta3-prime-max-upper-limit** which is the capability of O-RU, related to the internal memory for combine operation. The detail for C/U-plane aspects is described in chapter 11.3 of [2].

ta3-prime-max: The latest time that FHM or cascade O-RU is allowed to send UL U-plane message to north-node relative to reception timing at O-RU antenna.

ta3-prime-max-upper-limit: The upper limit for the configurable ta3-prime-max value. This is the capability information of O-RU that comes from the O-RU internal memory for the combine operation.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

14.5 Details of O-RU operations for shared cell

This sub-section provides the function detail of O-RU operations for a shared cell.

14.5.1 O-RU Information for Shared Cell

Interfaces to south-node of O-RU with Copy and Combine function

Cascade O-RU has one additional transport interface to south-node and FHM has more than one additional transport interfaces to south-nodes, illustrated below.

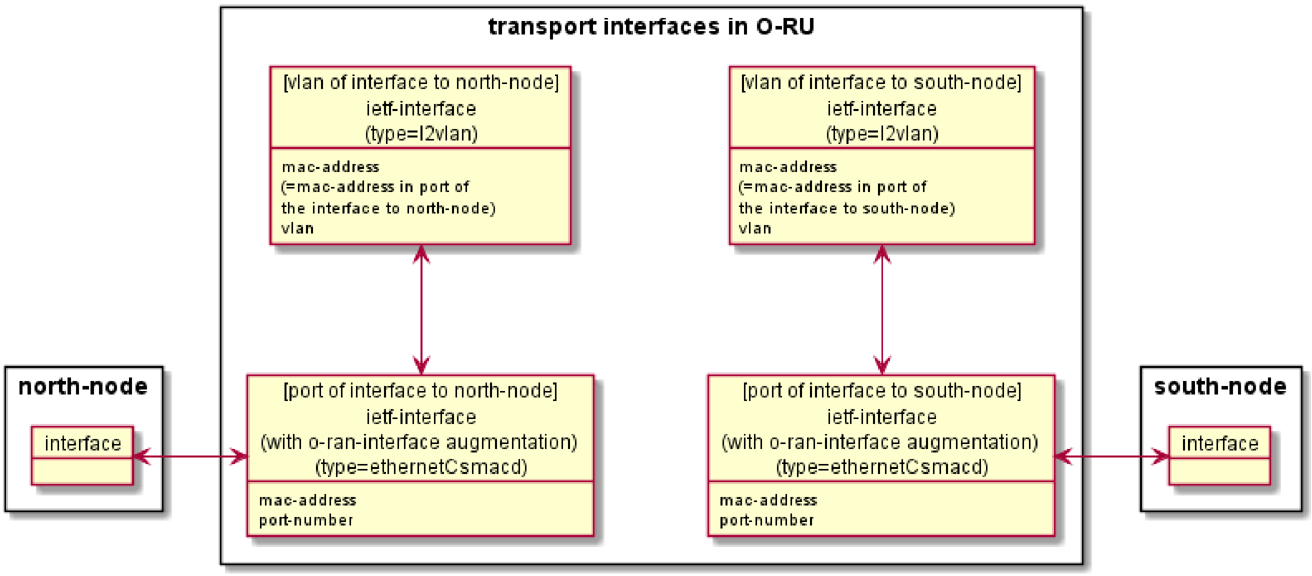


Figure 75: Transport Interfaces of both side of O-RU

Both the interface to north-node and the interface to south-node can be defined by ietf-interface YANG model with **type** = ethernetCsmacd, augmented by o-ran-interface for **mac-address** and **port-number**. Transceiver module is defined by o-ran-transceiver which refers the **port-number** for these interfaces. The maximum number of the interfaces is just the number of physical interfaces within the O-RU.

The role of interfaces shall be detected by topology discovery procedure described in subsection 14.5.2.

If the O-RU has any interfaces to south-node and if they are utilized for shared cell scenario, the NETCONF client shall configure the higher layer ietf-interface (**type** =l2vlan) including configuring the corresponding **mac-address** and C/U-plane **vlan-id** configuration for each ietf-interface (**type** = ethernetCsmacd) to south-node, in addition to the higher layer ietf-interface (**type** = l2vlan) configured for the ietf-interface (**type** =ethernetCsmacd) to north-node in section 4.3.

If an interface to a south-node is not used for shared cell scenario, the NETCONF client doesn't need to configure the higher layer ietf-interface (type = l2vlan) for it.

Capability of O-RU with Copy and Combine function

The configuration for Copy and Combine function is defined in the o-ran-shared-cell.yang module. The presence of this yang module signalled in O-RU's YANG library indicates that O-RU can support the copy and combine function. The **shared-cell-module-cap** container includes the information for the internal maximum processing delay for both the copy function and the combine function required for delay management operations. The **shared-cell-module-cap** container also includes the information defining the maximum numbers of copy and combine functions supported. This information is used by the NETCONF client to determine how many south-nodes can be supported and how many eaxc-ids can be used for copy and combine procedures. It also contains the information defining the **compression** capability supported by the FHM.

For the cascade mode, the cascade O-RU shall support normal O-RU operations, i.e., radio transmission and reception. For the FHM mode, the FHM doesn't have the capability for radio transmission and reception. The o-ran-shared-

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

cell.yang module defines the feature **FHM** to indicate that O-RU acts as FHM and doesn't have the capability of radio transmission and reception.

Yang modules for FHM mode

Especially for the FHM mode, some of the yang modules are not necessary because the FHM doesn't have the capability for radio transmission and reception. The following yang modules are not applicable for O-RU (FHM). For more detail, please see Annex C.1.

- o-ran-ald module and o-ran-ald-port: Antenna Line device is directly connected to O-RU.
- o-ran-laa-operations and o-ran-laa: out-scope for LAA and radio transmission related only.
- o-ran-module-cap: radio transmission related parameters only
- o-ran-beamforming: radio transmission (beamforming) specific parameters only
- o-ran-uplane-conf: radio transmission (uplane configuration) specific parameters only.

14.5.2 Topology Discovery procedure

The O-DU shall determine the topology (adjacency relationships) of O-RUs for the shared cell. In this version of the specification, only Ethernet based transport of C/U sessions for shared cells is supported. The shared cell topology discovery procedure operates at the Ethernet layer, using the LBM/LBR connectivity checking procedure defined in sub-section 4.6 to fill the Ethernet Forwarding Information Base (FIB), and is based on the O-DU recovering the FIB. The transparent bridge functionality is used in the shared cell capable O-RUs operating in FHM and/or cascade mode.

In a typical operation, an O-DU performs a two-stage procedure for determining the topology:

I. In a first stage, the O-DU performs Ethernet connectivity monitoring on all discovered O-RU MAC addresses, using the procedures defined in sub-section 4.6.1. The sending of the Ethernet loopback responses from the discovered MAC addresses ensures that the transparent bridges in the O-RUs operating in FHM and cascade mode will automatically learn the MAC addresses switched through these devices.
Note: In addition to Ethernet connectivity monitoring, DHCP discovery, call home and M-plane connection establishment can be also be used to populate information in the FIB and so may be sufficient when an O-RU only has a single port configured for interfacing to its north-node.

II. In a second stage, the O-DU uses the o-ran-ethernet-forwarding.yang module to discover which MAC addresses have been learnt by the Ethernet bridge functionality. The O-DU uses the individual Ethernet forwarding table entries to determine the adjacency relationships.

The following figure illustrates an example of the MAC address configuration for a set of O-RUs configured in cascade mode and FHM mode of operations, together with the associated Transparent Bridge FIB tables.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

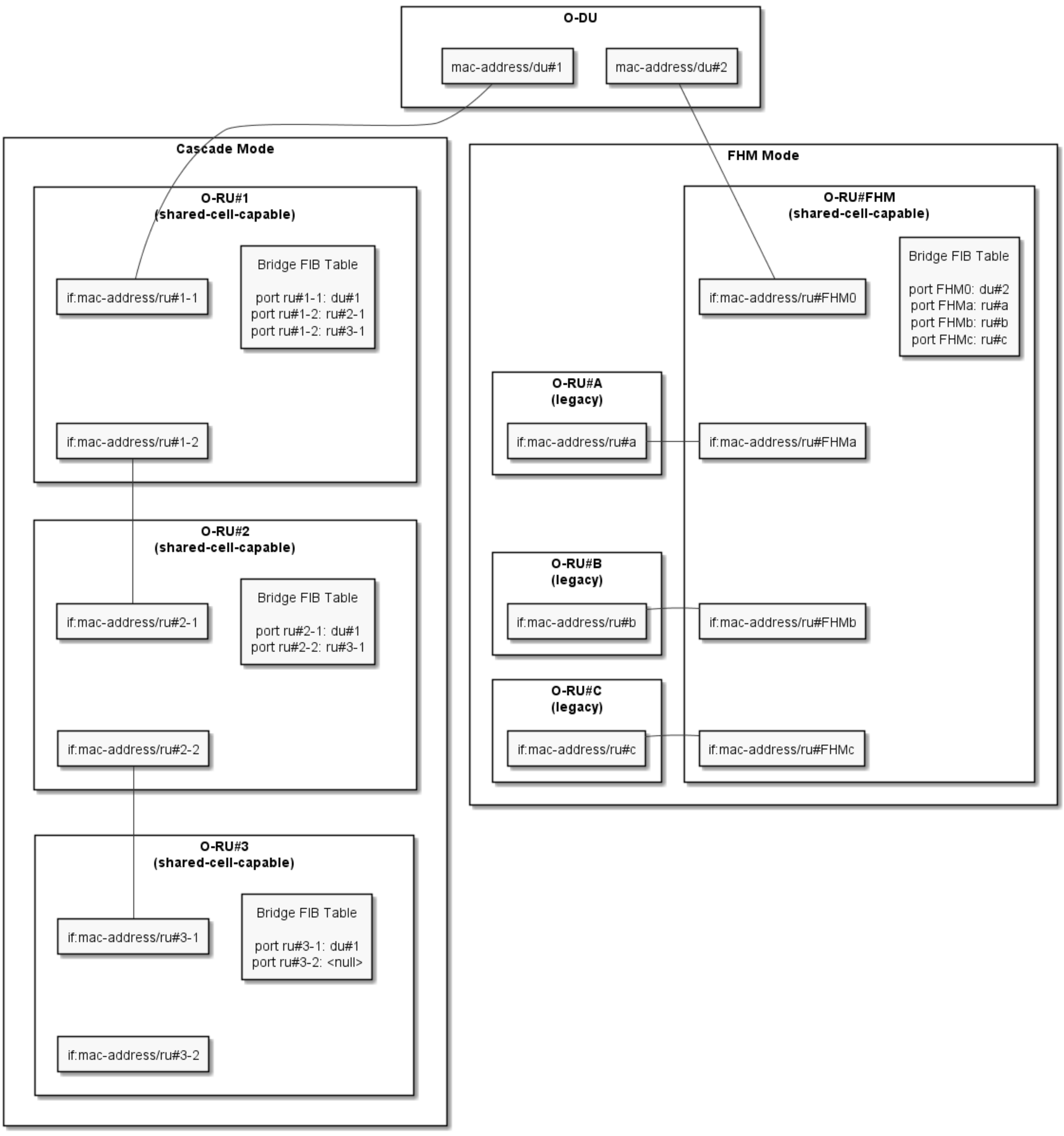


Figure 76: Bridge FIB table entries learned from LBM/LMR connectivity check procedures

14.5.3 Shared Cell Configuration

Shared cell configuration consists of shared cell copy entities and shared cell combine entities. For the shared cell configuration, the choice-case statement **shared-cell-copy-combine-model** is used for future enhancement.

Both shared cell copy entities and shared cell combine entities define the **transport-flows** for the processing elements of the interface to north-node and the interface to south-node. The combine entities contain the **ta3-prime-max** to be used in delay management as described in sub-section 14.4.

As per conventional o-ran-processing-element.yang, the **transport-flow** (**eth-flow**) of the processing element is a combination of **o-du-mac-address**, **ru-mac-address** and **vlan-id** for legacy **eth-flow**.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

The O-RU management plane introduces 2 eth-flow options for shared cell scenario:

- The **transport-flow** definition for the interface to north-node: **north-eth-flow** is a combination of **north-node-mac-address**, **ru-mac-address** and **vlan-id**.
- The **transport-flow** definition for the interface to south-node: **south-eth-flow** is a combination of **south-node-mac-address**, **ru-mac-address** and **vlan-id**.

Either legacy **eth-flow** or **north-eth-flow** can be used for last (southern) O-RUs in star or chain topology.

The leaf **o-du-mac-address, north-node-mac-address, south-node-mac-address** and **ru-mac-address** are configured as follows:

FHM mode:

- **north-eth-flow** for the interface to north-node:
  - **north-node-mac-address**: MAC address of the north-node (O-DU)
  - **ru-mac-address**: MAC address of the interface to north-node in O-RU(FHM)
  - **vlan-id**
- **south-eth-flow** for the interface to south-node:
  - **south-node-mac-address**: MAC address of the south-node (O-RU)
  - **ru-mac-address**: MAC address of the interface to south-node in O-RU(FHM)
  - **vlan-id**

Note: same **vlan-id** is configured on both sets of processing elements.

Cascade mode:

- **north-eth-flow** for the interface to north-node:
  - **north-node-mac-address**: MAC address of the north-node (O-DU or O-RU interface to south-node)
  - **ru-mac-address**: MAC address of the interface to north-node in O-RU(cascade)
  - **vlan-id**
- **south-eth-flow** for the interface to south-node:
  - **south-node-mac-address**: MAC address of the south-node (O-RU interface to north-node)
  - **ru-mac-address**: MAC address of the interface to south-node in O-RU(cascade)
  - **vlan-id**

Note: same vlan-id is configured on both sets of processing elements.

Southern O-RU for FHM or Cascade mode:

- **north-eth-flow** for the interface to north-node: (Shared cell capable O-RU case)
  - **north-node-mac-address**: MAC address of the north-node (O-RU(FHM) interface to south-node)
  - **ru-mac-address**: MAC address of the interface to north-node in O-RU
  - **vlan-id**

or

- **eth-flow** for the interface to north-node: (Non shared cell capable O-RU case)
  - **o-du-mac-address**: MAC address of the north-node (O-RU(FHM) interface to south-node)
  - **ru-mac-address**: MAC address of the interface to north-node in O-RU
  - **vlan-id**

Copy and Combine functions are disabled if not configured, meaning the functions are disabled by default.

The following figure illustrates the shared cell configuration and transport configuration.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

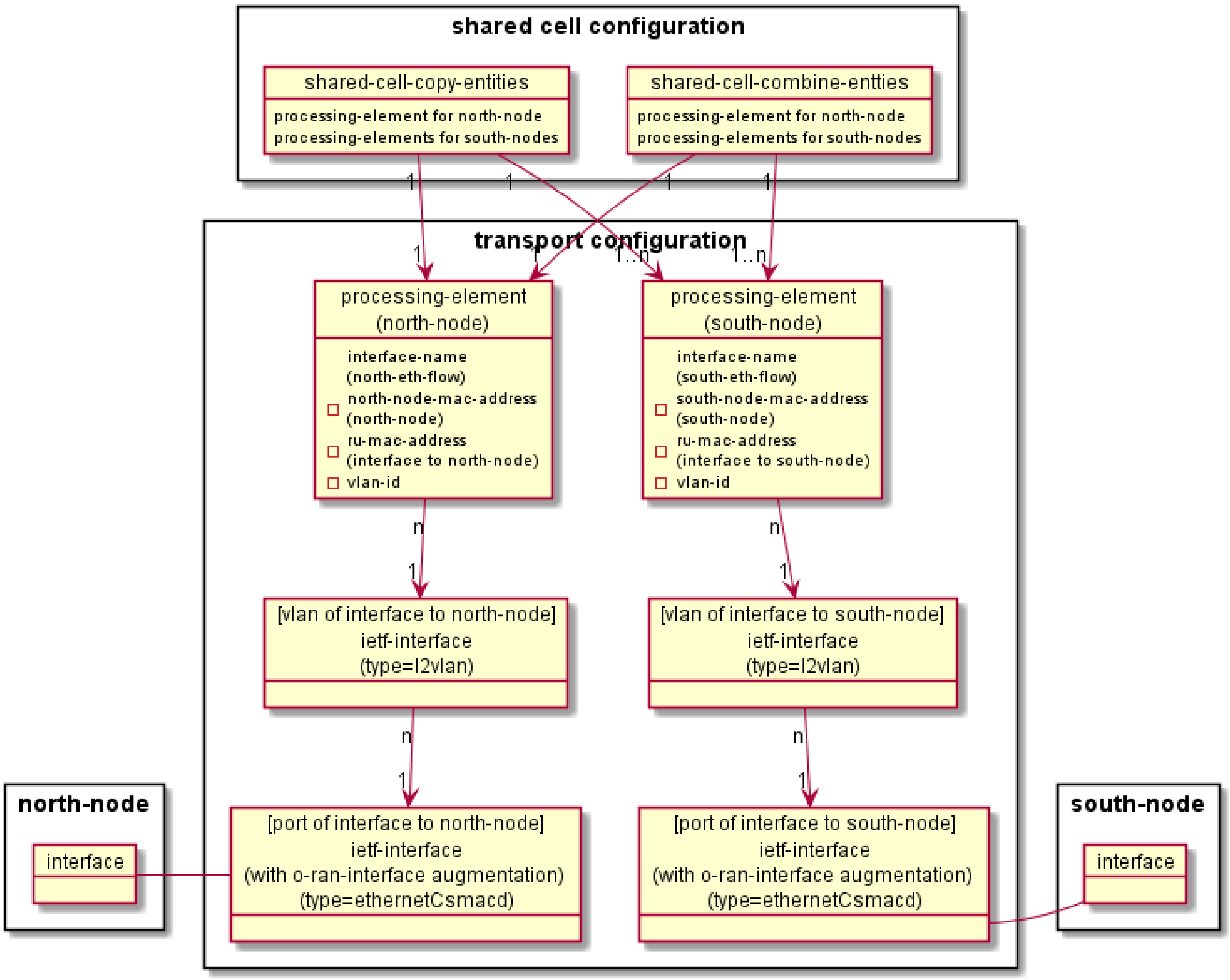


Figure 77: Relation of Shared Cell Copy and Combine Entities and Transport Configuration

For the cascade mode, the processing element for north-node will be connected to the **low-level-tx(rx)-links** in u-plane configuration as in the Figure 50 in chapter 12.2.3. In the O-RU (FHM), there is no radio transmission. The following figure illustrates the example of topology diagram and shared cell copy/combine entities. Each blue thick line indicate the **transport-flow** in the processing element.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

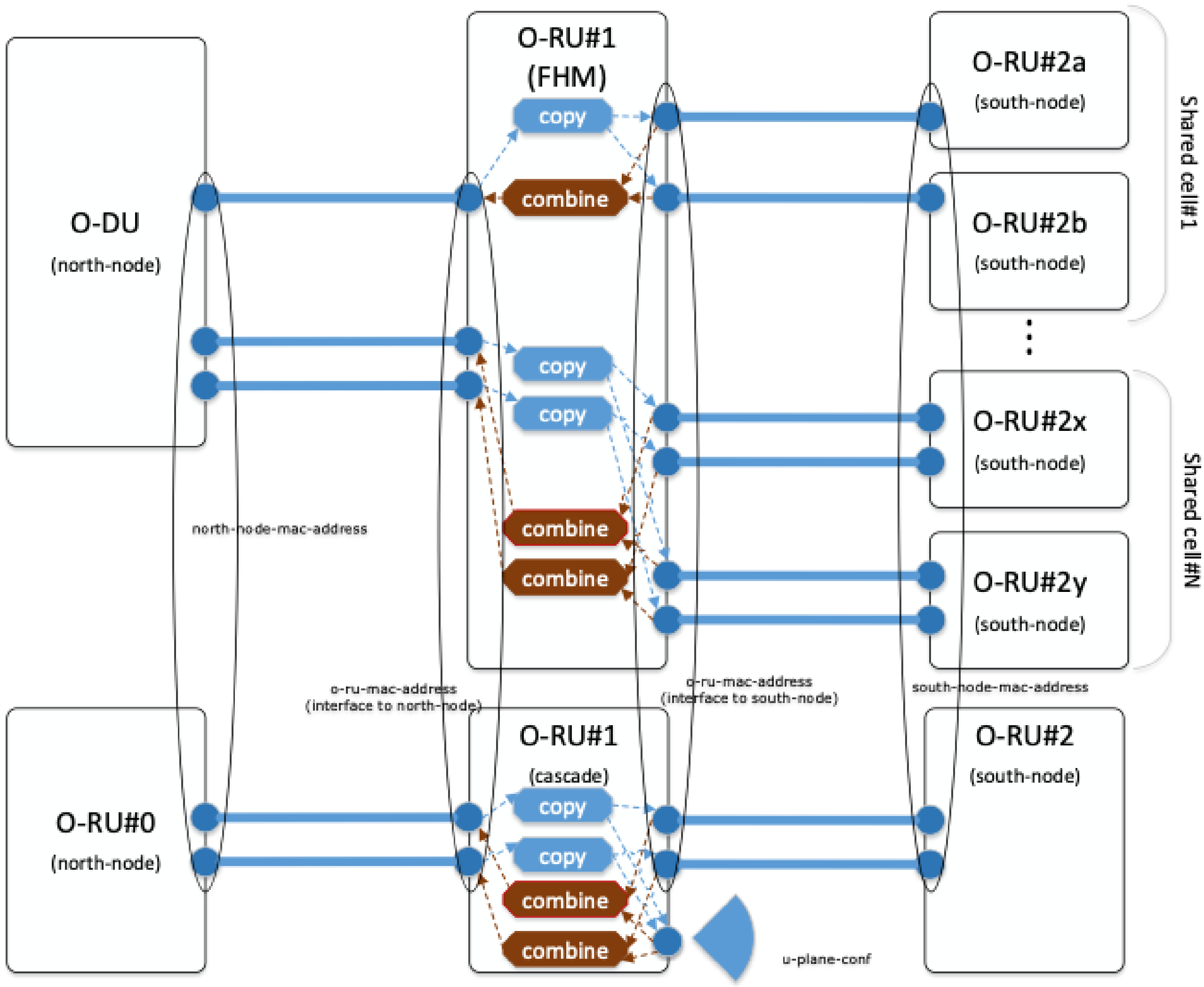


Figure 78: Example of Topology and Shared cell Copy/Combine entities

The following is the information for this example of topology configuration.

FHM mode)

One **transport-flow** for north-node corresponds to the two **transport-flow**s for south-nodes: O-RU#2a and 2b respectively. O-RU#2a and #2b have same u-plane configuration.

Two **transport-flow**s for north-node correspond to the two times two **transport-flow**s for south-nodes: O-RU#2x and #2y respectively. O-RU#2x and #2y have same u-plane configuration.

Cascade mode)

Two **transport-flow**s for north-node correspond to two **transport-flow**s for south-node and own u-plane-configuration. O-RU#1 and O-RU#2 have same u-plane configuration.

Note: “two **transport-flow**s” above is just the example scenario that two optical physical lines are used for fronthaul connection, which has been supported in this specification. The total capacity of the interfaces to north-node for FHM is assumed that required traffic can be transported for multiple shared cells. O-RU Controller shall ensure that the capacity of any link shall not be exceeded due to copy and combine configuration.

<u>Information only for the C/U-plane behaviour as the background at COMMON/SELECTIVE-BEAM-ID/SELECTIVE shared cell copy and combine mode.</u>

For the traffic from the north-node to O-RU (FHM / cascade), there are DL C-plane and U-plane traffic and UL C-plane traffic in the transport interface. When operating with **COMMON** shared cell copy and combine mode, all C/U-plane traffic received from north-node is copied and forwarded to the south-node(s). For the traffic from south-node, there is

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

UL U-plane traffic only in each transport interface to south-node. When operating with **COMMON** shared cell copy and combine mode, all U-plane traffic received from the south-node(s) is combined and forwarded to the north-node. It is assumed that common compression mechanism for uplink is configured by M-plane for all O-RUs in one shared cell network.

When operating with **SELECTIVE** or **SELCTIVE-BEAM-ID** shared cell copy and combine mode, some selected C/U-plane traffic received from the north-node are copied and forwarded to the south-node and some selected U-plane traffic received from south-node are combined and forwarded to the north-node. This version of the specification supports **SELECTIVE-BEAM-ID**. **SELECTIVE** will be supported in a future version.

14.5.4 U-plane Configuration for FHM mode

For the FHM mode, O-RU (FHM) doesn't need to have u-plane configuration defined in o-ran-uplane-conf.yang module. Instead, O-RU (FHM) needs to have shared-cell specific u-plane configuration. The **shared-cell-copy-uplane-config** is the **eaxc-id** list used by DL C-plane, DL U-plane and UL C-plane traffic. It also contains **downlink-radio-frame-offset** and **downlink-sfn-offset** to define the downlink timing of t=0 for the reception window. The **shared-cell-combine-uplane-config** is the **compression** method of the UL U-plane traffic applied to O-RUs within the shared cell network in shared cell combine function. The **compression** method is configurable per **eaxc-id**. It also contains **downlink-radio-frame-offset**, **downlink-sfn-offset** and **n-ta-offset** to define the uplink timing of t=0 for the configured **ta3-prime-max**. In addition, **number-of-prb** is also contained for the case that all PRBs in numPrbc are controlled by C-plane message.

