ETSI TS 104 023 V12.01 (2024-05)

Publicly Available Specification (PAS); O-RAN Fronthaul Management Plane Specification v12.01; (O-RAN.WG4.MP.0-R003-v12.01)

Reference
DTS/MSG-001143

Keywords
fronthaul, management plane, PAS

ETSI

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

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

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Contents 3
Intellectual Property Rights 10
Foreword....10
Modal verbs terminology....10
Executive summary 10
1 Scope 11
2 References 11
2.1 Normative references 11
2.2 Informative references....14
3 Definition of terms, symbols and abbreviations....14
3.1 Terms....14
3.2 Abbreviations 16
4 General 17
4.1 Conventions....17
4.2 Void....17
4.3 Revision and compatibility handling....17
4.4 Namespace compatibility handling 18
5 High level description 18
5.1 Top level functional description, terminology, including hybrid, hierarchical 18
5.1.1 Architecture for O-RAN WG4 fronthaul functional split 18
5.1.2 M-Plane architecture model....19
5.1.3 Transport network....20
5.1.4 M-Plane functional description....20
5.2 Interfaces 21
5.3 YANG module introduction....22
5.4 Security 22
6 "Start-up" installation....24
6.1 General 24
6.2 Management plane transport aspects....27
6.2.1 Transport establishment....27
6.2.2 O-RU identification in DHCP....30
6.2.3 Management plane VLAN discovery aspects....31
6.2.4 O-RU management plane IP address assignment 31
6.2.5 O-RU controller discovery....32
6.2.6 Certificate handling....33
6.2.6.0 Trust anchor provisioning 33
6.2.6.1 Certificate enrolment....34
6.2.6.2 CMPv2 based certificate enrolment 34
6.2.6.3 Operation with vendor-signed certificates 35
6.2.7 Event-Collector discovery 35
6.3 NETCONF call home to O-RU controller(s)....36
6.4 NETCONF connection establishment 39
6.4.0 General....39
6.4.1 NETCONF security 39
6.4.2 NETCONF authentication 39
6.4.3 User account provisioning 40
6.4.3.1 General....40
6.4.3.2 Certificates to NETCONF usernames mapping list provisioning 41
6.5 NETCONF access control 41
6.6 NETCONF capability discovery 43
6.7 Monitoring NETCONF connectivity....44
6.8 Closing a NETCONF session....47

6.9 PNF registration 47
6.9.1 Introduction....47
6.9.2 PNF registration procedure 47
6.9.3 Encoding of PNF registration notification 48

7 O-RU to O-DU interface management....49
7.1 O-RU interfaces....49
7.2 Transceiver 50
7.3 C/U-Plane VLAN configuration....50
7.4 O-RU C/U-Plane IP address assignment 50
7.5 Definition of processing elements 51
7.6 O-DU verification of C/U-Plane transport connectivity....52
7.6.1 C/U-Plane transport connectivity verification 52
7.6.2 Ethernet connectivity monitoring procedure....53
7.6.2.1 Transport connectivity verification monitoring Procedure 53
7.6.2.2 Validating the transport configuration 53
7.6.2.3 Monitor network connectivity....53
7.6.2.4 Managing ethernet connectivity monitoring procedure....54
7.6.3 IP connectivity monitoring procedure....54
7.6.3.1 Monitoring procedure....54
7.6.3.2 Managing IP connectivity monitoring procedure....54
7.7 C/U-Plane delay management 54
7.7.1 Introduction....54
7.7.2 Delay parameters 54
7.7.3 Reception window monitoring....55
7.7.4 External antenna delay control....55
7.8 O-RU adaptive delay capability 56
7.9 Measuring transport delay parameters....56
7.10 O-RU monitoring of C/U-Plane connectivity....57
7.11 Bandwidth management 57
7.12 IEEE 802.1X port based access control....57
7.12.1 Configuring port based access control....57
7.12.2 EAP authentication 58
7.12.3 Certificate time validation 58

8 Software management 59
8.1 General 59
8.1.1 Introduction....59
8.1.2 Software slots....59
8.1.3 Software management procedures 59
8.2 Software build 59
8.3 Software package 60
8.3.1 Software package name 60
8.3.2 File server storage of software package....60
8.3.3 Software package content and manifest.xml file 60
8.4 Software inventory 61
8.5 Download 63
8.6 Install....65
8.7 Bringing software into operation....67
8.7.1 Procedure 67
8.7.2 Software activation 67
8.7.3 Reset 69
8.8 Software update scenario....69
8.9 Factory reset 70

9 Configuration management 70
9.1 Baseline configuration....70
9.1.1 NETCONF operations 70
9.1.2 Retrieve state 70
9.1.3 Modify state....71
9.1.4 Retrieve parameters 74
9.1.5 Modify parameters....75
9.1.6 Deleting parameters 76

9.1.7 Notification framework....78
9.1.7.1 General....78
9.1.7.2 Event streams....78
9.1.7.3 ....Event stream discovery and subscription....78
9.2 Framework for optional feature handling....78
9.3 M-Plane operational state....79
9.4 Notification of updates to configuration datastore....79
9.4.1 Introduction....79
9.4.2 Subscribing to updates from an O-RU....79
9.5 Resetting O-RU....80
9.5.1 O-RU reset procedure....80
9.5.2 O-RU controller triggered O-RU reset operation....80
9.5.3 O-RU controller triggered O-RU reset procedure....80

10 Performance management....81
10.1 General....81
10.2 Measurement activation and de-activation....81
10.3 Collection and reporting of measurement result....82
10.3.0 Introduction....82
10.3.1 NETCONF process for dynamic subscriptions....83
10.3.2 File management process....84
10.3.3 Configured subscription process....86