FHM may have multiple sets of shared cell networks as described in the Figure 78 E.g., O-RU#2a and O-RU#2b are one shared cell network. O-RU#2x and O-RU#2y are another shared cell network. These two shared cell networks are separated transport layer level definitions using separate processing-element/transport-flows. Nevertheless, the O-RU Controller shall ensure that the eaxc-id allocation for the shared cell networks shall be unique per link (downlink or uplink) in one O-RU (FHM). NETCONF client (O-RU Controller) shall ensure to allocate unique eaxc-id for O-RU(s) per link within the shared cell network(s) in one O-RU (FHM).

Note: **shared-cell-copy-uplane-config** and **shared-cell-combine-uplane-config** are not applicable to the cascade mode. Instead, o-ran-uplane-conf.yang module is applied.

Note: if the FHM supports the C/U-plane monitoring timer described in section 4.10, then depending on how long the O-DU takes to initiate sending of C/U plane data flows, it may be advisable for the NETCONF client to initially disable the operation of the timer for the FHM before carrier activation to O-RUs that are south-nodes for the FHM. Such an approach will avoid the FHM sending spurious alarm notifications triggered by O-DU delays in initializing the sending of C/U plane data that exceed the default timer value. Once C/U plane data flows have commenced, the NETCONF client can re-configure the timer with the desired value and hence activate monitoring of the C/U plane connectivity by the FHM.

14.5.5 Support of Selective Transmission and Reception Function

This section describes M-Plane support for selective transmission and reception function which is specified in chapter 11.2.1 of [2]. There are two things to be introduced for supporting the function from M-Plane perspective;

- Feature which indicates FHM support for selective transmission and reception function
- Configuration parameters which indicate the mapping information between global beamId, O-RU(s) and O-RU local beamId.

If FHM indicates the feature "SELECTIVE-BEAM-ID" to O-DU, O-DU can configure the mapping information between global beamId, O-RU(s) and O-RU local beamId to the FHM to use selective transmission and reception function. For configuring the information, **mapping-table-for-selective-beam-id** is used.

14.6 Cascade-FHM mode

The introduction of Cascade-FHM mode is described in chapter 11.5 of [2]. This section describes mainly shared cell configuration on cascaded FHMs. For other common parts, like startup procedures, topology discovery procedure etc., please refer to relevant parts in chapter 14. In this version of the specification, the maximum level of cascaded FHMs is limited to 2. The first cascade FHM nearest to O-DU is named FHM#1, the second FHM is named FHM#2.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

14.6.1 Shared Cell Configuration on cascaded FHMs

The NETCONF client needs to configure shared cell copy entities and shared cell combine entities on FHM#1 and FHM#2 respectively.

In Figure 79, there are two types of FHM to FHM traffic: 1) Type 1: the cell of traffic is on O-RUs of both FHM#1 and FHM#2. For example, the Cell#1 of Same cell scenario; 2) Type 2: the cell of traffic is only on O-RUs of FHM#2. For example, the Cell#2 of Two cells Scenario.

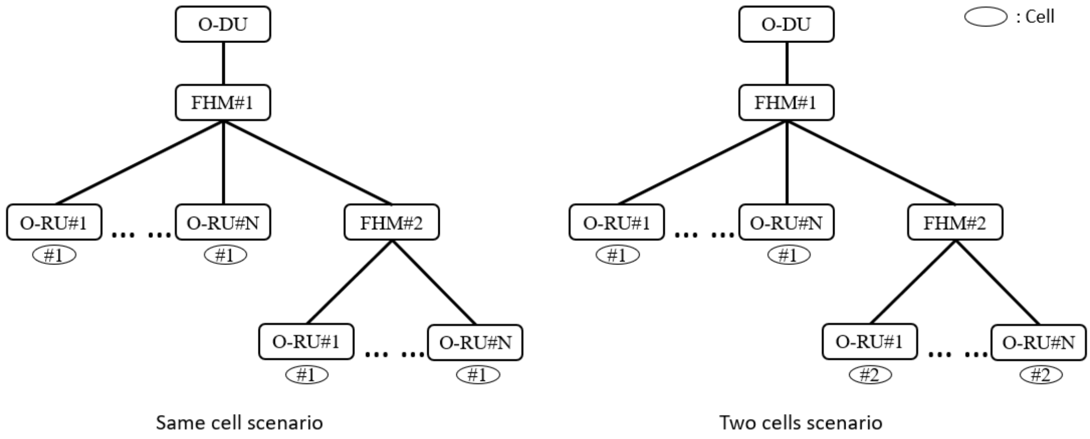


Figure 79: Typical cell scenarios in Cascade-FHM mode

For Type 1, taking Cell#1 of Same cell scenario as example, the shared cell configuration on FHM#1 and FHM#2 are as follows:

FHM#1:

- In DL direction, it needs one element of **shared-cell-copy-entities** with one of south nodes connecting to FHM#2 and other south nodes connecting to the O-RUs serving FHM#1, and with north node connecting to O-DU. NETCONF Client selects eaxc-ids carried by the CU-Plane messages who will go to FHM#2 and fills them to eaxc-id list of shared-cell-copy-uplane-config.
- In UL direction, it needs one element of **shared-cell-combine-entities** with one of south nodes connecting to FHM#2 and other south nodes connecting to the O-RUs serving FHM#1, and with north node connecting to O-DU. NETCONF Client selects eaxc-ids carried by the U-Plane messages who are from FHM#2 and fills them to eaxc-id list of shared-cell-combine-uplane-config.

FHM#2:

- In DL direction, it needs one element of **shared-cell-copy-entities** with north node connecting to FHM#1 and south nodes connecting to the O-RUs serving FHM#2. NETCONF Client selects eaxc-ids carried by the CU-Plane messages who are from FHM#1 and fills them to eaxc-id list of shared-cell-copy-uplane-config.
- In UL direction, it needs one element of **shared-cell-combine-entities** with north node connecting to FHM#1 and south nodes connecting to the O-RUs serving FHM#2. NETCONF Client selects eaxc-ids carried by the U-Plane messages who will go to FHM#1 and fills them to eaxc-id list of shared-cell-combine-uplane-config.

For Type 2, taking Cell#2 of Two cells scenario as example, the shared cell configuration on FHM#1 and FHM#2 are as follows:

FHM#1:

- In DL or UL direction, existing only one south node which connects to FHM#2 and other steps are no difference with FHM#1 of Type 1;

  Note: in Two cells scenario, there will be two elements of shared-cell-copy-entities in DL direction in FHM#1, one is for Cell#1 and another is for Cell#2. Similarly, there will be two elements of shared-cell-combine-entities in UL direction in FHM#1, one is for Cell#1 and another is for Cell#2.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

FHM#2:

- There is no difference with FHM#2 of Type 1.

Both FHM#1 and FHM#2 don't need to have u-plane configuration defined in o-ran-uplane-conf.yang module since no radio transmission on the two FHMs.

Chapter 15 Configured Subscriptions

15.1 Introduction

The support by an O-RU of configured subscriptions is an optional capability, advertised by the O-RU indicating it supports the **ietf-subscribed-notifications** YANG model [37] in its YANG library together with the **configured** feature. This capability enables an O-RU Controller to install a subscription via configuration of the O-RU's datastore. Importantly, the lifetime of such a configured subscription is not limited to the lifetime of the NETCONF session used to establish it, enabling a configured subscription to persist even when an O-RU has been temporarily disconnected from the network.

The **ietf-subscribed-notifications** YANG model defines a transport agnostic mechanism for subscribing to and receiving content from an event stream in an O-RU. An O-RU that supports configured subscriptions shall also support the **encode-json** feature together with the augmentation of the **ietf-subscribed-notifications** YANG model by the **o-ran-ves-subscribed-notifications** YANG model.

15.2 Description

An O-RU controller discovers the event-streams supported by an O-RU. The O-RU Controller then establishes a configured subscription to a particular event-stream.

> Note, the same NACM privileges defined in Sub-section 3.4 shall also be used by the O-RU in determining whether an O-RU controller has privileges to establish a configured subscription to a particular event-stream.

Based on configured subscriptions, the O-RU sends asynchronous notifications over HTTPS to the configured Event-Collector. This capability can be used with any existing YANG notification, e.g., defined in YANG models published by the O-RAN Alliance or imported from other organizations.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

15.3 Procedure

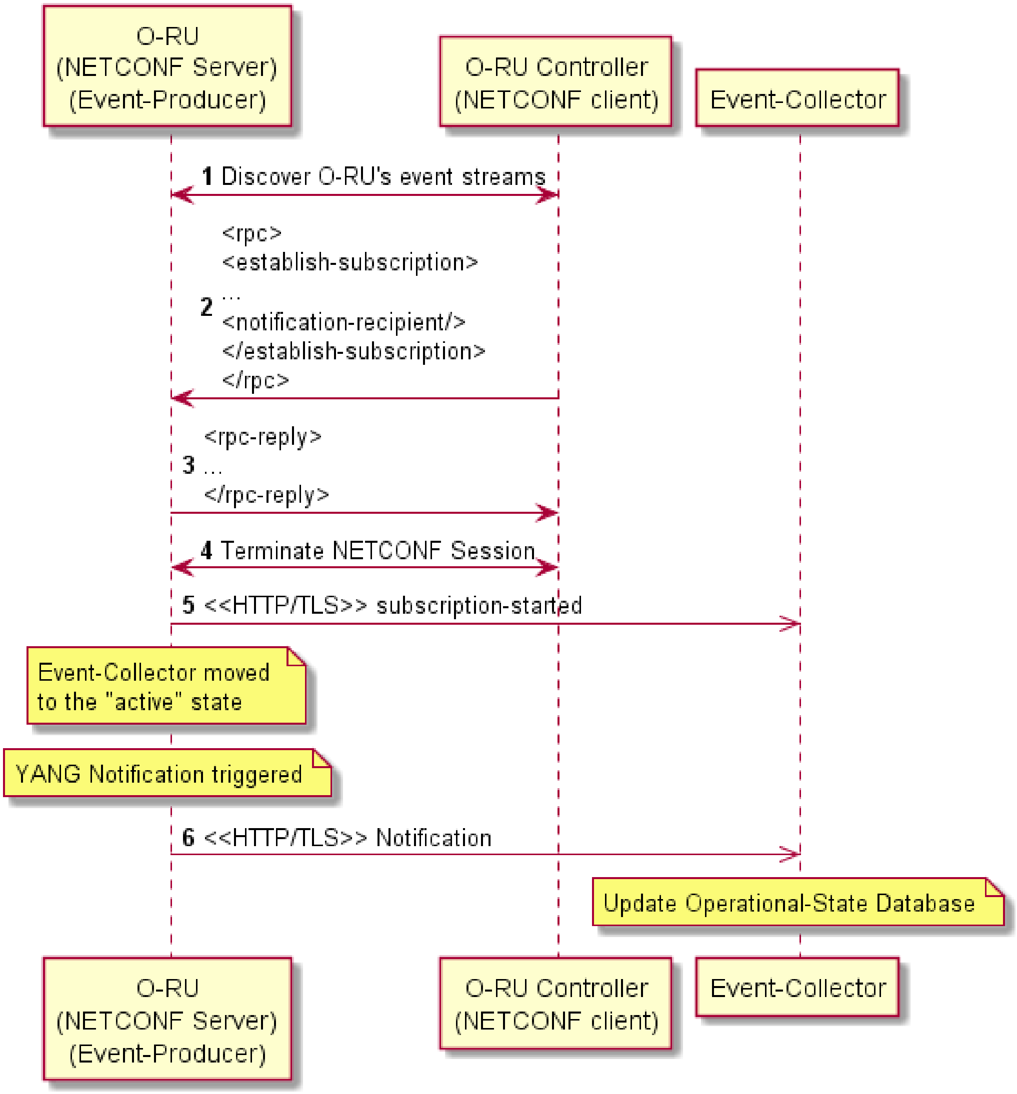


Figure 80:Message sequence exchange for provisioning a configured notification

Pre-condition: A NETCONF sessions is established between the O-RU and an O-RU Controller.

1. The O-RU controller gets the **streams** container and discovers the event-streams supported by an O-RU
2. The O-RU controller uses the “establish-subscription” RPC to configure a subscription to an event-stream. The RPC includes the **receiver** container augmented with the **notification-recipient** schema node that encodes the URI of the Event-Collector.
3. If the O-RU controller has the correct privileges, the O-RU accepts the configured subscription
4. As the lifetime of the configured subscription is not limited by the lifetime of the NETCONF session, the O-RU controller can terminate the NETCONF session without causing the subscription to be suspended.
5. After a subscription is successfully established, the O-RU immediately sends a "subscription-started" notification to the Event-Collector, according to RFC8639 [37] clause 2.5.
6. Upon an event that triggers a YANG notification, the O-RU sends a notification over HTTPS , according to RFC8639 clause 2.5.

Post-condition: The Event-Collector is able to update its operational-state datastore with the information received in the notification.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

15.4 Notification Encoding

The O-RU shall support JSON encoding as specified in RFC 7951 [38]. An example notification object generated using the **o-ran-file-management.yang** model and encoded following RFC 7951 is illustrated in Figure 81.


```
{ "ietf-restconf:notification": {
    "eventTime": "2020-11-11T20:20:00Z",
    "o-ran-file-management:file-upload-notification": {
      "local-logical-file-path": "/O-
RAN/PM/C201805181300+0900_201805181330+0900_ABC0123456.csv",
      "remote-file-path": "/home/pm/
C201805181300+0900_201805181330+0900_ABC0123456.csv",
      "status": "SUCCESS"
    }
  }
}
```


Figure 81:Example of a JSON encoded YANG Notification

15.5 Notification Transport

As described in RFC8639 clause 2.5.7, an O-RU supporting configured subscriptions must provide a YANG data model capturing the necessary transport-specific configuration parameters. O-RAN compliant Event-Producers shall support the **o-ran-ves-subscribed-notifications** YANG model.

For transport, the notification JSON objects encoded according to sub-section 15.4 are further encapsulated in VES events. In order to enable the notifications sent to the Event-Collector to retain their native notification format/schema as defined in O-RAN, IETF and other YANG models, the ONAP/VES header is used to enable the decoupling of the notification payload from the overall VES event format, as illustrated in Figure 81.

The VES common header shall include the following fields:

- The value of the eventName field shall be set to “ORU-YANG/<model-identifier>:<notification-identifier>”
- The value of the eventID field shall be set to “stndDefined-ORU-YANG-nnnnnnnnn", where nnnnnnnnn represents the integer key for the event
- The value of the sourceName and reportingEntityName fields shall both be set to the value of the **ru-instance-id** leaf defined in the o-ran-operations YANG model.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

```
{ "event": {
  "commonEventHeader": {
    "version": "4.1",
    "vesEventListenerVersions": "7.2",
    "domain": "stndDefined",
    "eventName": "O-RU-YANG/o-ran-file-management:file-upload-notification",
    "eventID": "stndDefined-ORU-YANG-000000249",
    "sequence": 0,
    "priority": "Normal",
    "sourceName": "vendorA_ORUAA100_FR1918010111",
    "reportingEntityName": "vendorA_ORUAA100_FR1918010111",
    "stndDefinedNamespace": "urn:o-ran:file-management:1.0"
    "startEpochMicrosec": 1605126000000000,
    "lastEpochMicrosec": 1605126000000000
  },
  "stndDefinedFields": {
    "schemaReference": "https://gerrit.o-ran-sc.org/r/gitweb?p=scp/oam/modeling.git;a=blob;f=data-model/yang/published/o-ran/ru-fh/o-ran-file-management.yang",
    "data": {
      "ietf-restconf:notification": {
        "eventTime": "2020-11-11T20:20:00Z",
        "o-ran-file-management:file-upload-notification": {
          "local-logical-file-path": ""/O-RAN/PM/C201805181300+0900_201805181330+0900_ABC0123456.csv",
          "remote-file-path": "/home/pm/C201805181300+0900_201805181330+0900_ABC0123456.csv",
          "status": "SUCCESS"
        }
      }
    },
    "stndDefinedFieldsVersion": "1.0"
  }
```


Figure 82:Example of a JSON encoded VES Event Carrying a YANG Notification

The VES events carrying the notifications are sent to the Event-Collector over HTTPS with a POST operation following the VES specification [36]. The complete protocol stack is illustrated in Figure 83.

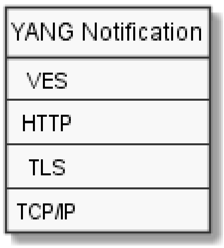


Figure 83: Protocol stack for O-RAN VES transport of YANG notifications

15.6 Monitoring the Communications Channel between O-RU and Event-Collector

An O-RU controller can use the NETCONF monitoring capability described in sub-section 3.6 to trigger the repeated sending of a **supervision-notification** by the O-RU to a subscribed O-RU Controller to ensure the channel is operational and able to transport asynchronous notifications using NETCONF. An equivalent capability is required to be supported by those O-RUs that support configured subscriptions transported using JSON/HTTPS, to enable the monitoring of the communications channel from the O-RU to the Event-Collector. The format of the heartbeat notifications is identical to the Event-Collector Notification Format determined for the operation of pnfRegistration as

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

described in sub-section 3.1.6, i.e., in this version of the specification, this capability adopts the ONAP defined guidelines for Heartbeat, as defined in [36].

15.6.1 Heartbeat Encoding

The VES common header shall include the following fields:

- The value of the sourceName and reportingEntityName fields shall both be set to the value of the **ru-instance-id** leaf defined in the o-ran-operations YANG model.

An example heartbeat encoding is illustrated in Figure 84.


```
{ "event": {
  "commonEventHeader": {
    "version": "4.1",
    "vesEventListenerVersions": "7.2",
    "domain": "heartbeat",
    "eventID": "heartbeat-00000001",
    "eventName": "heartbeat-oru",
    "sequence": 0,
    "priority": "Normal",
    "sourceName": "vendorA_ORUAA100_FR1918010111",
    "reportingEntityName": "vendorA_ORUAA100_FR1918010111",
    "startEpochMicrosec": 1605126000000000,
    "lastEpochMicrosec": 1605126000000000
  },
  "heartbeatFields": {
    "heartbeatFieldsVersion": "3.0",
    "heartbeatInterval": "60"
  }
}}
```


Figure 84:Example of a JSON encoded VES Event Carrying a Heartbeat Notification

15.6.2 Heartbeat Control

Control of the heartbeat does not use the configured subscriptions capability. An O-RU controller configures heartbeat operation using the o-ran-supervision YANG model. An O-RU Controller shall configure the **heartbeat-recipient-id** to the address(es) of the Heartbeat Event-Collector and optionally configure the **heartbeat-interval** leaf to a non-default heartbeat interval. In order to terminate operation of the monitoring the communications channel between O-RU and the Event-Collector, the O-RU shall delete the configuration in the **event-collector-monitoring** container.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

15.6.3 Heartbeat Procedure

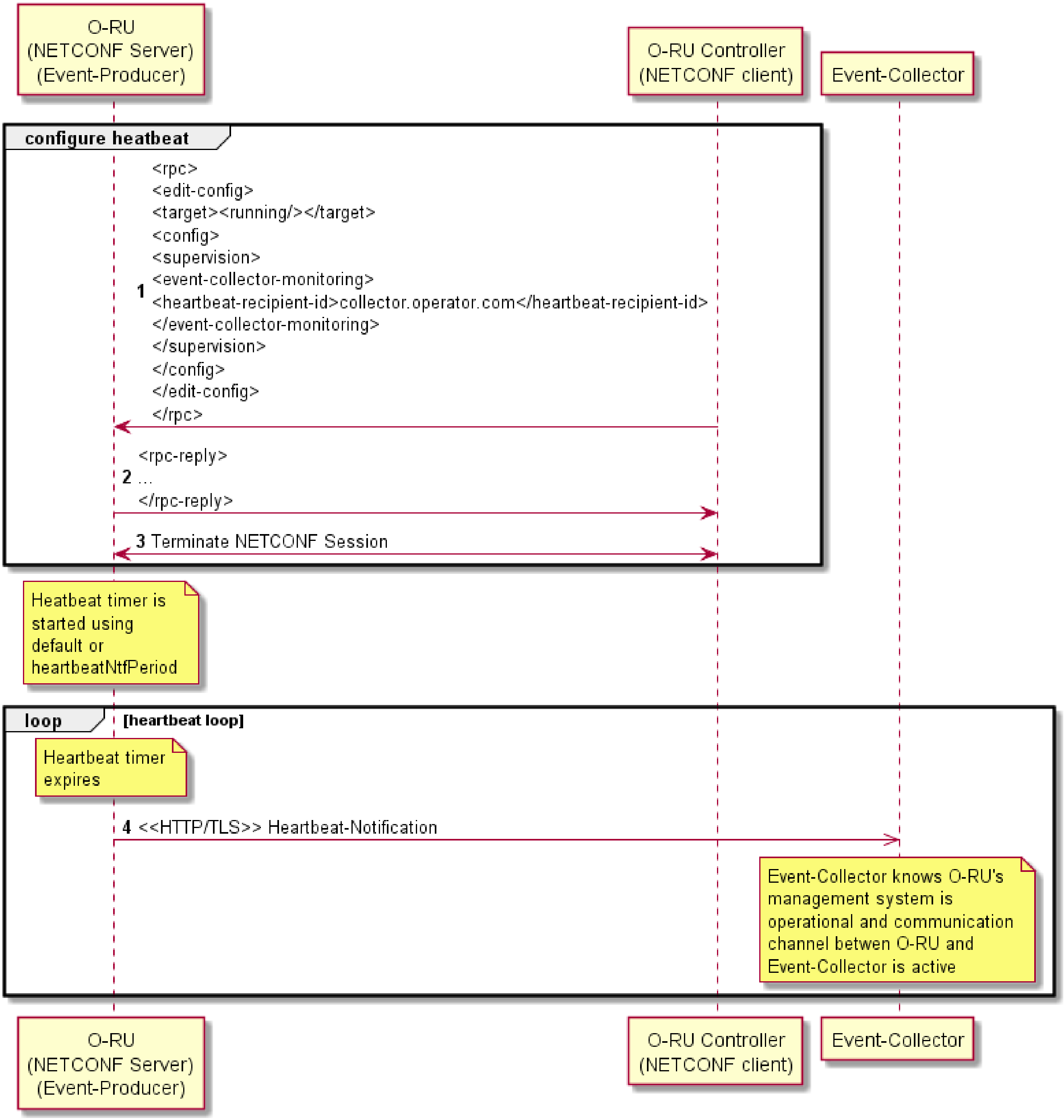


Figure 85:Message sequence exchange for heartbeat operation

In contrast to the monitoring of NETCONF connectivity described in sub-section 3.6 which define O-RU procedures when monitoring of NETCONF connectivity fails, there is no equivalent O-RU functionality defined if an O-RU determines that monitoring of the communications channel between O-RU and event-collector fails, e.g., if the O-RU is unable to establish a TLS connection to the event-collector.

Operation of the SMO when it determines that the monitoring of the communications channel between an O-RU and event-collector fails is out of scope of this specification.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Annex A Alarm definition

This section contains example alarms which may be reported.

Obviously, alarms that are not applicable in given HW design or SW configuration shall not be reported. For example, alarms related to fan monitoring are applicable to HW variants with fans.

In many cases alarm detection method is HW specific. It is assumed that alarm detection method is reliable to avoid undetected alarms and false alarms. It is also expected that the NETCONF Server is applying mechanisms to avoid unreasonably fast toggling of alarms' state.

The example alarms table has following columns:

Fault id – Numerical identifier of alarm. This ID shall be used in <alarm-notif> message (fault-id parameter).

Name – Name of the alarm.

Meaning – Description of alarm, describes high level meaning of the alarm.

Start condition – Defines conditions which must be fulfilled to generate alarm. If filtering time is needed, then it must be defined in this column.

Cancel condition – Defines conditions which must be fulfilled to cancel alarm. If filtering time is needed, then it must be defined in this column.

NETCONF Server actions on detection – Defines actions of the NETCONF Server after alarm has been detected.

NETCONF Server actions on cancel – Defines actions of NETCONF Server after alarm has been cancelled.

System recovery actions – Describes gNB level recovery actions of the NETCONF Client after alarm has been indicated by NETCONF Server. This field is informative only; actions taken by the NETCONF Client are not restricted nor defined in this document. System recovery action "Reset" refers to NETCONF Client forcing reset of O-RU.

Source – Defines possible sources of the alarm (alarm is within O-RU). The following are XML encoding provides examples of the component names defined in various YANG modules that may be alarm sources inside the O-RU.

- **< hardware xmlns= "urn:o-ran:hardware:1.0"><component><o-ran-name/></component></hardware>**
- **<fan-tray xmlns= "urn:o-ran:fan:1.0"><fan-state><name/></fan-state></fan-tray>**
- **<sync xmlns= "urn:o-ran:sync:1.0"><gnss-state><name/></gnss-state></sync>**
- **<external-io xmlns= "urn:o-ran:externalio:1.0"><input><name/></input></external-io>**
- **<external-io xmlns= "urn:o-ran:externalio:1.0"><output><name/></output></external-io>**
- **<ald-ports-io xmlns= "urn:o-ran:ald-port:1.0"><ald-port><name/></ald-port></ald-ports-io>**
- **<port-transceivers xmlns= "urn:o-ran:transceiver:1.0"><port-transceiver-data><name/></ port-transceiver-data ></port-transceivers>**
- **<interfaces xmlns= "urn:ietf:params:xml:ns:yang:ietf-interfaces"><interface><name/></interface></interfaces>**
- **<processing-elements xmlns= "urn:o-ran:processing-element.1.0"><ru-elements><name/></ru-elements></processing-elements>**
- **<-ran-uplane-conf xmlns= "urn:o-ran:uplane-conf.1.0"><low-level-tx-links><name/></low-level-tx-links></ o-ran-uplane-conf>**
- **<o-ran-uplane-conf xmlns= "urn:o-ran:uplane-conf.1.0"><low-level-rx-links><name/></low-level-rx-links></o-ran-uplane-conf>**

Note: If Source will not fit to any of above, or is empty, it means that external device (like Antenna Line Devices) cause an alarm (fault is out of the O-RU). Then additional text in alarm notification is needed to clearly say what may be possible fault source.

Severity – Defines severity of the alarm [30].

Critical – sub unit for which alarm has been generated is not working and cannot be used.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Major – sub unit for which alarm has been generated is degraded, it can be used but performance might be degraded.

Minor – sub unit for which alarm has been generated is still working.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

1

Fault id	Name	Meaning	Start condition	Cancel condition	NETCONF Server actions on detection	NETCONF Server actions on cancel	System recovery actions	Source	Severity
1	Unit temperature is high	Unit temperature is higher than expected.	Unit temperature exceeded HW implementation specific value for reasonably long filtering time (e.g. 1 minute).	Unit temperature is below HW implementation specific value for reasonably long filtering time (e.g. 1 minute).	SW implementation specific.	None.	None.	Module	Minor
2	Unit dangerously overheating	Unit temperature is dangerously high.	Unit temperature exceeded HW implementation specific value for reasonably long filtering time (e.g. 1 minute).	Unit temperature is below HW implementation specific value for reasonably long filtering time (e.g. 1 minute). AND Ambient temperature is below predefined HW implementation specific value	Unit deactivates all carriers to prevent HW damage.	None.	None.	Module	Critical
3	Ambient temperature violation	Calculated ambient temperature value goes outside the allowed ambient temperature range.	Calculated ambient temperature goes outside the allowed HW specific ambient temperature range.	Calculated ambient temperature not any more outside the allowed HW specific ambient temperature range	SW implementation specific.	None.	None.	Module	Minor
4	Temperature too low	During startup: The temperature inside the unit is too low. Heating of unit is ongoing. Wait until the alarm is canceled. During runtime: The temperature inside the module is too low.	Unit temperature is below HW implementation specific value.	Unit temperature is x Celsius above HW implementation specific value. Additionally: cancellation of critical alarm (reported during startup) is mandatory within x minutes.	HW implementation specific (e.g. enable heating). Note that actions taken must not interfere with normal unit operation if such is commanded by NETCONF Client	HW implementation specific (e.g. disable the heating). Note that actions taken must not interfere with normal unit operation if such is commanded by NETCONF Client.	None.	Module	Critical during startup Minor during runtime
5	Cooling fan broken	Fan(s) do not run.	HW implementation specific.	HW implementation specific.	None.	None.	None.	Fan supervision	Critical (if cooling is severely degraded) Major (otherwise)

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Fault id	Name	Meaning	Start condition	Cancel condition	NETCONF Server actions on detection	NETCONF Server actions on cancel	System recovery actions	Source	Severity
6	No fan detected	Unit cannot identify the used fan type or the fan is not installed at all.	HW implementation specific.	HW implementation specific.	SW implementation specific.	None.	None.	Fan supervision	Minor
7	Tuning failure	A filter has not been able to tune on an appropriate sub-band properly.	HW implementation specific.	HW implementation specific.	None.	None.	None.	Antenna line	Critical
8	Filter unit faulty	Major failure has been detected by the filter.	HW implementation specific.	HW implementation specific.	None.	None.	None.	Antenna line	Critical
9	Transmission quality deteriorated	The TX signal quality may be out of specification limits.	HW and SW implementation specific.	HW and SW implementation specific.	None.	None.	None.	Antenna line	Major
10	RF Module overvoltage protection faulty	Module's overvoltage protection is broken.	HW implementation specific.	None.	None.	None.	None.	Module	Minor
11	Configuring failed	Configuration failed because of a HW or SW fault.	SW or HW fault detected during configuration.	None.	None.	None.	Reset.	Module	Critical
12	Critical file not found	Critical configuration file is missing.	Critical configuration file is detected missing.	None.	None.	None.	Reset.	Module	Critical
13	File not found	Non-critical configuration file is missing.	Non-critical configuration file is detected missing.	None.	None.	None.	None.	Module	Major
14	Configuration file corrupted	conflicting or corrupted configuration data.	conflicting or corrupted configuration data detected.	Unit detects that previously missing file is present.	None.	None.	None.	Module or antenna line	Major
15	Unit out of order	The Unit is out of order because of a software or hardware fault.	HW and SW implementation specific.	HW and SW implementation specific.	None.	None.	Reset.	Module	Critical
16	Unit unidentified	The permanent memory in the module is corrupted and the module product code or serial number is missing, or the module product code is unknown.	Not able to read data from information storage or data is such that module identity or serial number is missing or module identity is unknown.	None.	None.	None.	None.	Module	Major

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Fault id	Name	Meaning	Start condition	Cancel condition	NETCONF Server actions on detection	NETCONF Server actions on cancel	System recovery actions	Source	Severity
17	No external sync source	The Unit lost lock to the incoming clock.	HW implementation specific.	HW implementation specific.	None.	None.	Reset.	Module	Major
18	Synchronization Error	Unit is out of synchronization.	HW implementation specific.	HW implementation specific.	Unit shuts down all RF emission to prevent environment distortion and deactivates all carriers.	None.	Reset or none.	Module or antenna line	Critical
19	TX out of order	TX path is not usable.	HW implementation specific.	HW implementation specific.	None.	None.	Reset or none.	Antenna line	Critical
20	RX out of order	RX path is not usable.	HW implementation specific.	None.	None.	None.	Reset or none.	Antenna line	Critical
21	Increased BER detected on the optical connection	Increased bit error rate has been detected on the optical link which results in sporadical errors in downlink baseband processing.	HW implementation specific (the detected BER on optical link is degrading RF operation).	HW implementation specific (the detected BER on optical link is not degrading RF operation).	Module Agent starts HW implementation specific recovery to keep the RF operation ongoing.	Module Agent stops HW implementation specific recovery actions.	None.	Module	Major
22	Post test failed	Power-on self test failed at start-up.	HW and SW implementation specific.	None.	Unit reset x times for recovery.	None.	None.	Module	Critical
23	FPGA SW update failed	The FPGA software update has failed.	FPGA SW checksum is not correct match after FPGA SW update is detected.	None.	None.	None.	None.	Module	Major
24	Unit blocked	Unit is blocked.	Parameter **blocked** of the Module element is set to “TRUE”.	Parameter **blocked** of the Module Element is set to “FALSE”	Blocked unit shuts down all RF emission and turns off power on antenna lines and ALD ports	None.	None.	Module	Critical
25	Reset Requested	Unit detected a transient problem which significantly affects operation that requires a reset as a recovery.	HW implementation specific	None.	None.	None.	Reset.	Module	Critical
26	Power Supply Faulty	Input power to module has fault, unstable or broken.	HW implementation specific	None	None	None	Reset or None	Module	Critical, Major or Minor
27	Power Amplifier faulty	One of power amplifiers in module has fault, unstable or broken	HW implementation specific	None	None	None	Reset or None	Tx-array and/or antenna line	Major

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Fault id	Name	Meaning	Start condition	Cancel condition	NETCONF Server actions on detection	NETCONF Server actions on cancel	System recovery actions	Source	Severity
28	C/U-plane logical Connection faulty	One of logical C/U-plane connection has fault, unstable or broken.	One of C/U-plane processing elements detects the error of C/U-plane connection faulty, (when the O-RU's CU plane monitoring timer expires).	Deactivation or removal of carrier related to all low-level-tx-links being mentioned as alarm source.	None	None	Deactivation or removal of carrier related to all low-level-tx-links being mentioned as alarm source.	low-level-tx-link-and/or low-level-rx-link	Major, Minor or warning
29	Transceiver Fault	Unit has detected a transceiver fault	HW and SW implementation specific.	None.	None	None.	None.	Module	Critical
30	Interface Fault	Unit has detected a fault with one of its interfaces	HW and SW implementation specific.	None.	Unit reset x times for recovery.	None.	None.	Module	Major or Critical
31	Unexpected C/U-plane message content fault	C/U-plane message content was faulty for undetermined reason.	C/U-plane detects unexpected message content.	Carrier affected by this fault was removed	SW implementation specific	None	None	Tx-link and/or Rx link	Major, Minor or warning
32	Triggering failure of antenna calibration	O-RU has previously sent a notification antenna-calibration-required and has not received an antenna-calibration-start RPC	Major - O-RU has not received an RPC trigger for start antenna calibration within 60 seconds after triggering the sending of the antenna-calibration-required notification Critical – After O-RU specific number of repetitions based on O-RU implementation/ Failure of O-RU self-calibration	Arrival of RPC antenna calibration start/Success of O-RU self-calibration	Major-None, Critical-Reset or None	Major-None, Critical-None	Major -Send RPC antenna calibration start/self-calibration Critical- Reset	Module	Major/ Critical
33	Dying Gasp	O-RU is suffering from an unrecoverable condition such as power failure	Critical – O-RU is experiencing a dying gasp event	Re-starting of the O-RU, e.g., after power has been restored	HW implementation specific	None	None	Module	Critical

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Annex B Counter definition

measurement-group	measurement-object	report-info	object-unit	Note
transceiver-stats	RX_POWER	**max and time** **min and time** **first and time** **latest and time** **frequency-table**	PORT_NUMBER	Type decimal64 including 4 fraction-digits for **max**, **min**, **first** and **latest**. A parameter date-and-time is reported for each additionally. Configurable parameters: **function**, **bin-count**, **lower-bound**, **upper-bound** are defined. For more detail see B.1.1 and B.1.2. Type uint32 is used for **frequency-table**.
	RX_POWER_LANE_2			
	RX_POWER_LANE_3			
	RX_POWER_LANE_4			
	TX_POWER			
	TX_POWER_LANE_2			
	TX_POWER_LANE_3			
	TX_POWER_LANE_4			
	TX_BIAS_COUNT			
	TX_BIAS_COUNT_LANE_2			
	TX_BIAS_COUNT_LANE_3			
	TX_BIAS_COUNT_LANE_4			
	VOLTAGE			
	TEMPARATURE			
rx-window-stats	RX_ON_TIME	**count**	RU, TRANSPORT, or EAXC_ID	Type yang: counter64 is used for the count. When **object-unit** is EAXC_ID, TRANSPORT is reported as additional parameter for EAXC_ID.
	RX_EARLY			
	RX_LATE			
	RX_CORRUPT			
	RX_DUPL			
	RX_TOTAL			
	RX_ON_TIME_C			
	RX_EARLY_C			
	RX_LATE_C			
	RX_SEQID_ERR			
	RX_SEQID_ERR_C			
	RX_ERR_DROP			

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

tx-measurement-objects	TX_TOTAL TX_TOTAL_C	**count**	RU, TRANSPORT, or EAXC_ID	Type yang: counter64 is used for the count. When **object-unit** is EAXC_ID, TRANSPORT is reported as additional parameter for EAXC_ID.
epe-stats	POWER TEMPERATURE	**max** **min** **average**	Hardware component type, e.g., O-RAN-RADIO, O-RU-POWER-AMPLIFIER, O-RU-FPGA, power-supply, fan, cpu	Type decimal64 including 4 fraction-digits for **max**, **min**, **average.** Power measured using method described in 5.1.1.19 Power, Energy and Environmental (PEE) measurements of 3GPP TS 28.552 Unit of power: watts (W) Temperature measured using method described in Power measured using method described in 5.1.1.19 Power, Energy and Environmental (PEE) measurements of 3GPP TS 28.552 Unit of temperature: Celsius

Table 8: Counters definition

A parameter: **measurement-interval** is defined per group of **measurement-object**s.

A parameter: **active** is defined per **measurement-object**.

The **object-unit** for the **measurement-object** of rx-window-measurement can be selected per RU, per TRANSPORT, or EAXC_ID. RU is assumed to support one of the **object-unit**s for the rx-window-measurement.

TRANSPORT indicates the **name** of **transport-flow** in o-ran-processing-element YANG.

The type Uint16 is used for EAXC_ID. Measurement result shall contain additional information **name** for its **transport-flow** when EAXC_ID is selected for the **object-unit**.

A feature "GRANULARITY-EAXC-ID-MEASUREMENT" and a feature "GRANULARITY-TRANSPORT-MEASUREMENT" are defined as optional definition in O-RU.

B.1 Transceiver Statistics

The transceiver-measurement includes the performance measurement of transceivers as shown in the following table.

measurement-object	**Description**
RX_POWER	Measured Rx input power in mW for SFP or lane 1 of QSFP
RX_POWER_LANE_2	Measured Rx input power in mW for lane 2 of QSFP

RX_POWER_LANE_3	Measured Rx input power in mW for lane 3 of QSFP
RX_POWER_LANE_4	Measured Rx input power in mW for lane 4 of QSFP
TX_POWER	Measured Tx input power in mW for SFP or lane 1 of QSFP.
TX_POWER_LANE_2	Measured Tx input power in mW for lane 2 of QSFP
TX_POWER_LANE_3	Measured Tx input power in mW for lane 3 of QSFP
TX_POWER_LANE_4	Measured Tx input power in mW for lane 4 of QSFP
TX_BIAS_COUNT	Internally measured Tx Bias Current in mA for SFP or lane 1 of QSFP
TX_BIAS_COUNT_LANE_2	Internally measured Tx Bias Current in mA for lane 2 of QSFP
TX_BIAS_COUNT_LANE_3	Internally measured Tx Bias Current in mA for lane 3 of QSFP
TX_BIAS_COUNT_LANE_4	Internally measured Tx Bias Current in mA for lane 4 of QSFP
VOLTAGE	Internally measured transceiver supply voltage in mV
TEMPARATURE	Internally measured optional laser temperature in degrees Celsius.

Table 9: Transceiver Measurements

B.1.1 Statistics Calculation

When configured by the NETCONF client, the O-RU captures value of monitored parameters. Then the O-RU calculates x = f(s), where f(s) is a function selected for specific statistics instance. The function f(s) can be one of the following:

- f(s) = s
- f(s) = $LOG_{10}(s)$,

where $LOG_{10}(s)$ is logarithm with base 10. To avoid issues with infinity, the O-RU assumes that for s < 10^{-128} value of $LOG_{10}(s)$ is -128.

The value of x = f(s) is applied to **first**, **latest**, **min** and **max** values; related timestamps are also updated; frequency table is updated as described in the following section.

When local measurement interval, which is not same as **transceiver-measurement-interval** of the **measurement-object**, passes the O-RU captures value of a monitored parameter (s). Then the O-RU calculates x = f(s), where f is a function selected for specific parameter. The local measurement interval is up to the O-RU implementation matter and typically around 10 sec – 60 sec at earliest.

The value of x = f(s) is applied to latest value; related timestamp is updated.
The O-RU updates statistics:

- If x < min value then x is applied to min value and related timestamp is updated.
- If x > max value then x is applied to max value and related timestamp is updated.
- Value of x is used to update frequency table as described in section Frequency Table below.

After updates O-RU waits another interval to elapse.

B.1.2 Frequency Table Generation

Let n = **bin-count**, a = **lower-bound**, b = **upper-bound**, x = f(s) where s is value of monitored parameter and f is a function selected for statistics via parameter **function**.

- If n = 0 then frequency table is empty and is not updated.
- If n > 0 there are n bins: h_k where k = 0...n-1. Initial value of each bin is zero (h_k = 0 for k = 0...n-1).
- If x < a then bin h_0 is incremented.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

- If $b \leq x$ and $n > 1$ then bin h_{n-1} is incremented.
- If $a \leq x$ and $x < b$ and $n > 2$ then bin h_k is incremented for k such that
- $k-1 \leq (n-2) * (x-a) / (b-a) < k$.

Note value of bin should saturate at maximum without overflowing (the value is not incremented above $2^{32}-1$). Equivalently:

- For $k = 0$, h_k is a number of values x such that $x < a$.
- For $k = 1 \ldots n-2$, h_k is a number of values x such that
- $a + (b-a) * (k-1) / (n-2) \leq x < a + (b-a) * (k) / (n-2)$.
- For $k = n-1$, h_k is a number of values x such that $b \leq x$.

Example:

function = LOG_{10}, **bin-count** = 14, **lower-bound** = -12, **upper-bound** = 0

- parameter value s = 0, x = f(0) = -128, $-128 < -12 = a \rightarrow h_0$ is incremented
- parameter value s = $1e^{-12}$, x = f($1e^{-12}$) = -12, (14-2)*(-12-(-12))/(0-(-12)) = $12*0/12 < 1 \rightarrow h_1$ is incremented
- parameter value s = $9.99e^{-12}$, x = f($9.99e^{-12}$) = -11.0004, (14-2)*(-11.0004-(-12))/(0-(-12)) = $12*0.9996/12 < 1$ $\rightarrow h_1$ is incremented
- parameter value s = $1e^{-1}$, x = f($1e^{-1}$) = -1, (14-2)*(-1-(-12))/(0-(-12)) = $12*11/12 < 12 \rightarrow h_{12}$ is incremented
- parameter value s = 1, x = f(1) = 0, $0 \geq 0 = b \rightarrow h_{13}$ is incremented

B.2 Rx Window Statistics

The rx-window-measurement includes the performance measurement for the reception window as following table.

measurement-object	Description
RX_ON_TIME	The number of data packet received on time (applies to user data reception window) within the **rx-window-measurement-interval**
RX_EARLY	The number of data packet received too early (applies to user data reception window) within the **rx-window-measurement-interval**
RX_LATE	The number of data packet received too late (applies to user data reception window) within the **rx-window-measurement-interval**
RX_CORRUPT	The number of data packet, which is corrupt or whose header is incorrect, received within the **rx-window-measurement-interval**
RX_DUPL	This counter is deprecated
RX_TOTAL	The total number of received packet (data and control), within the **rx-window-measurement-interval**
RX_ON_TIME_C	The number of control packets, received on time within the **rx-window-measurement-interval**
RX_EARLY_C	The number of control packets, received before the start of reception window within the **rx-window-measurement-interval**
RX_LATE_C	The number of control packets, received after the end of reception window within the **rx-window-measurement-interval**
RX_SEQID_ERR	The number of data packets, received with an erroneous sequence ID within the **rx-window-measurement-interval**

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

RX_SEQID_ERR_C	The number of control packets, received with an erroneous sequence ID within the **rx-window-measurement-interval**
RX_ERR_DROP	The total number of inbound messages which are discarded by the receiving O-RAN entity for any reason within the **rx-window-measurement-interval**

Table 10: Rx Window Measurement

B.3 Tx Statistics

The tx-measurements include the measurement according to the following table.

measurement-object	**Description**
TX_TOTAL	The number of outbound packets (data and control), transmitted within the **tx-measurement-interval**
TX_TOTAL_C	the number of outbound control packets, transmitted within the **tx-measurement-interval** (This counter is required only if RU supports LAA/LBT capabilities**)**

Table 11: Tx Measurement

B.4 Energy, Power and Environmental Statistics

The epe-stats include the performance measurement for energy, power and environmental parameters as shown in the following table. An O-RU will report its supported measurement objects per hardware component class.

measurement-object	**Description**
POWER	Value of measured power consumed by identified hardware component
TEMPERATURE	Value of measured temperature of identified hardware component

Table 12: Energy, Power and Environmental Measurements

Annex C Optional Multi-Vendor Functionality

C.1: Optional Namespace

Some of the YANG models are optional for the O-RU to support. In this version of the management plane specification, the following YANG models are optional to support. If an O-RU/NETCONF server does not return the namespace associated with an optional YANG model, the NETCONF client can infer that the O-RU does not support the optional capability associated with the model.

No	Optional Functionality	Reference	Namespace
1	Antenna Line Device	Chapter 11.3	"urn:o-ran:ald-port:x.y" "urn:o-ran:ald: x.y "
2	External IO Port	Chapter 11.4	"urn:o-ran:external-io:x.y "
3	eCPRI delay measurement	Chapter 4.7	"urn:o-ran:message5:x.y "

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

4	UDP Echo functionality for IP based transport verification	Chapter 4.6	"urn:o-ran:udpecho:x.y "
5	Beamforming	Chapter 12.4	"urn:o-ran:beamforming:x.y "
6	FAN	-	“urn:o-ran:fan:x.y”
7	LAA	Chapter 13	"urn:o-ran:laa:x.y " "urn:o-ran:laa-operations:x.y "
8	Antenna calibration	Chapter 12.5	"urn:o-ran:antcal: x.y "
9	Shared cell (common to FHM and Cascade modes)	Chapter 14	"urn:o-ran:shared-cell:x.y" "urn:o-ran:ethernet-fwd:x.y"
10	Configured subscription transported using VES common header	Chapter 15	"urn:o-ran:ves-sn:1.0"

Table 13: Optional O-RAN Namespace

No	Optional Functionality	Reference	Namespace
1	Notification of Updates to Configuration Datastore	Chapter 6.4	"urn:ietf:params:xml:ns:yang:ietf-netconf-notifications "
2	(Transport agnostic) sSubscriptions to YANG notifications	Chapter 15	"urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications"

Table 14: Optional IETF Namespace

Whereas the above two tables describe those optional YANG modules associated with optional features, there are also scenarios where support of an optional feature means that previously defined mandatory YANG models become optional. Table **15** describes those optional features that when supported result in YANG models becoming optional.

No	Optional Functionality	Reference	Namespace
1	FHM in shared cell	Chapter 14.5.1	"urn:o-ran:module-cap:x.y " "urn:o-ran:uplane-conf:x.y "

Table 15**: Not mandatory O-RAN Namespace for FHM.**

C.2: Optional YANG Features

Some of the O-RAN defined YANG models define optional feature support. The optional multi-vendor features defined in the O-RAN defined YANG models are shown below.

No	Optional Feature	Namespace	Feature name
1	Adaptive O-RU delay profile	"urn:o-ran:delay:x.y "	ADAPTIVE-RU-PROFILE
2	O-RU Energy saving	"urn:o-ran:hardware:x.y "	ENERGYSAVING

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

3	Alias MAC address based C/U transport	"urn:o-ran:interfaces:x.y "	ALIASMAC-BASED-CU-PLANE
4	UDP/IP based C/U Transport	"urn:o-ran:interfaces:x.y "	UDPIP-BASED-CU-PLANE
5	Dynamic Beamforming Configuration	"urn:o-ran:beamforming:x.y "	MODIFY-BF-CONFIG
6	GNSS Support	"urn:o-ran:sync:x.y "	GNSS
7	ALD overcurrent reporting	"urn:o-ran:ald-port:x.y "	OVERCURRENT-SUPPORTED
8	TRANSPORT in rx-window-measurement	"urn:o-ran:performance-management:x.y "	GRANULARITY-TRANSPORT-MEASUREMENT
9	EAXC_ID in rx-window-measurement	"urn:o-ran:performance-management:x.y "	GRANULARITY-EAXC-ID-MEASUREMENT
10	LAA Support	"urn:o-ran: module-cap "	LAA
11	Transport Fragmentation	"urn:o-ran:module-cap:x.y "	TRANSPORT-FRAGMENTATION
12	GNSS Anti Jamming	"urn:o-ran:sync:x.y "	ANTI-JAM
13	Tiliting pre-defined beams	"urn:o-ran:beamforming:x.y"	BEAM-TILT
14	Shared cell support	"urn:o-ran:processing-element:x.y"	SHARED_CELL
15	FHM support, no capability of radio transmission and reception	"urn:o-ran:shared-cell:x.y"	FHM
16	Dynamic Spectrum Sharing	"urn:o-ran: module-cap:x.y"	DSS_LTE_NR
17	EAXC-ID Grouping	"urn:o-ran: module-cap:x.y"	EAXC-ID-GROUP-SUPPORTED
18	Configurable FS offset	"urn:o-ran:compression-factors:x.y "	CONFIGURABLE-FS-OFFSET
19	eAxC specific gain correction	"urn:o-ran:uplane-conf:x.y "	EAXC-GAIN-CORRECTION
20	TX gain reference level control	"urn:o-ran:uplane-conf:x.y "	TX-REFERENCE-LEVEL
21	Static configuration of PRACH pattern	"urn:o-ran: module-cap:x.y"	PRACH-STATIC-CONFIGURATION-SUPPORTED
22	Static configuration of SRS pattern	"urn:o-ran: module-cap:x.y"	SRS-STATIC-CONFIGURATION-SUPPORTED
23	Configurable TDD pattern	"urn:o-ran: module-cap:x.y"	CONFIGURABLE-TDD-PATTERN-SUPPORTED
24	Optimizations for non-persistent M-Plane	"urn:o-ran:feat:1.0"	NON-PERSISTENT-MPLANE

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

25	U-plane transmission window control configuration over M-plane	"urn:o-ran:feat:1.0"	STATIC-TRANSMISSION-WINDOW-CONTROL
26	U-plane transmission window control configuration over C-plane	"urn:o-ran:feat:1.0"	DYNAMIC-TRANSMISSION-WINDOW-CONTROL
27	Transmission of UL U-plane messages distributed uniformly over transmission window	"urn:o-ran:feat:1.0"	UNIFORMLY-DISTRIBUTED-TRANSMISSION
28	Ordered transmission	"urn:o-ran:feat:1.0"	ORDERED-TRANSMISSION
29	Independent U-plane transmission window per endpoint	"urn:o-ran:feat:1.0"	INDEPENDENT-TRANSMISSION-WINDOW-CONTROL

Table 16: Optional O-RAN WG4 defined feature support

Some of the O-RAN defined YANG models augment existing YANG models which have optional features defined. The optional features defined in these “common” models are shown in the table below.

No	Optional Feature	Namespace	Feature name
1	RFC 6933: Entity MIB	"urn:ietf:params:xml:ns:yang:ietf-hardware"	entity-mib
2	RFC 4268: Entity State MIB	"urn:ietf:params:xml:ns:yang:ietf-hardware"	hardware-state
3	RFC 3433: Entity Sensor Management Information Base	"urn:ietf:params:xml:ns:yang:ietf-hardware"	hardware-sensor
4	O-RU allows user-controlled interfaces to be named arbitrarily	"urn:ietf:params:xml:ns:yang:ietf-interfaces"	arbitrary-names
5	O-RU supports pre-provisioning of interface configuration, i.e., it is possible to configure an interface whose physical interface hardware is not present on the device	"urn:ietf:params:xml:ns:yang:ietf-interfaces"	pre-provisioning
6	RFC 2863: The Interfaces Group MIB	"urn:ietf:params:xml:ns:yang:ietf-interfaces"	if-mib
7	O-RU supports configuring non-contiguous subnet masks	"urn:ietf:params:xml:ns:yang:ietf-ip"	ipv4-non-contiguous-netmasks
8	O-RU supports privacy extensions for stateless address autoconfiguration in IPv6	"urn:ietf:params:xml:ns:yang:ietf-ip"	ipv6-privacy-autoconf
9	O-RU supports configured YANG Notifications	"urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications"	configured
10	O-RU supports JSON encoding of subscriptions to YANG notifications	"urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications"	encode-json

Table 17: Optional feature support in common models

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

C.3: Optional Capabilities Exposed Using O-RAN YANG Models

In addition to optional namespaces and optional features within supported namespaces, certain O-RAN defined YANG models are used to be able to expose support for certain optional capabilities by the O-RU.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

No	Optional Feature	Namespace	Leaf
1	Type of synchronization source supported by O-RU	"urn:o-ran:sync:x.y "	/ sync/:sync-status/supported-reference-types
2	O-RU supports extended Category A operation – more than 8 spatial streams	"urn:o-ran:module-cap:x.y "	/module-capability/ru-capabilities/ ru-supported-category and /module-capability/ru-capabilities/ number-of-spatial-streams
3	O-RU supports Category B operation – precoding in the O-RU	"urn:o-ran:module-cap:x.y "	/module-capability/ru-capabilities/ ru-supported-category
4	O-RU supports the capability to apply the modified beamforming configuration by using rpc activate-beamforming-config without deletion of tx-array-carriers and rx-array-carriers	"urn:o-ran:beamforming:x.y "	/beamforming-config/operational-properties/update-bf-non-delete
5	O-RU supports the capability to store the modified beamforming configuration file in the reset persistent memory	"urn:o-ran:beamforming:x.y "	/beamforming-config/operational-properties/persistent-bf-files
6	Optional VLAN optimized searching	"urn:o-ran:mplane-interfaces:x.y "	/mplane-info/searchable-mplane-access-vlans-info
7	Configurable CoS marking for C, U and M-Plane	"urn:o-ran:interfaces:x.y "	augmented /if:interfaces/if:interface: with u-plane-marking c-plane-marking and m-plane-marking
8	Configurable DSCP marking for C, U and M-Plane	"urn:o-ran:interfaces:x.y "	augmented /if:interfaces/if:interface: with u-plane-marking c-plane-marking and m-plane-marking
9	Ethernet Frame MTU	"urn:o-ran:interfaces:x.y"	augmented /if:interfaces/if:interface: with l2-mtu
10	VLAN Tagging	"urn:o-ran:interfaces:x.y"	augmented /if:interfaces/if:interface: with vlan-tagging
11	IEEE 1914.3 header support	"urn:o-ran:operations:x.y "	/operational-info/declarations/supported-header-mechanism/protocol
12	eCPRI Concatenation support	"urn:o-ran:operations:x.y "	/operational-info/declarations/supported-header-mechanism/ecpri-concatenation-support
13	O-RU local management of the LAA contention window	"urn:o-ran:module-cap:x.y "	/module-capability/band-capabilities/sub-band-info/self-configure
14	O-RU supports LAA ending in Downlink Pilot Time Slot (DwPTS)	"urn:o-ran:laa:x.y"	/laa-config/ laa-ending-dwpts-supported
15	O-RU supports configurable timer for C/U plane monitoring	"urn:o-ran: supervision:x.y"	/supervision/cu-plane-monitoring

Table 18: Optional capabilities in O-RAN defined YANG models

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Annex D YANG Module Graphical Representation

The different released version of the set of YANG modules for the O-RU can be downloaded from O-RAN's website http://www.o-ran.org/specifications/ . The YANG models are available in a zip file, whose name is used to represent the version of the YANG model and follows the numerical format defined in subsection 1.1 with the periods replaced with "-", i.e., YANG models for release 1.0.0 of the M-Plane specification are available in the file 1-0-0.zip. This zip file includes all published revisions of the YANG models supporting a particular release of the M-Plane specification.

This Annex provides a set of "tree-views" of the modules to provide a simplified graphical representation of the data models. These trees have been automatically generated using the pyang YANG validator tool. If there are any inconsistencies between the tree representation in this Annex and the yang models, the yang models shall take precedence.

D.1 System Folder

D.1.1 o-ran-supervision.yang Module

The format for the supervision module is provided below.

```
module: o-ran-supervision
  +--rw supervision
  +--rw cu-plane-monitoring!
     |  +--rw configured-cu-monitoring-interval?   uint8
     +--rw event-collector-monitoring {or-feat:NON-PERSISTENT-MPLANE}?
        +--rw heartbeat-interval?       uint8
        +--rw heartbeat-recipient-id*   event-collector-id
  rpcs:
    +---x supervision-watchdog-reset
       +---w input
       |  +---w supervision-notification-interval?   uint16
       |  +---w guard-timer-overhead?                uint16
       +--ro output
          +--ro next-update-at?   yang:date-and-time
          +--ro error-message?    string

  notifications:
    +---n supervision-notification
```

D.1.2 o-ran-usermgmt.yang Module

The format for the user management module is provided below.

```
module: o-ran-usermgmt
  +--rw users
     +--rw user* [name]
        +--rw name            nacm:user-name-type
        +--rw account-type?   enumeration
        +--rw password?       password-type
        +--rw enabled?        boolean

  rpcs:
    +---x chg-password
       +---w input
       |  +---w currentPassword       password-type
       |  +---w newPassword           password-type
       |  +---w newPasswordConfirm    password-type
       +--ro output
          +--ro status            enumeration
          +--ro status-message?   string
```

D.1.3 o-ran-hardware.yang Module

The format for the hardware module is provided below.

```
module: o-ran-hardware
  augment /hw:hardware/hw:component:
    +--ro label-content
    |  +--ro model-name?      boolean
    |  +--ro serial-number?   boolean
```

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

```
    +--ro product-code              string
    +--rw energy-saving-enabled?   boolean {ENERGYSAVING}?
    +--ro dying-gasp-support?      Boolean
    +--rw last-service-date?       yang:date-and-time {or-feat:NON-PERSISTENT-MPLANE}?
  augment /hw:hardware/hw:component:
    +--rw o-ran-name    -> /hw:hardware/component/name
  augment /hw:hardware/hw:component/hw:state:
    +--ro power-state?          energysaving-state {ENERGYSAVING}?
    +--ro availability-state?   availability-type
  augment /hw:hardware-state-oper-enabled:
    +--ro availability-state?   -> /hw:hardware/component/state/o-ran-hw:availability-state
  augment /hw:hardware-state-oper-disabled:
    +--ro availability-state?   -> /hw:hardware/component/state/o-ran-hw:availability-state
```

D.1.4 o-ran-fan.yang Module

The format for the fan module is provided below.

```
module: o-ran-fan
  +--ro fan-tray
     +--ro fan-state* [name]
        +--ro name                      string
        +--ro fan-location?             uint8
        +--ro present-and-operating     boolean
        +--ro vendor-code?              uint8
        +--ro fan-speed?                percent
        +--ro target-speed?             uint16
```

D.1.5 o-ran-fm.yang Module

The format for the fault management module is provided below.

```
module: o-ran-fm
  +--ro active-alarm-list
     +--ro active-alarms* []
        +--ro fault-id             uint16
        +--ro fault-source         string
        +--ro affected-objects* []
        |  +--ro name    string
        +--ro fault-severity       enumeration
        +--ro is-cleared           boolean
        +--ro fault-text?          string
        +--ro event-time           yang:date-and-time

  notifications:
    +---n alarm-notif
       +--ro fault-id             uint16
       +--ro fault-source         string
       +--ro affected-objects* []
       |  +--ro name    string
       +--ro fault-severity       enumeration
       +--ro is-cleared           boolean
       +--ro fault-text?          string
       +--ro event-time           yang:date-and-time
```

D.1.6 o-ran-ves-subscribed-notifications.yang Module

The format for the ves subscribed notifications module is provided below.

```
module: o-ran-ves-subscribed-notifications
  augment /sn:subscriptions/sn:subscription/sn:receivers/sn:receiver:
    +--rw notification-recipient    inet:ur

```

D.2 Operations Folder

D.2.1 o-ran-operations.yang Module

The format for the operations module is provided below.

```
module: o-ran-operations
```

```
module: o-ran-operations
  +--rw operational-info
     +--ro declarations
     |  +--ro ru-instance-id?                string
     |  +--ro supported-mplane-version?      version
     |  +--ro supported-cusplane-version?    version
     |  +--ro supported-header-mechanism* [protocol]
     |  |  +--ro protocol                       enumeration
     |  |  +--ro ecpri-concatenation-support?   boolean
     |  |  +--ro protocol-version?              version
     |  +--ro supported-common-event-header-version?       version {or-feat:NON-PERSISTENT-MPLANE}?
     |  +--ro supported-ves-event-listener-version?        version {or-feat:NON-PERSISTENT-MPLANE}?
     |  +--ro supported-pnf-registration-fields-version?   version {or-feat:NON-PERSISTENT-MPLANE}?
     +--ro operational-state
     |  +--ro restart-cause?      enumeration
     |  +--ro restart-datetime?   yang:date-and-time
     +--rw clock
     |  +--rw timezone-name?         timezone-name
     |  +--rw timezone-utc-offset?   int16
     +--rw re-call-home-no-ssh-timer?   uint16

  rpcs:
    +---x reset
```

D.2.2 o-ran-file-management.yang Module

The format for the file management module is provided below

```
module: o-ran-file-management

  rpcs:
    +---x file-upload
    |  +---w input
    |  |  +---w local-logical-file-path    string
    |  |  +---w remote-file-path           string
    |  |  +---w (credentials)?
    |  |     +--:(password)
    |  |     |  +---w password!
    |  |     |  |  +---w password    string
    |  |     |  +---w server
    |  |     |     +---w keys* [algorithm]
    |  |     +--:(certificate)
    |  |        +---w certificate!
    |  +--ro output
    |     +--ro status?          enumeration
    |     +--ro reject-reason?   string
    +---x retrieve-file-list
    |  +---w input
    |  |  +---w logical-path        string
    |  |  +---w file-name-filter?   string
    |  +--ro output
    |     +--ro status?          enumeration
    |     +--ro reject-reason?   string
    |     +--ro file-list*       string
    +---x file-download
       +---w input
       |  +---w local-logical-file-path    string
       |  +---w remote-file-path           string
       |  +---w (credentials)?
       |     +--:(password)
       |     |  +---w password!
       |     |  |  +---w password    string
       |     |  +---w server
       |     |     +---w keys* [algorithm]
       |     +--:(certificate)
       |        +---w certificate!
       +--ro output
          +--ro status?          enumeration
          +--ro reject-reason?   string

  notifications:
    +---n file-upload-notification
    |  +--ro local-logical-file-path    string
    |  +--ro remote-file-path           string
    |  +--ro status?                    enumeration
```

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

```
   |  +--ro reject-reason?            string
   +---n file-download-event
      +--ro local-logical-file-path    string
      +--ro remote-file-path           string
      +--ro status?                    enumeration
      +--ro reject-reason?             string
```

D.2.3 o-ran-software-management.yang Module

The format for the software management module is provided below

```
module: o-ran-software-management
  +--ro software-inventory
     +--ro software-slot* [name]
        +--ro name              string
        +--ro status            enumeration
        +--ro active?           boolean
        +--ro running?          boolean
        +--ro access?           enumeration
        +--ro product-code?     -> /hw:hardware/component/o-ran-hw:product-code
        +--ro vendor-code?      string
        +--ro build-id?         string
        +--ro build-name?       string
        +--ro build-version?    string
        +--ro files* [name]
           +--ro name           string
           +--ro version?       string
           +--ro local-path     string
           +--ro integrity?     enumeration

  rpcs:
    +---x software-download
    |  +---w input
    |  |  +---w remote-file-path     inet:uri
    |  |  +---w (credentials)?
    |  |     +--:(password)
    |  |     |  +---w password!
    |  |     |  |  +---w password    string
    |  |     |  +---w server
    |  |     |     +---w keys* [algorithm]
    |  |     +--:(certificate)
    |  |        +---w certificate!
    |  +--ro output
    |     +--ro status                  enumeration
    |     +--ro error-message?          string
    |     +--ro notification-timeout?   int32
    +---x software-install
    |  +---w input
    |  |  +---w slot-name     -> /software-inventory/software-slot/name
    |  |  +---w file-names*   string
    |  +--ro output
    |     +--ro status           enumeration
    |     +--ro error-message?   string
    +---x software-activate
       +---w input
       |  +---w slot-name    -> /software-inventory/software-slot/name
       +--ro output
          +--ro status                  enumeration
          +--ro error-message?          string
          +--ro notification-timeout?   int32

  notifications:
    +---n download-event
    |  +--ro file-name        string
    |  +--ro status?          enumeration
    |  +--ro error-message?   string
    +---n install-event
    |  +--ro slot-name?       -> /software-inventory/software-slot/name
    |  +--ro status?          enumeration
    |  +--ro error-message?   string
    +---n activation-event
       +--ro slot-name?       -> /software-inventory/software-slot/name
       +--ro status?          enumeration
       +--ro return-code?     uint8
       +--ro error-message?   string
```

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

D.2.4 o-ran-lbm.yang Module

The format for the (Ethernet) loobback module is provided below

```
module: o-ran-lbm
  +--rw md-data-definitions
     +--rw maintenance-domain* [id]
        +--rw id                        string
        +--rw name?                     string
        +--rw md-level?                 md-level-type
        +--rw maintenance-association* [id]
           +--rw id                string
           +--rw name?             string
           +--rw component-list* [component-id]
              +--rw component-id                         uint32
              +--rw name?                                string
              +--rw vid*                                 -> /if:interfaces/interface/o-ran-int:vlan-id
              +--rw remote-meps*                         mep-id-type
              +--rw maintenance-association-end-point* [mep-identifier]
                 +--rw mep-identifier          mep-id-type
                 +--rw interface               -> /if:interfaces/interface/name
                 +--rw primary-vid             -> /if:interfaces/interface/o-ran-int:vlan-id
                 +--rw administrative-state    boolean
                 +--ro mac-address?            -> /if:interfaces/interface/o-ran-int:mac-address
                 +--ro loopback
                    +--ro replies-transmitted    yang:counter32
```

D.2.5 o-ran-udp-echo.yang Module

The format for the udp echo module is provided below

```
module: o-ran-udp-echo
  +--rw udp-echo {o-ran-int:UDPIP-BASED-CU-PLANE}?     +--rw enable-udp-echo?            boolean
     +--rw dscp-config?                  enumeration
     +--ro echo-replies-transmitted?   uint32
```

D.2.6 o-ran-ecpri-delay.yang Module

The format for the ecpri delay management module is provided below

```
module: o-ran-ecpri-delay
  +--rw ecpri-delay-message
     +--ro ru-compensation
     |  +--ro tcv2?   uint32
     |  +--ro tcv1?   uint32
     +--rw enable-message5?     boolean
     +--rw message5-sessions
        +--rw session-parameters* [session-id]
           +--rw session-id                 uint32
           +--rw processing-element-name?   -> /element:processing-elements/ru-elements/name
           +--ro flow-state
              +--ro responses-transmitted?   uint32
              +--ro requests-transmitted?    uint32
              +--ro followups-transmitted?   uint32
```

D.2.7 o-ran-performance-management.yang Module

The format for the performance management module is provided below

```
module: o-ran-performance-management
  +--rw performance-measurement-objects
     +--ro measurement-capabilitites
     |  +--ro transceiver-objects* [measurement-object]
     |  |  +--ro measurement-object    -> /performance-measurement-objects/transceiver-measurement-
objects/measurement-object
     |  +--ro rx-window-objects* [measurement-object]
     |  |  +--ro measurement-object    -> /performance-measurement-objects/rx-window-measurement-
objects/measurement-object
     |  +--ro tx-stats-objects* [measurement-object]
     |  |  +--ro measurement-object    -> /performance-measurement-objects/tx-measurement-
objects/measurement-object
     |  +--ro epe-stats-objects* [measurement-object]
```

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

```
     |     +--ro measurement-object    -> /performance-measurement-objects/epe-measurement-
objects/measurement-object
     |     +--ro component-class*      identityref
     +--rw enable-SFTP-upload?                 boolean
     +--rw enable-random-file-upload?          boolean
     +--rw remote-SFTP-uploads* [remote-SFTP-upload-path]
     |  +--rw remote-SFTP-upload-path    inet:uri
     +--rw transceiver-measurement-interval?   uint16
     +--rw rx-window-measurement-interval?     uint16
     +--rw epe-measurement-interval?           uint16
     +--rw tx-measurement-interval?            uint16
     +--rw notification-interval?              uint16
     +--rw file-upload-interval?               uint16
     +--ro max-bin-count                       uint16
     +--rw transceiver-measurement-objects* [measurement-object]
     |  +--rw measurement-object                enumeration
     |  +--rw active?                           boolean
     |  +--rw report-info*                      enumeration
     |  +--rw object-unit                       enumeration
     |  +--rw function?                         enumeration
     |  +--rw bin-count?                        uint16
     |  +--rw lower-bound?                      decimal64
     |  +--rw upper-bound?                      decimal64
     |  +--ro transceiver-measurement-result* [object-unit-id]
     |     +--ro object-unit-id    -> /if:interfaces/interface/o-ran-int:port-reference/port-number
     |     +--ro min
     |     |  +--ro value?   decimal64
     |     |  +--ro time?    yang-types:date-and-time
     |     +--ro max
     |     |  +--ro value?   decimal64
     |     |  +--ro time?    yang-types:date-and-time
     |     +--ro first
     |     |  +--ro value?   decimal64
     |     |  +--ro time?    yang-types:date-and-time
     |     +--ro latest
     |     |  +--ro value?   decimal64
     |     |  +--ro time?    yang-types:date-and-time
     |     +--ro frequeny-table*   uint32
     +--rw rx-window-measurement-objects* [measurement-object]
     |  +--rw measurement-object            enumeration
     |  +--rw active?                       boolean
     |  +--rw object-unit?                  enumeration
     |  +--rw report-info?                  enumeration
     |  +--ro (object-unit-id)?
     |     +--:(RU)
     |     |  +--ro name?                   -> /hw:hardware/component/name
     |     |  +--ro count                   uint64
     |     +--:(TRANSPORT)
     |     |  +--ro tr-measured-result* []
     |     |     +--ro name?    -> /o-ran-elements:processing-elements/ru-elements/name
     |     |     +--ro count    uint64
     |     +--:(EAXC_ID)
     |        +--ro eaxc-measured-result* []
     |           +--ro eaxc-id?          uint16
     |           +--ro count             uint64
     |           +--ro transport-name?   -> /o-ran-elements:processing-elements/ru-elements/name
     +--rw tx-measurement-objects* [measurement-object]
     |  +--rw measurement-object            enumeration
     |  +--rw active?                       boolean
     |  +--rw object-unit?                  enumeration
     |  +--rw report-info?                  enumeration
     |  +--ro (object-unit-id)?
     |     +--:(RU)
     |     |  +--ro name?                   -> /hw:hardware/component/name
     |     |  +--ro count                   uint64
     |     +--:(TRANSPORT)
     |     |  +--ro tr-measured-result* []
     |     |     +--ro name?    -> /o-ran-elements:processing-elements/ru-elements/name
     |     |     +--ro count    uint64
     |     +--:(EAXC_ID)
     |        +--ro eaxc-measured-result* []
     |           +--ro eaxc-id?          uint16
     |           +--ro count             uint64
     |           +--ro transport-name?   -> /o-ran-elements:processing-elements/ru-elements/name
     +--rw epe-measurement-objects* [measurement-object]
```

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

```
        +--rw measurement-object        enumeration
        +--rw active?                   boolean
        +--rw report-info*              enumeration
        +--ro epe-measurement-result* [object-unit-id]
           +--ro object-unit-id    -> /hw:hardware/component/class
           +--ro min?              decimal64
           +--ro max?              decimal64
           +--ro average?          decimal64

  notifications:
    +---n measurement-result-stats
       +--ro transceiver-stats* [measurement-object]
       |  +--ro measurement-object                -> /performance-measurement-objects/transceiver-
measurement-objects/measurement-object
       |  +--ro start-time?                       yang-types:date-and-time
       |  +--ro end-time?                         yang-types:date-and-time
       |  +--ro transceiver-measurement-result* [object-unit-id]
       |     +--ro object-unit-id    -> /if:interfaces/interface/o-ran-int:port-reference/port-number
       |     +--ro min
       |     |  +--ro value?   decimal64
       |     |  +--ro time?    yang-types:date-and-time
       |     +--ro max
       |     |  +--ro value?   decimal64
       |     |  +--ro time?    yang-types:date-and-time
       |     +--ro first
       |     |  +--ro value?   decimal64
       |     |  +--ro time?    yang-types:date-and-time
       |     +--ro latest
       |     |  +--ro value?   decimal64
       |     |  +--ro time?    yang-types:date-and-time
       |     +--ro frequeny-table*   uint32
       +--ro rx-window-stats* [measurement-object]
       |  +--ro measurement-object            -> /performance-measurement-objects/rx-window-measurement-
objects/measurement-object
       |  +--ro start-time?                   yang-types:date-and-time
       |  +--ro end-time?                     yang-types:date-and-time
       |  +--ro (object-unit-id)?
       |     +--:(RU)
       |     |  +--ro name?                   -> /hw:hardware/component/name
       |     |  +--ro count                   uint64
       |     +--:(TRANSPORT)
       |     |  +--ro tr-measured-result* []
       |     |     +--ro name?    -> /o-ran-elements:processing-elements/ru-elements/name
       |     |     +--ro count    uint64
       |     +--:(EAXC_ID)
       |        +--ro eaxc-measured-result* []
       |           +--ro eaxc-id?          uint16
       |           +--ro count             uint64
       |           +--ro transport-name?   -> /o-ran-elements:processing-elements/ru-elements/name
       +--ro tx-stats* [measurement-object]
       |  +--ro measurement-object            -> /performance-measurement-objects/tx-measurement-
objects/measurement-object
       |  +--ro start-time?                   yang-types:date-and-time
       |  +--ro end-time?                     yang-types:date-and-time
       |  +--ro (object-unit-id)?
       |     +--:(RU)
       |     |  +--ro name?                   -> /hw:hardware/component/name
       |     |  +--ro count                   uint64
       |     +--:(TRANSPORT)
       |     |  +--ro tr-measured-result* []
       |     |     +--ro name?    -> /o-ran-elements:processing-elements/ru-elements/name
       |     |     +--ro count    uint64
       |     +--:(EAXC_ID)
       |        +--ro eaxc-measured-result* []
       |           +--ro eaxc-id?          uint16
       |           +--ro count             uint64
       |           +--ro transport-name?   -> /o-ran-elements:processing-elements/ru-elements/name
       x--ro epe-stats
       |  +--ro start-time?               yang-types:date-and-time
       |  +--ro end-time?                 yang-types:date-and-time
       |  +--ro epe-measurement-result* [object-unit-id]
       |     +--ro object-unit-id    -> /hw:hardware/component/class
       |     +--ro min?              decimal64
       |     +--ro max?              decimal64
       |     +--ro average?          decimal64
```

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

```
      +--ro epe-statistics* [measurement-object]
         +--ro measurement-object       -> /performance-measurement-objects/epe-measurement-
objects/measurement-object
         +--ro start-time?              yang-types:date-and-time
         +--ro end-time?                yang-types:date-and-time
         +--ro epe-measurement-result* [object-unit-id]
            +--ro object-unit-id    -> /hw:hardware/component/class
            +--ro min?              decimal64
            +--ro max?              decimal64
            +--ro average?          decimal64

```

D.2.8 o-ran-uplane-conf.yang Module

The format for the userplane configuration module is provided below

```
module: o-ran-uplane-conf
  +--rw user-plane-configuration
     +--rw low-level-tx-links* [name]
     |  +--rw name                    string
     |  +--rw processing-element      -> /o-ran-pe:processing-elements/ru-elements/name
     |  +--rw tx-array-carrier        -> /user-plane-configuration/tx-array-carriers/name
     |  +--rw low-level-tx-endpoint   -> /user-plane-configuration/low-level-tx-endpoints/name
     +--rw low-level-rx-links* [name]
     |  +--rw name                        string
     |  +--rw processing-element          -> /o-ran-pe:processing-elements/ru-elements/name
     |  +--rw rx-array-carrier            -> /user-plane-configuration/rx-array-carriers/name
     |  +--rw low-level-rx-endpoint       -> /user-plane-configuration/low-level-rx-endpoints/name
     |  +--rw user-plane-uplink-marking?  -> /o-ran-pe:processing-elements/enhanced-uplane-mapping/uplane-
mapping/up-marking-name
     +--ro endpoint-types* [id]
     |  +--ro id                                                           uint16
     |  +--ro supported-section-types* [section-type]
     |  |  +--ro section-type                   uint8
     |  |  +--ro supported-section-extensions*   uint8
     |  +--ro supported-frame-structures*                                  uint8
     |  +--ro managed-delay-support?                                       enumeration
     |  +--ro multiple-numerology-supported?                               boolean
     |  +--ro max-numerology-change-duration?                              uint16
     |  +--ro max-control-sections-per-data-section?                       uint8
     |  +--ro max-sections-per-symbol?                                     uint16
     |  +--ro max-sections-per-slot?                                       uint16
     |  +--ro max-highest-priority-sections-per-slot?                      uint16
     |  x--ro max-remasks-per-section-id?                                  uint8
     |  +--ro max-uplane-section-header-per-symbol?                        uint16
     |  +--ro max-uplane-section-header-per-slot?                          uint16
     |  +--ro max-beams-per-symbol?                                        uint16
     |  +--ro max-beams-per-slot?                                          uint16
     |  +--ro max-beam-updates-per-slot?                                   uint16
     |  +--ro max-beam-updates-per-symbol?                                 uint16
     |  +--ro max-prb-per-symbol?                                          uint16
     |  +--ro prb-capacity-allocation-granularity*                         uint16
     |  +--ro max-numerologies-per-symbol?                                 uint16
     |  +--ro static-transmission-window-control-supported?                boolean {feat:STATIC-
TRANSMISSION-WINDOW-CONTROL}?
     |  +--ro uniformly-distributed-transmission-supported?                boolean {feat:STATIC-
TRANSMISSION-WINDOW-CONTROL and feat:UNIFORMLY-DISTRIBUTED-TRANSMISSION}?
     |  +--ro ordered-transmission-supported?                              boolean {feat:ORDERED-
TRANSMISSION}?
     |  +--ro dynamic-transmission-window-control-supported?               boolean {feat:DYNAMIC-
TRANSMISSION-WINDOW-CONTROL}?
     |  +--ro dynamic-transmission-window-control-per-section-supported?   boolean {feat:DYNAMIC-
TRANSMISSION-WINDOW-CONTROL}?
     |  +--ro dynamic-uniformly-distributed-transmission-supported?        boolean {feat:DYNAMIC-
TRANSMISSION-WINDOW-CONTROL and feat:UNIFORMLY-DISTRIBUTED-TRANSMISSION}?
     |  +--ro dynamic-uniformly-distributed-transmission-per-section-supported?   boolean
     |  +--ro transmission-buffering-capacity* [] {feat:STATIC-TRANSMISSION-WINDOW-CONTROL or feat:DYNAMIC-
TRANSMISSION-WINDOW-CONTROL}?
     |     +--ro iq-bitwidth?                        uint8
     |     +--ro compression-type                    enumeration
     |     x--ro bitwidth?                           uint8
     |     +--ro (compression-format)?
     |     |  +--:(no-compresison)
     |     |  +--:(block-floating-point)
     |     |  |  +--ro exponent?                     uint8
```

```
        |      |  +--:(block-floating-point-selective-re-sending)
        |      |  |  +--ro sres-exponent?                        uint8
        |      |  +--:(block-scaling)
        |      |  |  +--ro block-scalar?                         uint8
        |      |  +--:(u-law)
        |      |  |  +--ro comp-bit-width?                       uint8
        |      |  |  +--ro comp-shift?                           uint8
        |      |  +--:(beam-space-compression)
        |      |  |  +--ro active-beam-space-coeficient-mask*    uint8
        |      |  |  +--ro block-scaler?                         uint8
        |      |  +--:(modulation-compression)
        |      |  |  +--ro csf?                                  uint8
        |      |  |  +--ro mod-comp-scaler?                      uint16
        |      |  +--:(modulation-compression-selective-re-sending)
        |      |     +--ro sres-csf?                             uint8
        |      |     +--ro sres-mod-comp-scaler?                 uint16
        |     +--ro max-buffered-prbs?                           uint32
        |     +--ro max-buffered-symbols?                        uint32
        +--rw transmission-window-schedules* [id] {feat:STATIC-TRANSMISSION-WINDOW-CONTROL}?
        |  +--rw id          uint16
        |  +--rw schedule* [symbol]
        |     +--rw symbol    uint16
        |     +--rw offset?   uint16
        +--ro endpoint-capacity-sharing-groups* [id]
        |  +--ro id                                          uint16
        |  +--ro max-control-sections-per-data-section?     uint8
        |  +--ro max-sections-per-symbol?                    uint16
        |  +--ro max-sections-per-slot?                      uint16
        |  +--ro max-highest-priority-sections-per-slot?    uint16
        |  x--ro max-remasks-per-section-id?                 uint8
        |  +--ro max-uplane-section-header-per-symbol?      uint16
        |  +--ro max-uplane-section-header-per-slot?        uint16
        |  +--ro max-beams-per-symbol?                       uint16
        |  +--ro max-beams-per-slot?                         uint16
        |  +--ro max-beam-updates-per-slot?                  uint16
        |  +--ro max-beam-updates-per-symbol?                uint16
        |  +--ro max-prb-per-symbol?                         uint16
        |  +--ro max-numerologies-per-symbol?                uint16
        |  +--ro max-endpoints?                              uint16
        |  +--ro max-managed-delay-endpoints?                uint16
        |  +--ro max-non-managed-delay-endpoints?            uint16
        |  +--ro transmission-buffering-capacity* [] {feat:STATIC-TRANSMISSION-WINDOW-CONTROL or feat:DYNAMIC-
TRANSMISSION-WINDOW-CONTROL}?
        |     +--ro iq-bitwidth?                                 uint8
        |     +--ro compression-type                             enumeration
        |     x--ro bitwidth?                                    uint8
        |     +--ro (compression-format)?
        |      |  +--:(no-compresison)
        |      |  +--:(block-floating-point)
        |      |  |  +--ro exponent?                             uint8
        |      |  +--:(block-floating-point-selective-re-sending)
        |      |  |  +--ro sres-exponent?                        uint8
        |      |  +--:(block-scaling)
        |      |  |  +--ro block-scalar?                         uint8
        |      |  +--:(u-law)
        |      |  |  +--ro comp-bit-width?                       uint8
        |      |  |  +--ro comp-shift?                           uint8
        |      |  +--:(beam-space-compression)
        |      |  |  +--ro active-beam-space-coeficient-mask*    uint8
        |      |  |  +--ro block-scaler?                         uint8
        |      |  +--:(modulation-compression)
        |      |  |  +--ro csf?                                  uint8
        |      |  |  +--ro mod-comp-scaler?                      uint16
        |      |  +--:(modulation-compression-selective-re-sending)
        |      |     +--ro sres-csf?                             uint8
        |      |     +--ro sres-mod-comp-scaler?                 uint16
        |     +--ro max-buffered-prbs?                           uint32
        |     +--ro max-buffered-symbols?                        uint32
        +--ro endpoint-prach-group* [id]
        |  +--ro id                                    uint16
        |  +--ro supported-prach-preamble-formats*    prach-preamble-format
        +--ro supported-compression-method-sets* [id]
        |  +--ro id                                uint16
        |  +--ro compression-method-supported* []
        |     +--ro iq-bitwidth?                                 uint8
```

```
     |     +--ro (compression-format)?
     |     |  +--:(no-compresison)
     |     |  +--:(block-floating-point)
     |     |  |  +--ro exponent?                          uint8
     |     |  +--:(block-floating-point-selective-re-sending)
     |     |  |  +--ro sres-exponent?                     uint8
     |     |  +--:(block-scaling)
     |     |  |  +--ro block-scalar?                      uint8
     |     |  +--:(u-law)
     |     |  |  +--ro comp-bit-width?                    uint8
     |     |  |  +--ro comp-shift?                        uint8
     |     |  +--:(beam-space-compression)
     |     |  |  +--ro active-beam-space-coeficient-mask*   uint8
     |     |  |  +--ro block-scaler?                      uint8
     |     |  +--:(modulation-compression)
     |     |  |  +--ro csf?                               uint8
     |     |  |  +--ro mod-comp-scaler?                   uint16
     |     |  +--:(modulation-compression-selective-re-sending)
     |     |     +--ro sres-csf?                          uint8
     |     |     +--ro sres-mod-comp-scaler?              uint16
     |     +--ro fs-offset*                               uint8 {cf:CONFIGURABLE-FS-OFFSET}?
     +--ro static-low-level-tx-endpoints* [name]
     |  +--ro name                                string
     |  +--ro restricted-interfaces*              -> /if:interfaces/interface/name
     |  +--ro array                               -> /user-plane-configuration/tx-arrays/name
     |  +--ro endpoint-type?                      -> ../../endpoint-types/id
     |  +--ro capacity-sharing-groups*            -> ../../endpoint-capacity-sharing-groups/id
     |  +--ro supported-reference-level* [id] {TX-REFERENCE-LEVEL}?
     |  |  +--ro id      uint16
     |  |  +--ro min     decimal64
     |  |  +--ro max     decimal64
     |  +--ro compression
     |  |  +--ro dynamic-compression-supported?           boolean
     |  |  +--ro realtime-variable-bit-width-supported?   boolean
     |  |  +--ro supported-compression-set-id?            -> ../../../supported-compression-method-sets/id
     |  +--ro configurable-tdd-pattern-supported?   boolean {mcap:CONFIGURABLE-TDD-PATTERN-SUPPORTED}?
     |  +--ro tdd-group?                          uint8
     +--ro static-low-level-rx-endpoints* [name]
     |  +--ro name                                string
     |  +--ro restricted-interfaces*              -> /if:interfaces/interface/name
     |  +--ro array                               -> /user-plane-configuration/rx-arrays/name
     |  +--ro endpoint-type?                      -> ../../endpoint-types/id
     |  +--ro capacity-sharing-groups*            -> ../../endpoint-capacity-sharing-groups/id
     |  +--ro prach-group?                        -> ../../endpoint-prach-group/id
     |  +--ro compression
     |  |  +--ro dynamic-compression-supported?           boolean
     |  |  +--ro realtime-variable-bit-width-supported?   boolean
     |  |  +--ro supported-compression-set-id?            -> ../../../supported-compression-method-sets/id
     |  +--ro static-config-supported?            enumeration
     |  +--ro max-prach-patterns?                 uint8
     |  +--ro max-srs-patterns?                   uint8
     |  +--ro configurable-tdd-pattern-supported?   boolean {mcap:CONFIGURABLE-TDD-PATTERN-SUPPORTED}?
     |  +--ro tdd-group?                          uint8
     |  +--ro transmission-order?                 uint32 {feat:ORDERED-TRANSMISSION}?
     |  +--ro transmission-order-group?           uint32 {feat:ORDERED-TRANSMISSION}?
     +--rw low-level-tx-endpoints* [name]
     |  +--rw name                                 -> /user-plane-configuration/static-low-level-tx-
endpoints/name
     |  +--rw compression!
     |  |  +--rw iq-bitwidth?                          uint8
     |  |  +--rw compression-type                      enumeration
     |  |  x--rw bitwidth?                             uint8
     |  |  +--rw (compression-format)?
     |  |  |  +--:(no-compresison)
     |  |  |  +--:(block-floating-point)
     |  |  |  |  +--rw exponent?                       uint8
     |  |  |  +--:(block-floating-point-selective-re-sending)
     |  |  |  |  +--rw sres-exponent?                  uint8
     |  |  |  +--:(block-scaling)
     |  |  |  |  +--rw block-scalar?                   uint8
     |  |  |  +--:(u-law)
     |  |  |  |  +--rw comp-bit-width?                 uint8
     |  |  |  |  +--rw comp-shift?                     uint8
     |  |  |  +--:(beam-space-compression)
     |  |  |  |  +--rw active-beam-space-coeficient-mask*   uint8
```

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

```
     |  |  |  |  +--rw block-scaler?                        uint8
     |  |  |  +--:(modulation-compression)
     |  |  |  |  +--rw csf?                                 uint8
     |  |  |  |  +--rw mod-comp-scaler?                     uint16
     |  |  |  +--:(modulation-compression-selective-re-sending)
     |  |  |     +--rw sres-csf?                            uint8
     |  |  |     +--rw sres-mod-comp-scaler?                uint16
     |  |  +--rw fs-offset?                                 uint8 {cf:CONFIGURABLE-FS-OFFSET}?
     |  |  +--rw dynamic-compression-configuration* [id]
     |  |     +--rw id                    uint16
     |  |     +--rw iq-bitwidth?          uint8
     |  |     +--rw compression-method?   enumeration
     |  |     +--rw fs-offset?            uint8 {cf:CONFIGURABLE-FS-OFFSET}?
     |  +--rw frame-structure?                       uint8
     |  +--rw cp-type?                               enumeration
     |  +--rw cp-length                              uint16
     |  +--rw cp-length-other                        uint16
     |  +--rw offset-to-absolute-frequency-center    int32
     |  +--rw number-of-prb-per-scs* [scs]
     |  |  +--rw scs              mcap:scs-config-type
     |  |  +--rw number-of-prb    uint16
     |  +--rw e-axcid
     |  |  +--rw o-du-port-bitmask      uint16
     |  |  +--rw band-sector-bitmask    uint16
     |  |  +--rw ccid-bitmask           uint16
     |  |  +--rw ru-port-bitmask        uint16
     |  |  +--rw eaxc-id                uint16
     |  +--rw coupling-to?                           -> /mcap:module-capability/ru-capabilities/coupling-
methods/coupling-via-frequency-and-time
     |  +--rw coupling-method?                       enumeration
     |  +--rw configurable-tdd-pattern-supported?    -> /user-plane-configuration/static-low-level-rx-
endpoints[name=current()/../name]/configurable-tdd-pattern-supported
     +--rw low-level-rx-endpoints* [name]
     |  +--rw name                                   -> /user-plane-configuration/static-low-level-rx-
endpoints/name
     |  +--rw compression
     |  |  +--rw iq-bitwidth?                               uint8
     |  |  +--rw compression-type                           enumeration
     |  |  x--rw bitwidth?                                  uint8
     |  |  +--rw (compression-format)?
     |  |  |  +--:(no-compresison)
     |  |  |  +--:(block-floating-point)
     |  |  |  |  +--rw exponent?                            uint8
     |  |  |  +--:(block-floating-point-selective-re-sending)
     |  |  |  |  +--rw sres-exponent?                       uint8
     |  |  |  +--:(block-scaling)
     |  |  |  |  +--rw block-scalar?                        uint8
     |  |  |  +--:(u-law)
     |  |  |  |  +--rw comp-bit-width?                      uint8
     |  |  |  |  +--rw comp-shift?                          uint8
     |  |  |  +--:(beam-space-compression)
     |  |  |  |  +--rw active-beam-space-coeficient-mask*   uint8
     |  |  |  |  +--rw block-scaler?                        uint8
     |  |  |  +--:(modulation-compression)
     |  |  |  |  +--rw csf?                                 uint8
     |  |  |  |  +--rw mod-comp-scaler?                     uint16
     |  |  |  +--:(modulation-compression-selective-re-sending)
     |  |  |     +--rw sres-csf?                            uint8
     |  |  |     +--rw sres-mod-comp-scaler?                uint16
     |  |  +--rw fs-offset?                                 uint8 {cf:CONFIGURABLE-FS-OFFSET}?
     |  |  +--rw dynamic-compression-configuration* [id]
     |  |     +--rw id                    uint16
     |  |     +--rw iq-bitwidth?          uint8
     |  |     +--rw compression-method?   enumeration
     |  |     +--rw fs-offset?            uint8 {cf:CONFIGURABLE-FS-OFFSET}?
     |  +--rw frame-structure?                       uint8
     |  +--rw cp-type?                               enumeration
     |  +--rw cp-length                              uint16
     |  +--rw cp-length-other                        uint16
     |  +--rw offset-to-absolute-frequency-center    int32
     |  +--rw number-of-prb-per-scs* [scs]
     |  |  +--rw scs              mcap:scs-config-type
     |  |  +--rw number-of-prb    uint16
     |  +--rw ul-fft-sampling-offsets* [scs]
     |  |  +--rw scs                       mcap:scs-config-type
```

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

```
     |  |  +--rw ul-fft-sampling-offset?   uint16
     |  +--rw e-axcid
     |  |  +--rw o-du-port-bitmask      uint16
     |  |  +--rw band-sector-bitmask    uint16
     |  |  +--rw ccid-bitmask           uint16
     |  |  +--rw ru-port-bitmask        uint16
     |  |  +--rw eaxc-id                uint16
     |  +--rw eaxc-gain-correction?                decimal64 {EAXC-GAIN-CORRECTION}?
     |  +--rw non-time-managed-delay-enabled?      boolean
     |  +--rw coupling-to?                         -> /mcap:module-capability/ru-capabilities/coupling-
methods/coupling-via-frequency-and-time
     |  +--rw coupling-method?                     enumeration
     |  +--rw static-config-supported?             -> /user-plane-configuration/static-low-level-rx-
endpoints[name=current()/../name]/static-config-supported
     |  +--rw static-prach-configuration?          -> /user-plane-configuration/static-prach-
configurations/static-prach-config-id
     |  +--rw static-srs-configuration?            -> /user-plane-configuration/static-srs-
configurations/static-srs-config-id
     |  +--rw configurable-tdd-pattern-supported?  -> /user-plane-configuration/static-low-level-rx-
endpoints[name=current()/../name]/configurable-tdd-pattern-supported
     |  +--rw transmission-window-control?         enumeration {feat:STATIC-TRANSMISSION-WINDOW-CONTROL or
feat:DYNAMIC-TRANSMISSION-WINDOW-CONTROL}?
     |  +--rw transmission-window-schedule?        union {feat:STATIC-TRANSMISSION-WINDOW-CONTROL}?
     |  +--rw transmission-window-offset?          uint16 {feat:STATIC-TRANSMISSION-WINDOW-CONTROL}?
     |  +--rw transmission-window-size?            uint16 {feat:STATIC-TRANSMISSION-WINDOW-CONTROL}?
     |  +--rw transmission-type?                   enumeration {feat:STATIC-TRANSMISSION-WINDOW-CONTROL and
feat:UNIFORMLY-DISTRIBUTED-TRANSMISSION}?
     |  +--rw ordered-transmission?                boolean {feat:ORDERED-TRANSMISSION}?
     +--rw tx-array-carriers* [name]
     |  +--rw name                         string
     |  x--rw absolute-frequency-center?   uint32
     |  +--rw center-of-channel-bandwidth  uint64
     |  +--rw channel-bandwidth            uint64
     |  +--rw active?                      enumeration
     |  +--ro state                        enumeration
     |  +--rw type?                        enumeration
     |  +--ro duplex-scheme?               enumeration
     |  +--rw rw-duplex-scheme?            -> /user-plane-configuration/tx-array-
carriers[name=current()/../name]/duplex-scheme
     |  +--rw rw-type?                     -> /user-plane-configuration/tx-array-
carriers[name=current()/../name]/type
     |  +--rw band-number?                 -> /mcap:module-capability/band-capabilities/band-number
{mcap:LAA}?
     |  x--rw lte-tdd-frame
     |  |  +--rw subframe-assignment       enumeration
     |  |  +--rw special-subframe-pattern  enumeration
     |  +--rw laa-carrier-configuration {mcap:LAA}?
     |  |  +--rw ed-threshold-pdsch?          int8
     |  |  +--rw ed-threshold-drs?            int8
     |  |  +--rw tx-antenna-ports?            uint8
     |  |  +--rw transmission-power-for-drs?  int8
     |  |  +--rw dmtc-period?                 enumeration
     |  |  +--rw dmtc-offset?                 uint8
     |  |  +--rw lbt-timer?                   uint16
     |  |  +--rw max-cw-usage-counter* [priority]
     |  |     +--rw priority         enumeration
     |  |     +--rw counter-value?   uint8
     |  +--rw gain                         decimal64
     |  +--rw downlink-radio-frame-offset  uint32
     |  +--rw downlink-sfn-offset          int16
     |  +--rw reference-level?             decimal64 {TX-REFERENCE-LEVEL}?
     |  +--rw configurable-tdd-pattern?    -> /user-plane-configuration/configurable-tdd-patterns/tdd-
pattern-id {mcap:CONFIGURABLE-TDD-PATTERN-SUPPORTED}?
     +--rw rx-array-carriers* [name]
     |  +--rw name                         string
     |  x--rw absolute-frequency-center?   uint32
     |  +--rw center-of-channel-bandwidth  uint64
     |  +--rw channel-bandwidth            uint64
     |  +--rw active?                      enumeration
     |  +--ro state                        enumeration
     |  +--rw type?                        enumeration
     |  +--ro duplex-scheme?               enumeration
     |  +--rw downlink-radio-frame-offset  uint32
     |  +--rw downlink-sfn-offset          int16
     |  +--rw gain-correction              decimal64
```

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

```
      |  +--rw n-ta-offset                    uint32
      |  +--rw configurable-tdd-pattern?      -> /user-plane-configuration/configurable-tdd-patterns/tdd-
pattern-id {mcap:CONFIGURABLE-TDD-PATTERN-SUPPORTED}?
     +--ro tx-arrays* [name]
      |  +--ro name                                      string
      |  +--ro number-of-rows                            uint16
      |  +--ro number-of-columns                         uint16
      |  +--ro number-of-array-layers                    uint8
      |  +--ro horizontal-spacing?                       decimal64
      |  +--ro vertical-spacing?                         decimal64
      |  +--ro normal-vector-direction
      |  |  +--ro azimuth-angle?   decimal64
      |  |  +--ro zenith-angle?    decimal64
      |  +--ro leftmost-bottom-array-element-position
      |  |  +--ro x?   decimal64
      |  |  +--ro y?   decimal64
      |  |  +--ro z?   decimal64
      |  +--ro polarisations* [p]
      |  |  +--ro p                uint8
      |  |  +--ro polarisation    polarisation_type
      |  +--ro band-number                               -> /mcap:module-capability/band-capabilities/band-
number
      |  +--ro max-gain                                  decimal64
      |  +--ro independent-power-budget                  boolean
      |  +--ro capabilities* []
      |     +--ro max-supported-frequency-dl?   uint64
      |     +--ro min-supported-frequency-dl?   uint64
      |     +--ro max-supported-bandwidth-dl?   uint64
      |     +--ro max-num-carriers-dl?          uint32
      |     +--ro max-carrier-bandwidth-dl?     uint64
      |     +--ro min-carrier-bandwidth-dl?     uint64
      |     +--ro supported-technology-dl*      enumeration
     +--ro rx-arrays* [name]
      |  +--ro name                                      string
      |  +--ro number-of-rows                            uint16
      |  +--ro number-of-columns                         uint16
      |  +--ro number-of-array-layers                    uint8
      |  +--ro horizontal-spacing?                       decimal64
      |  +--ro vertical-spacing?                         decimal64
      |  +--ro normal-vector-direction
      |  |  +--ro azimuth-angle?   decimal64
      |  |  +--ro zenith-angle?    decimal64
      |  +--ro leftmost-bottom-array-element-position
      |  |  +--ro x?   decimal64
      |  |  +--ro y?   decimal64
      |  |  +--ro z?   decimal64
      |  +--ro polarisations* [p]
      |  |  +--ro p                uint8
      |  |  +--ro polarisation    polarisation_type
      |  +--ro band-number                               -> /mcap:module-capability/band-capabilities/band-
number
      |  +--ro gain-correction-range
      |  |  +--ro max    decimal64
      |  |  +--ro min    decimal64
      |  +--ro capabilities* []
      |     +--ro max-supported-frequency-ul?   uint64
      |     +--ro min-supported-frequency-ul?   uint64
      |     +--ro max-supported-bandwidth-ul?   uint64
      |     +--ro max-num-carriers-ul?          uint32
      |     +--ro max-carrier-bandwidth-ul?     uint64
      |     +--ro min-carrier-bandwidth-ul?     uint64
      |     +--ro supported-technology-ul*      enumeration
     +--ro relations* [entity]
      |  +--ro entity    uint16
      |  +--ro array1
      |  |  +--ro (antenna-type)?
      |  |     +--:(tx)
      |  |     |  +--ro tx-array-name?   -> /user-plane-configuration/tx-arrays/name
      |  |     +--:(rx)
      |  |        +--ro rx-array-name?   -> /user-plane-configuration/rx-arrays/name
      |  +--ro array2
      |  |  +--ro (antenna-type)?
      |  |     +--:(tx)
      |  |     |  +--ro tx-array-name?   -> /user-plane-configuration/tx-arrays/name
      |  |     +--:(rx)
```

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

```
     |  |        +--ro rx-array-name?   -> /user-plane-configuration/rx-arrays/name
     |  +--ro types* [relation-type]
     |     +--ro relation-type    enumeration
     |     +--ro pairs* [element-array1]
     |        +--ro element-array1    uint16
     |        +--ro element-array2?   uint16
     +--rw eaxc-id-group-configuration {mcap:EAXC-ID-GROUP-SUPPORTED}?
     |  +--rw max-num-tx-eaxc-id-groups?       -> /mcap:module-capability/ru-capabilities/eaxcid-grouping-
capabilities/max-num-tx-eaxc-id-groups
     |  +--rw max-num-tx-eaxc-ids-per-group?   -> /mcap:module-capability/ru-capabilities/eaxcid-grouping-
capabilities/max-num-tx-eaxc-ids-per-group
     |  +--rw max-num-rx-eaxc-id-groups?       -> /mcap:module-capability/ru-capabilities/eaxcid-grouping-
capabilities/max-num-rx-eaxc-id-groups
     |  +--rw max-num-rx-eaxc-ids-per-group?   -> /mcap:module-capability/ru-capabilities/eaxcid-grouping-
capabilities/max-num-rx-eaxc-ids-per-group
     |  +--rw tx-eaxc-id-group* [representative-tx-eaxc-id]
     |  |  +--rw representative-tx-eaxc-id    uint16
     |  |  +--rw member-tx-eaxc-id*           uint16
     |  +--rw rx-eaxc-id-group* [representative-rx-eaxc-id]
     |     +--rw representative-rx-eaxc-id    uint16
     |     +--rw member-rx-eaxc-id*           uint16
     +--rw static-prach-configurations* [static-prach-config-id] {mcap:PRACH-STATIC-CONFIGURATION-SUPPORTED}?
     |  +--rw static-prach-config-id    uint8
     |  +--rw pattern-period            uint16
     |  +--rw guard-tone-low-re         uint32
     |  +--rw num-prach-re              uint32
     |  +--rw guard-tone-high-re        uint32
     |  +--rw sequence-duration         uint32
     |  +--rw prach-patterns* [prach-pattern-id]
     |     +--rw prach-pattern-id         uint16
     |     +--rw number-of-repetitions    uint8
     |     +--rw number-of-occasions      uint8
     |     +--rw re-offset                uint32
     |     +--rw occasion-parameters* [occasion-id]
     |     |  +--rw occasion-id    uint8
     |     |  +--rw cp-length      uint16
     |     |  +--rw gp-length?     uint16
     |     |  +--rw beam-id        uint16
     |     +--rw frame-number             uint16
     |     +--rw sub-frame-id             uint16
     |     +--rw time-offset              uint16
     +--rw static-srs-configurations* [static-srs-config-id] {mcap:SRS-STATIC-CONFIGURATION-SUPPORTED}?
     |  +--rw static-srs-config-id    uint8
     |  +--rw pattern-period          uint16
     |  +--rw srs-patterns* [srs-pattern-id]
     |     +--rw srs-pattern-id     uint16
     |     +--rw sub-frame-id       uint16
     |     +--rw slot-id            uint16
     |     +--rw start-symbol-id    uint16
     |     +--rw beam-id            uint16
     |     +--rw num-symbol         uint16
     |     +--rw start-prbc         uint16
     |     +--rw num-prbc           uint16
     +--rw configurable-tdd-patterns* [tdd-pattern-id] {mcap:CONFIGURABLE-TDD-PATTERN-SUPPORTED}?
     |  +--rw tdd-pattern-id      uint8
     |  +--rw switching-points* [switching-point-id]
     |     +--rw switching-point-id    uint16
     |     +--rw direction             enumeration
     |     +--rw frame-offset          uint32
     +--rw general-config
        +--rw regularization-factor-se-configured?   boolean
        +--rw little-endian-byte-order?              boolean

  notifications:
    +---n tx-array-carriers-state-change
    |  +--ro tx-array-carriers* [name]
    |     +--ro name     -> /user-plane-configuration/tx-array-carriers/name
    |     +--ro state?   -> /user-plane-configuration/tx-array-carriers/state
    +---n rx-array-carriers-state-change
       +--ro rx-array-carriers* [name]
          +--ro name     -> /user-plane-configuration/rx-array-carriers/name
          +--ro state?   -> /user-plane-configuration/rx-array-carriers/state
```

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

D.2.9 o-ran-ald Module

The format for the ald module is provided below

```
module: o-ran-ald
  rpcs:
    +---x ald-communication
       +---w input
       |  +---w port-id        -> /ap:ald-ports-io/ald-port/port-id
       |  +---w ald-req-msg?   binary
       +--ro output
          +--ro port-id                       -> /ap:ald-ports-io/ald-port/port-id
          +--ro status                        enumeration
          +--ro error-message?                string
          +--ro ald-resp-msg?                 binary
          +--ro frames-with-wrong-crc?        uint32
          +--ro frames-without-stop-flag?     uint32
          +--ro number-of-received-octets?    uint32
```

D.2.10 o-ran-troubleshooting Module

The format for the troubleshooting module is provided below

```
module: o-ran-troubleshooting

  rpcs:
    +---x start-troubleshooting-logs
    |  +--ro output
    |     +--ro status?           enumeration
    |     +--ro failure-reason?   string
    +---x stop-troubleshooting-logs
       +--ro output
          +--ro status?           enumeration
          +--ro failure-reason?   string

  notifications:
    +---n troubleshooting-log-generated
       +--ro log-file-name*   string
```

D.2.11 o-ran-laa-operations Module

The format for the LAA operations module is provided below

```
rpcs:
  rpcs:
    +---x start-measurements {mcap:LAA}?
       +---w input
       |  +---w band-config* [band-number]
       |  |  +---w band-number                 band-num
       |  |  +---w channel-center-frequency*   uint16
       |  +---w duration-per-channel?    uint16
       |  +---w maximum-response-time?   uint16
       +--ro output
          +--ro band-config* [band-number]
             +--ro band-number                 band-num
             +--ro carrier-center-frequency*   uint16
             +--ro status?                     enumeration
             +--ro error-message?              string

  notifications:
    +---n measurement-result {mcap:LAA}?
       +--ro band-result* [band-number]
          +--ro band-number            band-num
          +--ro measurement-success?   boolean
          +--ro failure-message?       enumeration
          +--ro channel-result* [measured-channel]
             +--ro measured-channel    uint16
             +--ro occupancy-ratio?    uint8
             +--ro average-rssi?       int8
```

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

D.2.11 o-ran-trace Module

The format for the trace operations module is provided below

```
module: o-ran-trace

  rpcs:
    +---x start-trace-logs
    |  +--ro output
    |     +--ro status?           enumeration
    |     +--ro failure-reason?   string
    +---x stop-trace-logs
       +--ro output
          +--ro status?           enumeration
          +--ro failure-reason?   string

  notifications:
    +---n trace-log-generated
       +--ro log-file-name*          string
       +--ro is-notification-last?   Boolean
```

D.3 Interfaces Folder

D.3.1 o-ran-interfaces.yang Module

The format for the interfaces module is provided below

```
module: o-ran-interfaces
  augment /if:interfaces/if:interface:
    +--rw l2-mtu?             uint16
    +--rw alias-macs*         yang:mac-address {ALIASMAC-BASED-CU-PLANE}?
    +--rw vlan-tagging?       boolean
    +--rw class-of-service
    |  +--rw u-plane-marking?            pcp
    |  +--rw c-plane-marking?            pcp
    |  +--rw m-plane-marking?            pcp
    |  +--rw s-plane-marking?            pcp
    |  +--rw other-marking?              pcp
    |  +--rw enhanced-uplane-markings* [up-marking-name]
    |     +--rw up-marking-name     string
    |     +--rw enhanced-marking?   pcp
    +--ro interface-groups-id*   -> /if:interfaces/o-ran-int:interface-grouping/interfaces-groups/interface-
group-id
  augment /if:interfaces/if:interface:
    +--rw base-interface?   if:interface-ref
    +--rw vlan-id?          uint16
  augment /if:interfaces/if:interface:
    +--rw mac-address?      yang:mac-address
    +--rw port-reference
    |  +--rw port-name?     -> /hw:hardware/component/name
    |  +--rw port-number?   uint8
    +--ro last-cleared?     yang:date-and-time
  augment /if:interfaces/if:interface/ip:ipv4:
    +--rw diffserv-markings {UDPIP-BASED-CU-PLANE}?
       +--rw u-plane-marking?            inet:dscp
       +--rw c-plane-marking?            inet:dscp
       +--rw s-plane-marking?            inet:dscp
       +--rw other-marking?              inet:dscp
       +--rw enhanced-uplane-markings* [up-marking-name]
          +--rw up-marking-name     string
          +--rw enhanced-marking?   inet:dscp
  augment /if:interfaces/if:interface/ip:ipv6:
    +--rw diffserv-markings {UDPIP-BASED-CU-PLANE}?
       +--rw u-plane-marking?            inet:dscp
       +--rw c-plane-marking?            inet:dscp
       +--rw s-plane-marking?            inet:dscp
       +--rw other-marking?              inet:dscp
       +--rw enhanced-uplane-markings* [up-marking-name]
          +--rw up-marking-name     string
          +--rw enhanced-marking?   inet:dscp
  augment /if:interfaces/if:interface/ip:ipv4:
```

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

```
    +--rw m-plane-marking?   inet:dscp
  augment /if:interfaces/if:interface/ip:ipv6:
    +--rw m-plane-marking?   inet:dscp
  augment /if:interfaces:
    +--ro interface-grouping!
       +--ro interfaces-groups* [interface-group-id]
          +--ro interface-group-id                     uint8
          +--ro max-sustainable-ingress-bandwidth?   uint32
          +--ro max-sustainable-egress-bandwidth?    uint32

  rpcs:
    +---x reset-interface-counters
```

D.3.2 o-ran-processing-elements.yang Module

The format for the processing elements module is provided below

```
module: o-ran-processing-element
  +--rw processing-elements
     +--ro maximum-number-of-transport-flows?   uint16
     +--rw transport-session-type?             enumeration
     +--rw enhanced-uplane-mapping!
     |  +--rw uplane-mapping* [up-marking-name]
     |     +--rw up-marking-name         string
     |     +--rw (up-markings)?
     |        +--:(ethernet)
     |        |  +--rw up-cos-name?      -> /if:interfaces/interface/o-ran-int:class-of-service/enhanced-
uplane-markings/up-marking-name
     |        +--:(ipv4)
     |        |  +--rw upv4-dscp-name?   -> /if:interfaces/interface/ip:ipv4/o-ran-int:diffserv-
markings/enhanced-uplane-markings/up-marking-name
     |        +--:(ipv6)
     |           +--rw upv6-dscp-name?   -> /if:interfaces/interface/ip:ipv6/o-ran-int:diffserv-
markings/enhanced-uplane-markings/up-marking-name
     +--rw ru-elements* [name]
        +--rw name               string
        +--rw transport-flow
           +--rw interface-name?   -> /if:interfaces/interface/name
           +--rw aliasmac-flow {o-ran-int:ALIASMAC-BASED-CU-PLANE}?
           |  +--rw ru-aliasmac-address    -> /if:interfaces/interface[if:name = current()/../../interface-
name]/o-ran-int:alias-macs
           |  +--rw vlan-id?               -> /if:interfaces/interface[if:name = current()/../../interface-
name]/o-ran-int:vlan-id
           |  +--rw o-du-mac-address       yang:mac-address
           +--rw eth-flow
           |  +--rw ru-mac-address      -> /if:interfaces/interface[if:name = current()/../../interface-
name]/o-ran-int:mac-address
           |  +--rw vlan-id             -> /if:interfaces/interface[if:name = current()/../../interface-
name]/o-ran-int:vlan-id
           |  +--rw o-du-mac-address    yang:mac-address
           +--rw udpip-flow
           |  +--rw (address)
           |  |  +--:(ru-ipv4-address)
           |  |  |  +--rw ru-ipv4-address?     -> /if:interfaces/interface[if:name =
current()/../../interface-name]/ip:ipv4/address/ip
           |  |  +--:(ru-ipv6-address)
           |  |     +--rw ru-ipv6-address?     -> /if:interfaces/interface[if:name =
current()/../../interface-name]/ip:ipv6/address/ip
           |  +--rw o-du-ip-address           inet:ip-address
           |  +--rw ru-ephemeral-udp-port      inet:port-number
           |  +--rw o-du-ephemeral-udp-port    inet:port-number
           |  +--rw ecpri-destination-udp      inet:port-number
           +--rw north-eth-flow {SHARED_CELL}?
           |  +--rw ru-mac-address?          -> /if:interfaces/interface[if:name =
current()/../../interface-name]/o-ran-int:mac-address
           |  +--rw vlan-id?                 -> /if:interfaces/interface[if:name =
current()/../../interface-name]/o-ran-int:vlan-id
           |  +--rw north-node-mac-address?   yang:mac-address
           +--rw south-eth-flow {SHARED_CELL}?
              +--rw ru-mac-address?          -> /if:interfaces/interface[if:name =
current()/../../interface-name]/o-ran-int:mac-address
              +--rw vlan-id?                 -> /if:interfaces/interface[if:name =
current()/../../interface-name]/o-ran-int:vlan-id
              +--rw south-node-mac-address?   yang:mac-address
```

D.3.3 o-ran-transceiver.yang Module

The format for the (SFP) transciver module is provided below

```
module: o-ran-transceiver
  +--rw port-transceivers
     +--rw port-transceiver-data* [interface-name port-number]
        +--rw interface-name                      -> /if:interfaces/interface/name
        +--rw port-number                         -> /if:interfaces/interface[if:name =
current()/../interface-name]/o-ran-int:port-reference/port-number
        +--rw name?                               string
        +--ro present                             boolean
        +--ro vendor-id?                          string
        +--ro vendor-part?                        string
        +--ro vendor-rev?                         string
        +--ro serial-no?                          string
        +--ro SFF8472-compliance-code?            enumeration
        +--ro connector-type?                     enumeration
        +--ro identifier?                         enumeration
        +--ro nominal-bitrate?                    uint32
        +--ro low-bitrate-margin?                 uint8
        +--ro high-bitrate-margin?                uint8
        +--ro rx-power-type?                      enumeration
        +--ro rx-power?                           decimal64
        +--ro tx-power?                           decimal64
        +--ro tx-bias-current?                    decimal64
        +--ro voltage?                            decimal64
        +--ro temperature?                        decimal64
        +--ro additional-multi-lane-reporting* [lane]
           +--ro lane               uint8
           +--ro rx-power?          decimal64
           +--ro tx-bias-current?   decimal64
           +--ro tx-power?          decimal64
```

D.3.4 o-ran-mplane-int.yang Module

The format for the management plane interface module is provided below

```
module: o-ran-mplane-int
  +--rw mplane-info
     +--rw searchable-mplane-access-vlans-info
     |  +--rw searchable-access-vlans*   vlan-id
     |  +--rw vlan-range
     |     +--rw lowest-vlan-id?    vlan-id
     |     +--rw highest-vlan-id?   vlan-id
     +--rw m-plane-interfaces
     |  +--rw m-plane-sub-interfaces* [interface-name sub-interface]
     |  |  +--rw interface-name    -> /if:interfaces/interface/name
     |  |  +--rw sub-interface     -> /if:interfaces/interface[if:name = current()/../interface-name]/o-ran-
int:vlan-id
     |  |  +--ro client-info
     |  |     +--ro mplane-ipv4-info* [mplane-ipv4]
     |  |     |  +--ro mplane-ipv4    inet:ipv4-address
     |  |     |  +--ro port?          inet:port-number
     |  |     +--ro mplane-ipv6-info* [mplane-ipv6]
     |  |     |  +--ro mplane-ipv6    inet:ipv6-address
     |  |     |  +--ro port?          inet:port-number
     |  |     +--ro mplane-fqdn*        inet:domain-name
     |  +--rw m-plane-ssh-ports
     |     +--rw call-home-ssh-port?   inet:port-number
     |     +--rw server-ssh-port?      inet:port-number
     |  +--rw m-plane-tls-ports
     |     +--rw call-home-tls-port?   inet:port-number
     |     +--rw server-tls-port?      inet:port-number
     +--rw configured-client-info
        +--rw mplane-ipv4-info* [mplane-ipv4]
        |  +--rw mplane-ipv4    inet:ipv4-address
        |  +--rw port?          inet:port-number
        +--rw mplane-ipv6-info* [mplane-ipv6]
        |  +--rw mplane-ipv6    inet:ipv6-address
        |  +--rw port?          inet:port-number
        +--rw mplane-fqdn*        inet:domain-name
```

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

D.3.5 o-ran-dhcp.yang Module

The format for the dhcp module is provided below.

```
module: o-ran-dhcp
  +--ro dhcp
     +--ro interfaces* [interface]
     |  +--ro interface    if:interface-ref
     |  +--ro dhcpv4
     |  |  +--ro client-id?                string
     |  |  +--ro dhcp-server-identifier?   inet:ip-address
     |  |  +--ro domain-name?              string
     |  |  +--ro domain-name-servers*      inet:ip-address
     |  |  +--ro interface-mtu?            uint32
     |  |  +--ro default-gateways*         inet:ip-address
     |  |  +--ro netconf-clients* [client]
     |  |  |  +--ro client           netconf-client-id
     |  |  |  +--ro optional-port?   inet:port-number
     |  |  +--ro ca-ra-servers* [servers]
     |  |  |  +--ro servers         ca-ra-server-id
     |  |  |  +--ro port-number?    inet:port-number
     |  |  |  +--ro ca-ra-path?     string
     |  |  |  +--ro subject-name?   string
     |  |  |  +--ro protocol?       enumeration
     |  |  +--ro segw* [gateways]
     |  |  |  +--ro gateways    segw-id
     |  |  +--ro event-collectors*         event-collector-id {or-feat:NON-PERSISTENT-MPLANE}?
     |  |  +--ro event-collector-format?   enumeration {or-feat:NON-PERSISTENT-MPLANE}?
     |  +--ro dhcpv6
     |     +--ro dhcp-client-identifier
     |     +--ro dhcp-server-identifier
     |     +--ro domain-name?              string
     |     +--ro domain-name-servers*      inet:ip-address
     |     +--ro netconf-clients* [client]
     |     |  +--ro client           netconf-client-id
     |     |  +--ro optional-port?   inet:port-number
     |     +--ro ca-ra-servers* [servers]
     |     |  +--ro servers         ca-ra-server-id
     |     |  +--ro port-number?    inet:port-number
     |     |  +--ro ca-ra-path?     string
     |     |  +--ro subject-name?   string
     |     |  +--ro protocol?       enumeration
     |     +--ro segw* [gateways]
     |     |  +--ro gateways    segw-id
     |     +--ro event-collectors*         event-collector-id {or-feat:NON-PERSISTENT-MPLANE}?
     |     +--ro event-collector-format?   enumeration {or-feat:NON-PERSISTENT-MPLANE}?
     +--ro m-plane-dhcp
        x--ro private-enterprise-number?   uint16
        +--ro private-enterprise-num?      uint32
        +--ro vendor-class-data?           String
```

D.3.6 o-ran-externalio.yang Module

The format for the external input/output module is provided below

```
module: o-ran-externalio
  +--rw external-io
     +--ro input* [name]
     |  +--ro name       string
     |  +--ro port-in?   uint8
     |  +--ro line-in?   boolean
     +--ro output* [name]
     |  +--ro name        string
     |  +--ro port-out    uint8
     +--rw output-setting* [name]
        +--rw name        -> /external-io/output/name
        +--rw line-out?   boolean
  notifications:
    +---n external-input-change
       +--ro current-input-notification
          +--ro external-input* [name]
             +--ro name       -> /external-io/input/name
             +--ro io-port?   -> /external-io/input/port-in
             +--ro line-in?   -> /external-io/input/line-in
```

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

D.3.7 o-ran-ald-port.yang Module

The format for the Antenna Line Device module is provided below

```
module: o-ran-ald-port
  +--rw ald-ports-io
     +--ro over-current-supported?   boolean
     +--ro ald-port* [name]
     |  +--ro name                    string
     |  +--ro port-id                 uint8
     |  +--ro dc-control-support      boolean
     |  +--ro dc-enabled-status?      boolean
     |  +--ro supported-connector     enumeration
     +--rw ald-port-dc-control* [name]
        +--rw name          -> /ald-ports-io/ald-port/name
        +--rw dc-enabled?   boolean

  notifications:
    +---n overcurrent-report {OVERCURRENT-SUPPORTED}?
     | +--ro overload-condition
     |    +--ro overloaded-ports*   -> /ald-ports-io/ald-port/name
    +---n dc-enabled-status-change
       +--ro ald-port* [name]
          +--ro name                  -> /ald-ports-io/ald-port/name
          +--ro dc-enabled-status?   -> /ald-ports-io/ald-port/dc-enabled-status
```

D.3.8 o-ran-ethernet-forwarding.yang Module

The format for the module o-ran Ethernet forwarding is provided below.

```
module: o-ran-ethernet-forwarding
  +--rw ethernet-forwarding-table
     +--rw aging-time?        uint32
     +--ro filtering-entry* [address vlan-id]
        +--ro address     yang:mac-address
        +--ro vlan-id     uint16
        +--ro port-map* [port-ref]
           +--ro port-ref    -> /if:interfaces/interface/or-if:port-reference/port-number
```

D.4 Sync Folder

D.4.1 o-ran-sync.yang Module

The format for the synchronization module is provided below

```
module: o-ran-sync
  +--rw sync
     +--ro sync-status
     |  +--ro sync-state                    enumeration
     |  +--ro time-error?                   decimal64
     |  +--ro frequency-error?              decimal64
     |  +--ro supported-reference-types* [item]
     |     +--ro item    enumeration
     +--ro sync-capability
     |  +--ro sync-t-tsc    enumeration
     +--rw ptp-config
     |  +--rw domain-number?             uint8
     |  +--rw accepted-clock-classes* [clock-classes]
     |  |  +--rw clock-classes    uint8
     |  +--rw ptp-profile?               Enumeration
     |  +--rw delay-asymmetry?           int16
     |  +--rw g-8275-1-config
     |  |  +--rw multicast-mac-address?   enumeration
     |  |  x--rw delay-asymmetry?         int16
     |  +--rw g-8275-2-config
     |     +--rw local-ip-port?            -> /if:interfaces/interface/name
     |     +--rw master-ip-configuration* [local-priority]
     |     |  +--rw local-priority    uint8
     |     |  +--rw ip-address?       string
     |     +--rw log-inter-sync-period?       int8
```

```
     |     +--rw log-inter-announce-period?   int8
     +--rw ptp-status
     |  +--rw reporting-period?           uint8
     |  +--ro lock-state?                 enumeration
     |  +--ro clock-class?                uint8
     |  +--ro clock-identity?             string
     |  +--ro partial-timing-supported?   boolean
     |  +--ro sources* [local-port-number]
     |     +--ro local-port-number            -> /if:interfaces/interface/o-ran-int:port-reference/port-
number
     |     +--ro state?                       enumeration
     |     +--ro two-step-flag?               boolean
     |     +--ro leap61?                      boolean
     |     +--ro leap59?                      boolean
     |     +--ro current-utc-offset-valid?    boolean
     |     +--ro ptp-timescale?               boolean
     |     +--ro time-traceable?              boolean
     |     +--ro frequency-traceable?         boolean
     |     +--ro source-clock-identity?       string
     |     +--ro source-port-number?          uint16
     |     +--ro current-utc-offset?          int16
     |     +--ro priority1?                   uint8
     |     +--ro clock-class?                 uint8
     |     +--ro clock-accuracy?              uint8
     |     +--ro offset-scaled-log-variance?  uint16
     |     +--ro priority2?                   uint8
     |     +--ro grandmaster-clock-identity?  string
     |     +--ro steps-removed?               uint16
     |     +--ro time-source?                 uint8
     +--rw synce-config
     |  +--rw acceptance-list-of-ssm*   enumeration
     |  +--rw ssm-timeout?              uint16
     +--rw synce-status
     |  +--rw reporting-period?   uint8
     |  +--ro lock-state?         enumeration
     |  +--ro sources* [local-port-number]
     |     +--ro local-port-number    -> /if:interfaces/interface/o-ran-int:port-reference/port-number
     |     +--ro state?               enumeration
     |     +--ro quality-level?       uint8
     +--rw gnss-config {GNSS}?
     |  +--rw enable?                         boolean
     |  +--rw satellite-constelation-list*    enumeration
     |  +--rw polarity?                       enumeration
     |  +--rw cable-delay?                    uint16
     |  +--rw anti-jam-enable?                boolean {ANTI-JAM}?
     +--rw gnss-status {GNSS}?
        +--rw reporting-period?   uint8
        +--ro name?               string
        +--ro gnss-sync-status?   enumeration
        +--ro gnss-data
           +--ro satellites-tracked?   uint8
           +--ro location
           |  +--ro altitude?    int64
           |  +--ro latitude?    geographic-coordinate-degree
           |  +--ro longitude?   geographic-coordinate-degree
           +--ro gnss-rx-time-error?   decimal64

  notifications:
    +---n synchronization-state-change
    |  +--ro sync-state?   -> /sync/sync-status/sync-state
    +---n ptp-state-change
    |  +--ro ptp-state?   -> /sync/ptp-status/lock-state
    +---n synce-state-change
    |  +--ro synce-state?   -> /sync/synce-status/lock-state
    +---n gnss-state-change {GNSS}?
       +--ro gnss-state?   -> /sync/gnss-status/gnss-sync-status
```

D.5 Radio Folder

D.5.1 o-ran-module-cap.yang Module

The format for the module capabilitites module is provided below

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

```
module: o-ran-module-cap
  +--rw module-capability
     +--ro ru-capabilities
     |  +--ro ru-supported-category?                          enumeration
     |  x--ro number-of-ru-ports?                             uint8
     |  +--ro number-of-ru-ports-ul?                          uint8
     |  +--ro number-of-ru-ports-dl?                          uint8
     |  +--ro number-of-spatial-streams?                      uint8
     |  +--ro max-power-per-pa-antenna?                       decimal64
     |  +--ro min-power-per-pa-antenna?                       decimal64
     |  +--ro fronthaul-split-option?                         uint8
     |  +--ro format-of-iq-sample
     |  |  +--ro dynamic-compression-supported?             boolean
     |  |  +--ro realtime-variable-bit-width-supported?     boolean
     |  |  +--ro compression-method-supported* []
     |  |  |  +--ro iq-bitwidth?                             uint8
     |  |  |  +--ro compression-type                         enumeration
     |  |  |  x--ro bitwidth?                                uint8
     |  |  |  +--ro (compression-format)?
     |  |  |  |  +--:(no-compresison)
     |  |  |  |  +--:(block-floating-point)
     |  |  |  |  |  +--ro exponent?                          uint8
     |  |  |  |  +--:(block-floating-point-selective-re-sending)
     |  |  |  |  |  +--ro sres-exponent?                     uint8
     |  |  |  |  +--:(block-scaling)
     |  |  |  |  |  +--ro block-scalar?                      uint8
     |  |  |  |  +--:(u-law)
     |  |  |  |  |  +--ro comp-bit-width?                    uint8
     |  |  |  |  |  +--ro comp-shift?                        uint8
     |  |  |  |  +--:(beam-space-compression)
     |  |  |  |  |  +--ro active-beam-space-coeficient-mask*   uint8
     |  |  |  |  |  +--ro block-scaler?                      uint8
     |  |  |  |  +--:(modulation-compression)
     |  |  |  |  |  +--ro csf?                               uint8
     |  |  |  |  |  +--ro mod-comp-scaler?                   uint16
     |  |  |  |  +--:(modulation-compression-selective-re-sending)
     |  |  |  |     +--ro sres-csf?                          uint8
     |  |  |  |     +--ro sres-mod-comp-scaler?              uint16
     |  |  |  +--ro fs-offset*                               uint8 {cf:CONFIGURABLE-FS-OFFSET}?
     |  |  +--ro variable-bit-width-per-channel-supported?   boolean
     |  |  +--ro syminc-supported?                          boolean
     |  |  +--ro regularization-factor-se-supported?        boolean
     |  |  +--ro little-endian-supported?                   boolean
     |  +--ro ul-mixed-num-required-guard-rbs* [scs-a scs-b]
     |  |  +--ro scs-a                    scs-config-type
     |  |  +--ro scs-b                    scs-config-type
     |  |  +--ro number-of-guard-rbs-ul?   uint8
     |  +--ro dl-mixed-num-required-guard-rbs* [scs-a scs-b]
     |  |  +--ro scs-a                    scs-config-type
     |  |  +--ro scs-b                    scs-config-type
     |  |  +--ro number-of-guard-rbs-dl?   uint8
     |  +--ro energy-saving-by-transmission-blanks          boolean
     |  +--ro eaxcid-grouping-capabilities {o-ran-module-cap:EAXC-ID-GROUP-SUPPORTED}?
     |  |  +--ro max-num-tx-eaxc-id-groups?       uint8
     |  |  +--ro max-num-tx-eaxc-ids-per-group?   uint8
     |  |  +--ro max-num-rx-eaxc-id-groups?       uint8
     |  |  +--ro max-num-rx-eaxc-ids-per-group?   uint8
     |  +--ro dynamic-transport-delay-management-supported    boolean
     |  +--ro support-only-unique-ecpri-seqid-per-eaxc?       boolean
     |  +--ro coupling-methods
     |  |  +--ro coupling-via-frequency-and-time?                   boolean
     |  |  +--ro coupling-via-frequency-and-time-with-priorities?   Boolean
     |  |  +--ro coupling-via-frequency-and-time-with-priorities-optimized?   boolean
     |  +--ro ud-comp-len-supported?                          boolean
     +--ro band-capabilities* [band-number]
     |  +--ro band-number                  uint16
     |  +--ro sub-band-info {o-ran-module-cap:LAA}?
     |  |  +--ro sub-band-frequency-ranges* [sub-band]
     |  |  |  +--ro sub-band                  sub-band-string
     |  |  |  +--ro max-supported-frequency-dl?   uint64
     |  |  |  +--ro min-supported-frequency-dl?   uint64
     |  |  +--ro number-of-laa-scarriers?     uint8
     |  |  +--ro maximum-laa-buffer-size?     uint16
     |  |  +--ro maximum-processing-time?     uint16
     |  |  +--ro self-configure?              boolean
```

```
      |  +--ro max-supported-frequency-dl?   uint64
      |  +--ro min-supported-frequency-dl?   uint64
      |  +--ro max-supported-bandwidth-dl?   uint64
      |  +--ro max-num-carriers-dl?          uint32
      |  +--ro max-carrier-bandwidth-dl?     uint64
      |  +--ro min-carrier-bandwidth-dl?     uint64
      |  +--ro supported-technology-dl*      enumeration
      |  +--ro max-supported-frequency-ul?   uint64
      |  +--ro min-supported-frequency-ul?   uint64
      |  +--ro max-supported-bandwidth-ul?   uint64
      |  +--ro max-num-carriers-ul?          uint32
      |  +--ro max-carrier-bandwidth-ul?     uint64
      |  +--ro min-carrier-bandwidth-ul?     uint64
      |  +--ro supported-technology-ul*      enumeration
      |  +--ro max-num-component-carriers?   uint8
      |  +--ro max-num-bands?                uint16
      |  +--ro max-num-sectors?              uint8
      |  +--ro max-power-per-antenna?        decimal64
      |  +--ro min-power-per-antenna?        decimal64
      |  +--ro codebook-configuration_ng?    uint8
      |  +--ro codebook-configuration_n1?    uint8
      |  +--ro codebook-configuration_n2?    uint8
     +--rw rw-sub-band-info {o-ran-module-cap:LAA}?
        +--rw rw-number-of-laa-scarriers?   -> /module-capability/band-capabilities/sub-band-info/number-of-laa-scarriers
        +--rw rw-self-configure?            -> /module-capability/band-capabilities/sub-band-info/self-configure
```

D.5.2 o-ran-delay-management.yang Module

The format for the delay management module is provided below

```
module: o-ran-delay-management
  +--rw delay-management
     +--rw bandwidth-scs-delay-state* [bandwidth subcarrier-spacing]
     |  +--rw bandwidth             bandwidth
     |  +--rw subcarrier-spacing    uint32
     |  +--ro ru-delay-profile
     |     +--ro t2a-min-up       uint32
     |     +--ro t2a-max-up       uint32
     |     +--ro t2a-min-cp-dl    uint32
     |     +--ro t2a-max-cp-dl    uint32
     |     +--ro tcp-adv-dl       uint32
     |     +--ro ta3-min          uint32
     |     +--ro ta3-max          uint32
     |     +--ro t2a-min-cp-ul    uint32
     |     +--ro t2a-max-cp-ul    uint32
     +--rw adaptive-delay-configuration {ADAPTIVE-RU-PROFILE}?
        +--rw bandwidth-scs-delay-state* [bandwidth subcarrier-spacing]
        |  +--rw bandwidth             bandwidth
        |  +--rw subcarrier-spacing    uint32
        |  +--rw o-du-delay-profile
        |     +--rw t1a-max-up?      uint32
        |     +--rw tx-max?          uint32
        |     +--rw ta4-max?         uint32
        |     +--rw rx-max?          uint32
        |     +--rw t1a-max-cp-dl?   uint32
        +--rw transport-delay
           +--rw t12-min?   uint32
           +--rw t12-max?   uint32
           +--rw t34-min?   uint32
           +--rw t34-max?   uint32
```

D.5.3 o-ran-beamforming.yang Module

The format for the beamforming module is provided below

```
module: o-ran-beamforming
  +--ro beamforming-config
     x--ro per-band-config* [band-number]
     |  +--ro band-number         -> /mcap:module-capability/band-capabilities/band-number
     |  +--ro tx-array*           -> /up:user-plane-configuration/tx-arrays/name
     |  +--ro rx-array*           -> /up:user-plane-configuration/rx-arrays/name
     |  +--ro static-properties
```

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

```
|  |  +--ro rt-bf-weights-update-support?   boolean
|  |  +--ro (beamforming-type)?
|  |  |  +--:(frequency)
|  |  |  |  +--ro frequency-domain-beams
|  |  |  |     +--ro max-number-of-beam-ids                     uint16
|  |  |  |     +--ro initial-beam-id                            uint16
|  |  |  |     +--ro iq-bitwidth?                               uint8
|  |  |  |     +--ro compression-type                           enumeration
|  |  |  |     x--ro bitwidth?                                  uint8
|  |  |  |     +--ro (compression-format)?
|  |  |  |     |  +--:(no-compresison)
|  |  |  |     |  +--:(block-floating-point)
|  |  |  |     |  |  +--ro exponent?                            uint8
|  |  |  |     |  +--:(block-scaling)
|  |  |  |     |  |  +--ro block-scalar?                        uint8
|  |  |  |     |  +--:(u-law)
|  |  |  |     |  |  +--ro comp-bit-width?                      uint8
|  |  |  |     |  |  +--ro comp-shift?                          uint8
|  |  |  |     |  +--:(beam-space-compression)
|  |  |  |     |  |  +--ro active-beam-space-coeficient-mask*   uint8
|  |  |  |     |  |  +--ro block-scaler?                        uint8
|  |  |  |     |  +--:(modulation-compression)
|  |  |  |     |     +--ro csf?                                 uint8
|  |  |  |     |     +--ro mod-comp-scaler?                     uint16
|  |  |  |     +--ro additional-compression-method-supported* []
|  |  |  |        +--ro iq-bitwidth?                               uint8
|  |  |  |        +--ro compression-type                           enumeration
|  |  |  |        x--ro bitwidth?                                  uint8
|  |  |  |        +--ro (compression-format)?
|  |  |  |           +--:(no-compresison)
|  |  |  |           +--:(block-floating-point)
|  |  |  |           |  +--ro exponent?                            uint8
|  |  |  |           +--:(block-scaling)
|  |  |  |           |  +--ro block-scalar?                        uint8
|  |  |  |           +--:(u-law)
|  |  |  |           |  +--ro comp-bit-width?                      uint8
|  |  |  |           |  +--ro comp-shift?                          uint8
|  |  |  |           +--:(beam-space-compression)
|  |  |  |           |  +--ro active-beam-space-coeficient-mask*   uint8
|  |  |  |           |  +--ro block-scaler?                        uint8
|  |  |  |           +--:(modulation-compression)
|  |  |  |              +--ro csf?                                 uint8
|  |  |  |              +--ro mod-comp-scaler?                     uint16
|  |  |  +--:(time)
|  |  |  |  +--ro time-domain-beams
|  |  |  |     +--ro max-number-of-beam-ids                     uint16
|  |  |  |     +--ro initial-beam-id                            uint16
|  |  |  |     +--ro frequency-granularity                      enumeration
|  |  |  |     +--ro time-granularity                           enumeration
|  |  |  |     +--ro iq-bitwidth?                               uint8
|  |  |  |     +--ro compression-type                           enumeration
|  |  |  |     x--ro bitwidth?                                  uint8
|  |  |  |     +--ro (compression-format)?
|  |  |  |     |  +--:(no-compresison)
|  |  |  |     |  +--:(block-floating-point)
|  |  |  |     |  |  +--ro exponent?                            uint8
|  |  |  |     |  +--:(block-scaling)
|  |  |  |     |  |  +--ro block-scalar?                        uint8
|  |  |  |     |  +--:(u-law)
|  |  |  |     |  |  +--ro comp-bit-width?                      uint8
|  |  |  |     |  |  +--ro comp-shift?                          uint8
|  |  |  |     |  +--:(beam-space-compression)
|  |  |  |     |  |  +--ro active-beam-space-coeficient-mask*   uint8
|  |  |  |     |  |  +--ro block-scaler?                        uint8
|  |  |  |     |  +--:(modulation-compression)
|  |  |  |     |     +--ro csf?                                 uint8
|  |  |  |     |     +--ro mod-comp-scaler?                     uint16
|  |  |  |     +--ro additional-compression-method-supported* []
|  |  |  |        +--ro iq-bitwidth?                               uint8
|  |  |  |        +--ro compression-type                           enumeration
|  |  |  |        x--ro bitwidth?                                  uint8
|  |  |  |        +--ro (compression-format)?
|  |  |  |           +--:(no-compresison)
|  |  |  |           +--:(block-floating-point)
|  |  |  |           |  +--ro exponent?                            uint8
```

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

```
|  |  |  |           +--:(block-scaling)
|  |  |  |           |  +--ro block-scalar?                        uint8
|  |  |  |           +--:(u-law)
|  |  |  |           |  +--ro comp-bit-width?                      uint8
|  |  |  |           |  +--ro comp-shift?                          uint8
|  |  |  |           +--:(beam-space-compression)
|  |  |  |           |  +--ro active-beam-space-coeficient-mask*   uint8
|  |  |  |           |  +--ro block-scaler?                        uint8
|  |  |  |           +--:(modulation-compression)
|  |  |  |              +--ro csf?                                 uint8
|  |  |  |              +--ro mod-comp-scaler?                     uint16
|  |  |  +--:(hybrid)
|  |  |     +--ro hybrid-beams
|  |  |        +--ro max-number-of-beam-ids                     uint16
|  |  |        +--ro initial-beam-id                            uint16
|  |  |        +--ro frequency-granularity                      enumeration
|  |  |        +--ro time-granularity                           enumeration
|  |  |        +--ro iq-bitwidth?                               uint8
|  |  |        +--ro compression-type                           enumeration
|  |  |        x--ro bitwidth?                                  uint8
|  |  |        +--ro (compression-format)?
|  |  |        |  +--:(no-compresison)
|  |  |        |  +--:(block-floating-point)
|  |  |        |  |  +--ro exponent?                            uint8
|  |  |        |  +--:(block-scaling)
|  |  |        |  |  +--ro block-scalar?                        uint8
|  |  |        |  +--:(u-law)
|  |  |        |  |  +--ro comp-bit-width?                      uint8
|  |  |        |  |  +--ro comp-shift?                          uint8
|  |  |        |  +--:(beam-space-compression)
|  |  |        |  |  +--ro active-beam-space-coeficient-mask*   uint8
|  |  |        |  |  +--ro block-scaler?                        uint8
|  |  |        |  +--:(modulation-compression)
|  |  |        |     +--ro csf?                                 uint8
|  |  |        |     +--ro mod-comp-scaler?                     uint16
|  |  |        +--ro additional-compression-method-supported* []
|  |  |           +--ro iq-bitwidth?                               uint8
|  |  |           +--ro compression-type                           enumeration
|  |  |           x--ro bitwidth?                                  uint8
|  |  |           +--ro (compression-format)?
|  |  |              +--:(no-compresison)
|  |  |              +--:(block-floating-point)
|  |  |              |  +--ro exponent?                            uint8
|  |  |              +--:(block-scaling)
|  |  |              |  +--ro block-scalar?                        uint8
|  |  |              +--:(u-law)
|  |  |              |  +--ro comp-bit-width?                      uint8
|  |  |              |  +--ro comp-shift?                          uint8
|  |  |              +--:(beam-space-compression)
|  |  |              |  +--ro active-beam-space-coeficient-mask*   uint8
|  |  |              |  +--ro block-scaler?                        uint8
|  |  |              +--:(modulation-compression)
|  |  |                 +--ro csf?                                 uint8
|  |  |                 +--ro mod-comp-scaler?                     uint16
|  |  +--ro number-of-beams?                uint16
|  +--ro beam-information
|     +--ro number-of-beamforming-properties?   uint16
|     +--ro beamforming-properties* [beam-id]
|        +--ro beam-id                 uint16
|        +--ro beamforming-property
|           +--ro beam-type?                   enumeration
|           +--ro beam-group-id?               uint16
|           x--ro coarse-fine-beam-relation*                    beam-reference
|           x--ro neighbour-beams*                              beam-reference
|           +--ro coarse-fine-beam-capability-based-relation*   beam-capabilities-reference
|           +--ro neighbour-beams-capability-based*             beam-capabilities-reference
+--ro capabilities-groups* [capabilities-group]
|  +--ro capabilities-group    uint16
|  +--ro band-number?          -> /mcap:module-capability/band-capabilities/band-number
|  +--ro tx-array*             -> /up:user-plane-configuration/tx-arrays/name
|  +--ro rx-array*             -> /up:user-plane-configuration/rx-arrays/name
|  +--ro static-properties
|  |  +--ro rt-bf-weights-update-support?   boolean
|  |  +--ro (beamforming-type)?
|  |  |  +--:(frequency)
```

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

```
        |  |  |  |  +--ro frequency-domain-beams
        |  |  |  |     +--ro max-number-of-beam-ids                     uint16
        |  |  |  |     +--ro initial-beam-id                            uint16
        |  |  |  |     +--ro iq-bitwidth?                               uint8
        |  |  |  |     +--ro compression-type                           enumeration
        |  |  |  |     x--ro bitwidth?                                  uint8
        |  |  |  |     +--ro (compression-format)?
        |  |  |  |     |  +--:(no-compresison)
        |  |  |  |     |  +--:(block-floating-point)
        |  |  |  |     |  |  +--ro exponent?                            uint8
        |  |  |  |     |  +--:(block-scaling)
        |  |  |  |     |  |  +--ro block-scalar?                        uint8
        |  |  |  |     |  +--:(u-law)
        |  |  |  |     |  |  +--ro comp-bit-width?                      uint8
        |  |  |  |     |  |  +--ro comp-shift?                          uint8
        |  |  |  |     |  +--:(beam-space-compression)
        |  |  |  |     |  |  +--ro active-beam-space-coeficient-mask*   uint8
        |  |  |  |     |  |  +--ro block-scaler?                        uint8
        |  |  |  |     |  +--:(modulation-compression)
        |  |  |  |     |     +--ro csf?                                 uint8
        |  |  |  |     |     +--ro mod-comp-scaler?                     uint16
        |  |  |  |     +--ro additional-compression-method-supported* []
        |  |  |  |        +--ro iq-bitwidth?                               uint8
        |  |  |  |        +--ro compression-type                           enumeration
        |  |  |  |        x--ro bitwidth?                                  uint8
        |  |  |  |        +--ro (compression-format)?
        |  |  |  |           +--:(no-compresison)
        |  |  |  |           +--:(block-floating-point)
        |  |  |  |           |  +--ro exponent?                            uint8
        |  |  |  |           +--:(block-scaling)
        |  |  |  |           |  +--ro block-scalar?                        uint8
        |  |  |  |           +--:(u-law)
        |  |  |  |           |  +--ro comp-bit-width?                      uint8
        |  |  |  |           |  +--ro comp-shift?                          uint8
        |  |  |  |           +--:(beam-space-compression)
        |  |  |  |           |  +--ro active-beam-space-coeficient-mask*   uint8
        |  |  |  |           |  +--ro block-scaler?                        uint8
        |  |  |  |           +--:(modulation-compression)
        |  |  |  |              +--ro csf?                                 uint8
        |  |  |  |              +--ro mod-comp-scaler?                     uint16
        |  |  |  +--:(time)
        |  |  |  |  +--ro time-domain-beams
        |  |  |  |     +--ro max-number-of-beam-ids                     uint16
        |  |  |  |     +--ro initial-beam-id                            uint16
        |  |  |  |     +--ro frequency-granularity                      enumeration
        |  |  |  |     +--ro time-granularity                           enumeration
        |  |  |  |     +--ro iq-bitwidth?                               uint8
        |  |  |  |     +--ro compression-type                           enumeration
        |  |  |  |     x--ro bitwidth?                                  uint8
        |  |  |  |     +--ro (compression-format)?
        |  |  |  |     |  +--:(no-compresison)
        |  |  |  |     |  +--:(block-floating-point)
        |  |  |  |     |  |  +--ro exponent?                            uint8
        |  |  |  |     |  +--:(block-scaling)
        |  |  |  |     |  |  +--ro block-scalar?                        uint8
        |  |  |  |     |  +--:(u-law)
        |  |  |  |     |  |  +--ro comp-bit-width?                      uint8
        |  |  |  |     |  |  +--ro comp-shift?                          uint8
        |  |  |  |     |  +--:(beam-space-compression)
        |  |  |  |     |  |  +--ro active-beam-space-coeficient-mask*   uint8
        |  |  |  |     |  |  +--ro block-scaler?                        uint8
        |  |  |  |     |  +--:(modulation-compression)
        |  |  |  |     |     +--ro csf?                                 uint8
        |  |  |  |     |     +--ro mod-comp-scaler?                     uint16
        |  |  |  |     +--ro additional-compression-method-supported* []
        |  |  |  |        +--ro iq-bitwidth?                               uint8
        |  |  |  |        +--ro compression-type                           enumeration
        |  |  |  |        x--ro bitwidth?                                  uint8
        |  |  |  |        +--ro (compression-format)?
        |  |  |  |           +--:(no-compresison)
        |  |  |  |           +--:(block-floating-point)
        |  |  |  |           |  +--ro exponent?                            uint8
        |  |  |  |           +--:(block-scaling)
        |  |  |  |           |  +--ro block-scalar?                        uint8
        |  |  |  |           +--:(u-law)
```

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

```
|  |  |  |           |  +--ro comp-bit-width?                      uint8
|  |  |  |           |  +--ro comp-shift?                          uint8
|  |  |  |           +--:(beam-space-compression)
|  |  |  |           |  +--ro active-beam-space-coeficient-mask*   uint8
|  |  |  |           |  +--ro block-scaler?                        uint8
|  |  |  |           +--:(modulation-compression)
|  |  |  |              +--ro csf?                                 uint8
|  |  |  |              +--ro mod-comp-scaler?                     uint16
|  |  |  +--:(hybrid)
|  |  |     +--ro hybrid-beams
|  |  |        +--ro max-number-of-beam-ids                     uint16
|  |  |        +--ro initial-beam-id                            uint16
|  |  |        +--ro frequency-granularity                      enumeration
|  |  |        +--ro time-granularity                           enumeration
|  |  |        +--ro iq-bitwidth?                               uint8
|  |  |        +--ro compression-type                           enumeration
|  |  |        x--ro bitwidth?                                  uint8
|  |  |        +--ro (compression-format)?
|  |  |        |  +--:(no-compresison)
|  |  |        |  +--:(block-floating-point)
|  |  |        |  |  +--ro exponent?                            uint8
|  |  |        |  +--:(block-scaling)
|  |  |        |  |  +--ro block-scalar?                        uint8
|  |  |        |  +--:(u-law)
|  |  |        |  |  +--ro comp-bit-width?                      uint8
|  |  |        |  |  +--ro comp-shift?                          uint8
|  |  |        |  +--:(beam-space-compression)
|  |  |        |  |  +--ro active-beam-space-coeficient-mask*   uint8
|  |  |        |  |  +--ro block-scaler?                        uint8
|  |  |        |  +--:(modulation-compression)
|  |  |        |     +--ro csf?                                 uint8
|  |  |        |     +--ro mod-comp-scaler?                     uint16
|  |  |        +--ro additional-compression-method-supported* []
|  |  |           +--ro iq-bitwidth?                               uint8
|  |  |           +--ro compression-type                           enumeration
|  |  |           x--ro bitwidth?                                  uint8
|  |  |           +--ro (compression-format)?
|  |  |              +--:(no-compresison)
|  |  |              +--:(block-floating-point)
|  |  |              |  +--ro exponent?                            uint8
|  |  |              +--:(block-scaling)
|  |  |              |  +--ro block-scalar?                        uint8
|  |  |              +--:(u-law)
|  |  |              |  +--ro comp-bit-width?                      uint8
|  |  |              |  +--ro comp-shift?                          uint8
|  |  |              +--:(beam-space-compression)
|  |  |              |  +--ro active-beam-space-coeficient-mask*   uint8
|  |  |              |  +--ro block-scaler?                        uint8
|  |  |              +--:(modulation-compression)
|  |  |                 +--ro csf?                                 uint8
|  |  |                 +--ro mod-comp-scaler?                     uint16
|  |  +--ro number-of-beams?                uint16
|  +--ro beam-information
|     +--ro number-of-beamforming-properties?   uint16
|     +--ro beamforming-properties* [beam-id]
|        +--ro beam-id                 uint16
|        +--ro beamforming-property
|           +--ro beam-type?                   enumeration
|           +--ro beam-group-id?               uint16
|           +--ro coarse-fine-beam-relation*   beam-reference
|           +--ro neighbour-beams*             beam-reference
+--ro ue-specific-beamforming!
|  +--ro max-number-of-ues?   uint8
+--ro operational-properties {MODIFY-BF-CONFIG}?
|  +--ro number-of-writeable-beamforming-files    uint8
|  +--ro update-bf-non-delete?                    boolean
|  +--ro persistent-bf-files?                     boolean
+--ro beamforming-trough-attributes-supported?        boolean
+--ro beamforming-trough-ue-channel-info-supported?   boolean
+--ro beam-tilt {BEAM-TILT}?
   +--ro predefined-beam-tilt-offset-information* [band-number]
   |  +--ro band-number                         -> /mcap:module-capability/band-capabilities/band-number
   |  +--ro elevation-tilt-granularity          uint8
   |  +--ro azimuth-tilt-granularity            uint8
   |  +--ro minimum-supported-elevation-tilt    int16
```

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

```
        |  +--ro maximum-supported-elevation-tilt    int16
        |  +--ro minimum-supported-azimuth-tilt      int16
        |  +--ro maximum-supported-azimuth-tilt      int16
        |  +--ro run-time-tilt-offset-supported      boolean
        +--ro predefined-beam-tilt-state* [band-number]
           +--ro band-number                      -> /mcap:module-capability/band-capabilities/band-number
           +--ro offset-elevation-tilt-angle    int16
           +--ro offset-azimuth-tilt-angle      int16

  rpcs:
    +---x activate-beamforming-config {MODIFY-BF-CONFIG}?
    |  +---w input
    |  |  +---w beamforming-config-file    string
    |  |  +---w band-number?              -> /mcap:module-capability/band-capabilities/band-number
    |  +--ro output
    |     +--ro status           enumeration
    |     +--ro error-message?   string
    +---x activate-beamforming-config-by-capability-group {MODIFY-BF-CONFIG}?
    |  +---w input
    |  |  +---w beamforming-config-file    string
    |  |  +---w capabilities-group         -> /beamforming-config/capabilities-groups/capabilities-group
    |  +--ro output
    |     +--ro status           enumeration
    |     +--ro error-message?   string
    +---x modify-predefined-beam-tilt-offset {BEAM-TILT}?
       +---w input
       |  +---w predefined-beam-tilt-offset* [band-number] {BEAM-TILT}?
       |     +---w band-number                    -> /mcap:module-capability/band-capabilities/band-number
       |     +---w offset-elevation-tilt-angle?   int16
       |     +---w offset-azimuth-tilt-angle?     int16
       +--ro output
          +--ro status           enumeration
          +--ro error-message?   string

  notifications:
    +---n beamforming-information-update
    |  +--ro band-number?   -> /mcap:module-capability/band-capabilities/band-number
    +---n capability-group-beamforming-information-update
    |  +--ro capabilities-group    -> /beamforming-config/capabilities-groups/capabilities-group
    +---n predefined-beam-tilt-offset-complete {BEAM-TILT}?
       +--ro predefined-beam-tilt-state* [band-number]
          +--ro band-number                    -> /mcap:module-capability/band-capabilities/band-number
          +--ro offset-elevation-tilt-angle    int16
          +--ro offset-azimuth-tilt-angle      int16
```

D.5.4 o-ran-laa.yang Module

The format for the LAA module is provided below

```
module: o-ran-laa
  +--rw laa-config
     +--rw number-of-laa-scells?                    uint8
     +--rw multi-carrier-type?                      enumeration
     +--rw multi-carrier-tx?                        boolean
     +--rw multi-carrier-freeze?                    boolean
     +--rw laa-ending-dwpts-supported?              boolean
     +--rw laa-starting-in-second-slot-supported?   boolean
```

D.5.6 o-ran-antenna-calibration.yang Module

The format for the antenna calibration module is provided below

```
module: o-ran-antenna-calibration
  +--rw antenna-calibration
     +--ro antenna-calibration-capabilities
     |  +--ro self-calibration-support?                     Boolean
     |  +--ro coordinated-calibration-support?              boolean {O-RU-COORDINATED-ANT-CAL}?
     |  +--ro number-of-calibration-symbols-per-block-dl    uint8
     |  +--ro number-of-calibration-symbols-per-block-ul    uint8
     |  +--ro interval-between-calibration-blocks?          uint8
     |  +--ro number-of-calibration-blocks-per-step-dl      uint8
```

```
     |  +--ro number-of-calibration-blocks-per-step-ul      uint8
     |  +--ro interval-between-calibration-steps?           uint8
     |  +--ro number-of-calibration-steps                   uint8
     |  +--ro calibration-period?                           uint16 {O-RU-COORDINATED-ANT-CAL}?
     +--rw self-calibration-policy
        +--rw self-calibration-allowed?   Boolean
        +--rw coordinated-calibration-allowed?   boolean {O-RU-COORDINATED-ANT-CAL}?

  rpcs:
    +---x start-antenna-calibration
       +---w input
       |  +---w symbol-bitmask-dl          string
       |  +---w symbol-bitmask-ul          string
       |  +---w slot-bitmask-dl            string
       |  +---w slot-bitmask-ul            string
       |  +---w frame-bitmask-dl           string
       |  +---w frame-bitmask-ul           string
       |  +---w calibration-step-size      uint8
       |  +---w calibration-step-number    uint8
       |  +---w start-sfn                  uint16
       +--ro output
          +--ro status          enumeration
          +--ro error-message?   string

  notifications:
    +---n antenna-calibration-required
    |  +--ro dl-calibration-frequency-chunk* []
    |  |  +--ro start-calibration-frequency-dl?   uint64
    |  |  +--ro end-calibration-frequency-dl?     uint64
    |  +--ro ul-calibration-frequency-chunk* []
    |     +--ro start-calibration-frequency-ul?   uint64
    |     +--ro end-calibration-frequency-ul?     uint64
    +---n antenna-calibration-coordinated {O-RU-COORDINATED-ANT-CAL}?
    |  +--ro dl-calibration-frequency-chunk* []
    |  |  +--ro start-calibration-frequency-dl?   uint64
    |  |  +--ro end-calibration-frequency-dl?     uint64
    |  +--ro ul-calibration-frequency-chunk* []
    |  |  +--ro start-calibration-frequency-ul?   uint64
    |  |  +--ro end-calibration-frequency-ul?     uint64
    |  +--ro symbol-bitmask-dl                 string
    |  +--ro symbol-bitmask-ul                 string
    |  +--ro slot-bitmask-dl                   string
    |  +--ro slot-bitmask-ul                   string
    |  +--ro frame-bitmask-dl                  string
    |  +--ro frame-bitmask-ul                  string
    |  +--ro calibration-step-size             uint8
    |  +--ro calibration-step-number           uint8
    |  +--ro start-sfn                         uint16
    +---n antenna-calibration-result
       +--ro status             enumeration
       +--ro detailed-reason?   String
```

D.5.7 o-ran-shared-cell.yang Module

The format for the module o-ran shared cell is provided below.

```
module: o-ran-shared-cell
  +--rw shared-cell
     +--ro shared-cell-module-cap
     |  +--ro t-copy                            uint32
     |  +--ro t-combine                         uint32
     |  +--ro ta3-prime-max-upper-range         uint32
     |  +--ro max-number-node-copy-and-combine    uint8
     |  +--ro max-number-eaxcid-copy            uint8
     |  +--ro max-number-eaxcid-combine         uint8
     |  +--ro compression-method-supported* [] {FHM}?
     |     +--ro iq-bitwidth?                             uint8
     |     +--ro compression-type                         enumeration
     |     x--ro bitwidth?                                uint8
     |     +--ro (compression-format)?
     |        +--:(no-compresison)
     |        +--:(block-floating-point)
     |        |  +--ro exponent?                          uint8
     |        +--:(block-floating-point-selective-re-sending)
```

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

```
     |         |  +--ro sres-exponent?                      uint8
     |         +--:(block-scaling)
     |         |  +--ro block-scalar?                       uint8
     |         +--:(u-law)
     |         |  +--ro comp-bit-width?                     uint8
     |         |  +--ro comp-shift?                         uint8
     |         +--:(beam-space-compression)
     |         |  +--ro active-beam-space-coeficient-mask*   uint8
     |         |  +--ro block-scaler?                       uint8
     |         +--:(modulation-compression)
     |         |  +--ro csf?                                uint8
     |         |  +--ro mod-comp-scaler?                    uint16
     |         +--:(modulation-compression-selective-re-sending)
     |            +--ro sres-csf?                           uint8
     |            +--ro sres-mod-comp-scaler?               uint16
     +--rw shared-cell-config
        +--rw (shared-cell-copy-combine-mode)?
           +--:(COMMON)
           |  +--rw shared-cell-copy-entities* [name]
           |  |  +--rw name                              string
           |  |  +--rw north-node-processing-element?    -> /o-ran-pe:processing-elements/ru-elements/name
           |  |  +--rw south-node-processing-elements*   -> /o-ran-pe:processing-elements/ru-elements/name
           |  |  +--rw shared-cell-copy-uplane-config {FHM}?
           |  |     +--rw tx-eaxc-id* [eaxc-id]
           |  |     |  +--rw eaxc-id    uint16
           |  |     +--rw rx-eaxc-id* [eaxc-id]
           |  |     |  +--rw eaxc-id    uint16
           |  |     +--rw downlink-radio-frame-offset    uint32
           |  |     +--rw downlink-sfn-offset            int16
           |  +--rw shared-cell-combine-entities* [name]
           |     +--rw name                              string
           |     +--rw north-node-processing-element?       -> /o-ran-pe:processing-elements/ru-elements/name
           |     +--rw south-node-processing-elements*      -> /o-ran-pe:processing-elements/ru-elements/name
           |     +--rw ta3-prime-max?                       uint32
           |     +--rw shared-cell-combine-uplane-config {FHM}?
           |        +--rw rx-eaxc-id* [eaxc-id]
           |        |  +--rw eaxc-id              uint16
           |        |  +--rw comression-method
           |        |     +--rw iq-bitwidth?                           uint8
           |        |     +--rw compression-type                       enumeration
           |        |     x--rw bitwidth?                              uint8
           |        |     +--rw (compression-format)?
           |        |        +--:(no-compresison)
           |        |        +--:(block-floating-point)
           |        |        |  +--rw exponent?                        uint8
           |        |        +--:(block-floating-point-selective-re-sending)
           |        |        |  +--rw sres-exponent?                   uint8
           |        |        +--:(block-scaling)
           |        |        |  +--rw block-scalar?                    uint8
           |        |        +--:(u-law)
           |        |        |  +--rw comp-bit-width?                  uint8
           |        |        |  +--rw comp-shift?                      uint8
           |        |        +--:(beam-space-compression)
           |        |        |  +--rw active-beam-space-coeficient-mask*   uint8
           |        |        |  +--rw block-scaler?                    uint8
           |        |        +--:(modulation-compression)
           |        |        |  +--rw csf?                             uint8
           |        |        |  +--rw mod-comp-scaler?                 uint16
           |        |        +--:(modulation-compression-selective-re-sending)
           |        |           +--rw sres-csf?                        uint8
           |        |           +--rw sres-mod-comp-scaler?            uint16
           |        +--rw downlink-radio-frame-offset    uint32
           |        +--rw downlink-sfn-offset            int16
           |        +--rw n-ta-offset                    uint32
           |        +--rw number-of-prb                  uint16
           +--:(SELECTIVE-BEAM-ID) {FHM and SELECTIVE-BEAM-ID}?
           |  +--rw shared-cell-copy-entities-selective-beam-id* [name]
           |  |  +--rw name                                 string
           |  |  +--rw north-node-processing-element?        -> /o-ran-pe:processing-elements/ru-
elements/name
           |  |  +--rw south-node-processing-elements*       -> /o-ran-pe:processing-elements/ru-
elements/name
           |  |  +--rw mapping-table-for-selective-beam-id* [global-beam-id south-node-processing-elements]
           |  |  |  +--rw global-beam-id                    uint16
           |  |  |  +--rw south-node-processing-elements    -> /o-ran-pe:processing-elements/ru-elements/name
```

```
         |  |  |  +--rw local-beam-id?                     uint16
         |  |  +--rw shared-cell-copy-uplane-config {FHM}?
         |  |     +--rw tx-eaxc-id* [eaxc-id]
         |  |     |  +--rw eaxc-id    uint16
         |  |     +--rw rx-eaxc-id* [eaxc-id]
         |  |     |  +--rw eaxc-id    uint16
         |  |     +--rw downlink-radio-frame-offset    uint32
         |  |     +--rw downlink-sfn-offset           int16
         |  +--rw shared-cell-combine-entities-for-selective-beam-id* [name]
         |     +--rw name                              string
         |     +--rw north-node-processing-element?       -> /o-ran-pe:processing-elements/ru-elements/name
         |     +--rw south-node-processing-elements*      -> /o-ran-pe:processing-elements/ru-elements/name
         |     +--rw ta3-prime-max?                    uint32
         |     +--rw shared-cell-combine-uplane-config {FHM}?
         |        +--rw rx-eaxc-id* [eaxc-id]
         |        |  +--rw eaxc-id              uint16
         |        |  +--rw comression-method
         |        |     +--rw iq-bitwidth?                            uint8
         |        |     +--rw compression-type                        enumeration
         |        |     x--rw bitwidth?                               uint8
         |        |     +--rw (compression-format)?
         |        |        +--:(no-compresison)
         |        |        +--:(block-floating-point)
         |        |        |  +--rw exponent?                         uint8
         |        |        +--:(block-floating-point-selective-re-sending)
         |        |        |  +--rw sres-exponent?                    uint8
         |        |        +--:(block-scaling)
         |        |        |  +--rw block-scalar?                     uint8
         |        |        +--:(u-law)
         |        |        |  +--rw comp-bit-width?                   uint8
         |        |        |  +--rw comp-shift?                       uint8
         |        |        +--:(beam-space-compression)
         |        |        |  +--rw active-beam-space-coeficient-mask*   uint8
         |        |        |  +--rw block-scaler?                     uint8
         |        |        +--:(modulation-compression)
         |        |        |  +--rw csf?                              uint8
         |        |        |  +--rw mod-comp-scaler?                  uint16
         |        |        +--:(modulation-compression-selective-re-sending)
         |        |           +--rw sres-csf?                         uint8
         |        |           +--rw sres-mod-comp-scaler?             uint16
         |        +--rw downlink-radio-frame-offset    uint32
         |        +--rw downlink-sfn-offset           int16
         |        +--rw n-ta-offset                   uint32
         |        +--rw number-of-prb                 uint16
         +--:(SELECTIVE)
```

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

Annex ZZZ O-RAN Adopter License Agreement

BY DOWNLOADING, USING OR OTHERWISE ACCESSING ANY O-RAN SPECIFICATION, ADOPTER AGREES TO THE TERMS OF THIS AGREEMENT.

This O-RAN Adopter License Agreement (the "Agreement") is made by and between the O-RAN Alliance and the entity that downloads, uses or otherwise accesses any O-RAN Specification, including its Affiliates (the "Adopter").

This is a license agreement for entities who wish to adopt any O-RAN Specification.

Section 1: DEFINITIONS

1.1 "Affiliate" means an entity that directly or indirectly controls, is controlled by, or is under common control with another entity, so long as such control exists. For the purpose of this Section, "Control" means beneficial ownership of fifty (50%) percent or more of the voting stock or equity in an entity.

1.2 "Compliant Implementation" means any system, device, method or operation (whether implemented in hardware, software or combinations thereof) that fully conforms to a Final Specification.

1.3 "Adopter(s)" means all entities, who are not Members, Contributors or Academic Contributors, including their Affiliates, who wish to download, use or otherwise access O-RAN Specifications.

1.4 "Minor Update" means an update or revision to an O-RAN Specification published by O-RAN Alliance that does not add any significant new features or functionality and remains interoperable with the prior version of an O-RAN Specification. The term "O-RAN Specifications" includes Minor Updates.

1.5 "Necessary Claims" means those claims of all present and future patents and patent applications, other than design patents and design registrations, throughout the world, which (i) are owned or otherwise licensable by a Member, Contributor or Academic Contributor during the term of its Member, Contributor or Academic Contributorship; (ii) such Member, Contributor or Academic Contributor has the right to grant a license without the payment of consideration to a third party; and (iii) are necessarily infringed by a Compliant Implementation (without considering any Contributions not included in the Final Specification). A claim is necessarily infringed only when it is not possible on technical (but not commercial) grounds, taking into account normal technical practice and the state of the art generally available at the date any Final Specification was published by the O-RAN Alliance or the date the patent claim first came into existence, whichever last occurred, to make, sell, lease, otherwise dispose of, repair, use or operate a Compliant Implementation without infringing that claim. For the avoidance of doubt in exceptional cases where a Final Specification can only be implemented by technical solutions, all of which infringe patent claims, all such patent claims shall be considered Necessary Claims.

1.6 "Defensive Suspension" means for the purposes of any license grant pursuant to Section 3, Member, Contributor, Academic Contributor, Adopter, or any of their Affiliates, may have the discretion to include in their license a term allowing the licensor to suspend the license against a licensee who brings a patent infringement suit against the licensing Member, Contributor, Academic Contributor, Adopter, or any of their Affiliates.

Section 2: COPYRIGHT LICENSE

2.1 Subject to the terms and conditions of this Agreement, O-RAN Alliance hereby grants to Adopter a nonexclusive, nontransferable, irrevocable, non-sublicensable, worldwide copyright license to obtain, use and modify O-RAN Specifications, but not to further distribute such O-RAN Specification in any modified or unmodified way, solely in furtherance of implementations of an ORAN Specification.

2.2 Adopter shall not use O-RAN Specifications except as expressly set forth in this Agreement or in a separate written agreement with O-RAN Alliance.

Section 3: FRAND LICENSE

3.1 Members, Contributors and Academic Contributors and their Affiliates are prepared to grant based on a separate Patent License Agreement to each Adopter under Fair Reasonable And Non- Discriminatory (FRAND) terms and conditions with or without compensation (royalties) a nonexclusive, non-transferable, irrevocable (but subject to

Defensive Suspension), non-sublicensable, worldwide patent license under their Necessary Claims to make, have made, use, import, offer to sell, lease, sell and otherwise distribute Compliant Implementations; provided, however, that such license shall not extend: (a) to any part or function of a product in which a Compliant Implementation is incorporated that is not itself part of the Compliant Implementation; or (b) to any Adopter if that Adopter is not making a reciprocal grant to Members, Contributors and Academic Contributors, as set forth in Section 3.3. For the avoidance of doubt, the foregoing licensing commitment includes the distribution by the Adopter's distributors and the use by the Adopter's customers of such licensed Compliant Implementations.

3.2 Notwithstanding the above, if any Member, Contributor or Academic Contributor, Adopter or their Affiliates has reserved the right to charge a FRAND royalty or other fee for its license of Necessary Claims to Adopter, then Adopter is entitled to charge a FRAND royalty or other fee to such Member, Contributor or Academic Contributor, Adopter and its Affiliates for its license of Necessary Claims to its licensees.

3.3 Adopter, on behalf of itself and its Affiliates, shall be prepared to grant based on a separate Patent License Agreement to each Members, Contributors, Academic Contributors, Adopters and their Affiliates under Fair Reasonable And Non-Discriminatory (FRAND) terms and conditions with or without compensation (royalties) a nonexclusive, non-transferable, irrevocable (but subject to Defensive Suspension), non-sublicensable, worldwide patent license under their Necessary Claims to make, have made, use, import, offer to sell, lease, sell and otherwise distribute Compliant Implementations; provided, however, that such license will not extend: (a) to any part or function of a product in which a Compliant Implementation is incorporated that is not itself part of the Compliant Implementation; or (b) to any Members, Contributors, Academic Contributors, Adopters and their Affiliates that is not making a reciprocal grant to Adopter, as set forth in Section 3.1. For the avoidance of doubt, the foregoing licensing commitment includes the distribution by the Members', Contributors', Academic Contributors', Adopters' and their Affiliates' distributors and the use by the Members', Contributors', Academic Contributors', Adopters' and their Affiliates' customers of such licensed Compliant Implementations.

Section 4: TERM AND TERMINATION

4.1 This Agreement shall remain in force, unless early terminated according to this Section 4.

4.2 O-RAN Alliance on behalf of its Members, Contributors and Academic Contributors may terminate this Agreement if Adopter materially breaches this Agreement and does not cure or is not capable of curing such breach within thirty (30) days after being given notice specifying the breach.

4.3 Sections 1, 3, 5 - 11 of this Agreement shall survive any termination of this Agreement. Under surviving Section 3, after termination of this Agreement, Adopter will continue to grant licenses (a) to entities who become Adopters after the date of termination; and (b) for future versions of ORAN Specifications that are backwards compatible with the version that was current as of the date of termination.

Section 5: CONFIDENTIALITY

Adopter will use the same care and discretion to avoid disclosure, publication, and dissemination of O-RAN Specifications to third parties, as Adopter employs with its own confidential information, but no less than reasonable care. Any disclosure by Adopter to its Affiliates, contractors and consultants should be subject to an obligation of confidentiality at least as restrictive as those contained in this Section. The foregoing obligation shall not apply to any information which is: (1) rightfully known by Adopter without any limitation on use or disclosure prior to disclosure; (2) publicly available through no fault of Adopter; (3) rightfully received without a duty of confidentiality; (4) disclosed by O-RAN Alliance or a Member, Contributor or Academic Contributor to a third party without a duty of confidentiality on such third party; (5) independently developed by Adopter; (6) disclosed pursuant to the order of a court or other authorized governmental body, or as required by law, provided that Adopter provides reasonable prior written notice to O-RAN Alliance, and cooperates with O-RAN Alliance and/or the applicable Member, Contributor or Academic Contributor to have the opportunity to oppose any such order; or (7) disclosed by Adopter with O-RAN Alliance's prior written approval.

Section 6: INDEMNIFICATION

Adopter shall indemnify, defend, and hold harmless the O-RAN Alliance, its Members, Contributors or Academic Contributors, and their employees, and agents and their respective successors, heirs and assigns (the "Indemnitees"), against any liability, damage, loss, or expense (including reasonable attorneys' fees and expenses) incurred by or

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

imposed upon any of the Indemnitees in connection with any claims, suits, investigations, actions, demands or judgments arising out of Adopter's use of the licensed O-RAN Specifications or Adopter's commercialization of products that comply with O-RAN Specifications.

Section 7: LIMITATIONS ON LIABILITY; NO WARRANTY

EXCEPT FOR BREACH OF CONFIDENTIALITY, ADOPTER'S BREACH OF SECTION 3, AND ADOPTER'S INDEMNIFICATION OBLIGATIONS, IN NO EVENT SHALL ANY PARTY BE LIABLE TO ANY OTHER PARTY OR THIRD PARTY FOR ANY INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES RESULTING FROM ITS PERFORMANCE OR NON-PERFORMANCE UNDER THIS AGREEMENT, IN EACH CASE WHETHER UNDER CONTRACT, TORT, WARRANTY, OR OTHERWISE, AND WHETHER OR NOT SUCH PARTY HAD ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

O-RAN SPECIFICATIONS ARE PROVIDED "AS IS" WITH NO WARRANTIES OR CONDITIONS WHATSOEVER, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE. THE O-RAN ALLIANCE AND THE MEMBERS, CONTRIBUTORS OR ACADEMIC CONTRIBUTORS EXPRESSLY DISCLAIM ANY WARRANTY OR CONDITION OF MERCHANTABILITY, SECURITY, SATISFACTORY QUALITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, ERROR-FREE OPERATION, OR ANY WARRANTY OR CONDITION FOR O-RAN SPECIFICATIONS.

Section 8: ASSIGNMENT

Adopter may not assign the Agreement or any of its rights or obligations under this Agreement or make any grants or other sublicenses to this Agreement, except as expressly authorized hereunder, without having first received the prior, written consent of the O-RAN Alliance, which consent may be withheld in O-RAN Alliance's sole discretion. O-RAN Alliance may freely assign this Agreement.

Section 9: THIRD-PARTY BENEFICIARY RIGHTS

Adopter acknowledges and agrees that Members, Contributors and Academic Contributors (including future Members, Contributors and Academic Contributors) are entitled to rights as a third-party beneficiary under this Agreement, including as licensees under Section 3.

Section 10: BINDING ON AFFILIATES

Execution of this Agreement by Adopter in its capacity as a legal entity or association constitutes that legal entity's or association's agreement that its Affiliates are likewise bound to the obligations that are applicable to Adopter hereunder and are also entitled to the benefits of the rights of Adopter hereunder.

Section 11: GENERAL

This Agreement is governed by the laws of Germany without regard to its conflict or choice of law provisions.

This Agreement constitutes the entire agreement between the parties as to its express subject matter and expressly supersedes and replaces any prior or contemporaneous agreements between the parties, whether written or oral, relating to the subject matter of this Agreement.

Adopter, on behalf of itself and its Affiliates, agrees to comply at all times with all applicable laws, rules and regulations with respect to its and its Affiliates' performance under this Agreement, including without limitation, export control and antitrust laws. Without limiting the generality of the foregoing, Adopter acknowledges that this Agreement prohibits any communication that would violate the antitrust laws.

By execution hereof, no form of any partnership, joint venture or other special relationship is created between Adopter, or O-RAN Alliance or its Members, Contributors or Academic Contributors. Except as expressly set forth in this Agreement, no party is authorized to make any commitment on behalf of Adopter, or O-RAN Alliance or its Members, Contributors or Academic Contributors.

Copyright © 2021 by the O-RAN Alliance e.V.
Your use is subject to the terms of the O-RAN Adopter License Agreement in Annex ZZZ

In the event that any provision of this Agreement conflicts with governing law or if any provision is held to be null, void or otherwise ineffective or invalid by a court of competent jurisdiction, (i) such provisions will be deemed stricken from the contract, and (ii) the remaining terms, provisions, covenants and restrictions of this Agreement will remain in full force and effect.

Any failure by a party or third party beneficiary to insist upon or enforce performance by another party of any of the provisions of this Agreement or to exercise any rights or remedies under this Agreement or otherwise by law shall not be construed as a waiver or relinquishment to any extent of the other parties' or third party beneficiary's right to assert or rely upon any such provision, right or remedy in that or any other instance; rather the same shall be and remain in full force and effect.