11 Fault management....87
11.1 Introduction....87
11.2 Alarm notification....87
11.3 Manage alarms request to NETCONF clients....88
11.4 Fault sources....90
11.5 Manage alarms request to event-collector....91
11.6 Historical Alarms List....91

12 File management....91
12.1 Introduction....91
12.2 File system structure....92
12.3 File management operation: upload....93
12.4 File management operation: retrieve file list....94
12.5 File management operation: download....95

13 Synchronization aspects....96
13.1 Introduction....96
13.2 Sync status object....97
13.3 Sync capability object....98
13.4 PTP configuration....99
13.4.1 Introduction....99
13.4.2 G.8275.1 specific parameters....100
13.4.3 G.8275.2 specific parameters....100
13.5 PTP status....100
13.6 PLFS/SyncE configuration....101
13.7 PLFS/SyncE status....101
13.8 GNSS configuration....102
13.9 GNSS status....103

14 Operations use cases....103
14.1 Supervision failure handling and supervision termination handling....103
14.1.1 Supervision failure handling....103
14.1.2 Supervision termination handling....104
14.2 Log management....104
14.2.1 Introduction....104
14.2.2 Troubleshooting log management....105
14.3 Trace....109
14.4 Operational aspects of antenna line devices....111
14.4.1 Introduction....111
14.4.2 HDLC interworking....111

14.4.3 ALD operations ...112
14.5 Operational aspects of external IO ...114
14.5.1 Introduction...114
14.5.2 External input...114
14.5.3 External output...115
14.6 O-RU connectors...117
14.6.1 Introduction to O-RU connectors...117
14.6.2 Connectors related to antennas and antenna arrays...117
14.6.2.1 Antenna connectors...117
14.6.2.2 ...Distinguishable types of antenna connectors ...117
14.6.2.2.1 Feeder...117
14.6.2.2.2 Beamforming calibration...118

15 Details of O-RU operations...118
15.1 Retrieval of O-RU information ...118
15.2 User plane message routing...119
15.2.1 Introduction...119
15.2.2 Configurable format for eAxC_ID...119
15.2.3 U-Plane endpoint addressing ...120
15.2.4 General configuration scenario...120
15.3 Carrier configuration...123
15.3.1 Carrier creation ...123
15.3.2 Activation, deactivation and sleep ...124
15.3.3 Carrier state's relationship to synchronisation state...125
15.3.3.1 Synchronisation state and carrier state transitions...125
15.3.3.2 Synchronization lost and HOLDOVER mode expired ...126
15.3.3.3 External timing source restored...127
15.4 Beamforming...130
15.4.1 Beamforming configuration...130
15.4.2 Pre-defined beamforming configuration...130
15.4.3 Beamforming configuration update ...130
15.4.4 Tilting pre-defined beams...134
15.4.5 Dynamic beamforming control option...136
15.5 Antenna calibration ...136
15.5.1 Background...136
15.5.2 Overall operation ...137
15.5.2.1 General...137
15.5.2.2 Initiation...137
15.5.2.3 Self-calibration operation...138
15.5.2.4 Calibration completion...138
15.5.2.5 Antenna calibration procedure ...138
15.5.3 O-RU antenna calibration capability parameter configuration ...140
15.5.4 antenna-calibration-required notification parameters ...140
15.5.5 Start-antenna-calibration RPC request parameters ...141
15.5.6 Example antenna calibration operation...141
15.5.7 Calibration with multiple timing resource sets ...144
15.5.8 antenna-calibration-multiple-time-resource-params notification parameters ...144
15.6 Static configuration for PRACH and SRS...145
15.6.1 Background...145
15.6.2 Static configuration for PRACH processing...145
15.6.3 Frequency domain configuration ...145
15.6.4 Time domain configuration...146
15.6.5 Operation ...147
15.6.6 Static configuration for raw SRS processing ...148
15.6.7 Operation ...149
15.7 TDD pattern configuration ...149
15.8 C-Plane message limits ...150
15.9 Advanced endpoint capability report...150
15.10 U-Plane message limits ...155

16 Licensed assisted access...155

16.1 Introduction: 155
16.2 LAA-initiation process 156
16.2.1 LAA module capabilities 156
16.2.2 LAA O-RU parameter configuration 157
16.3 Carrier selection 158
16.3.1 LAA measurements 158
16.3.2 LAA carrier frequency configuration 158

17 Shared cell 158
17.1 Introduction 158
17.2 Architecture 159
17.3 Start-up and Installation 160
17.4 Performance management 164
17.5 Delay management 165
17.6 Details of O-RU operations for shared cell 166
17.6.1 O-RU information for shared cell 166
17.6.2 Topology discovery procedure 168
17.6.3 Shared cell configuration 169
17.6.4 U-plane configuration for FHM mode 173
17.6.5 Support of selective transmission and reception function 173
17.7 Cascade-FHM mode 174
17.7.1 Background 174
17.7.2 Shared cell configuration on cascaded FHMs 174

18 Configured subscriptions 175
18.1 Introduction 175
18.2 Description 175
18.3 Procedure 175
18.4 Notification encoding 176
18.5 Notification transport 177
18.6 Monitoring the communications channel between O-RU and Event-Collector 178
18.6.1 Background 178
18.6.2 Heartbeat encoding 179
18.6.3 Heartbeat control 179
18.6.4 Heartbeat procedure 179

19 Multi-Operator O-RU Operation 181
19.1 Introduction 181
19.2 High level shared O-RU architecture 181
19.3 Shared O-RU "start up" procedure 185
19.3.1 NETCONF server user account provisioning for shared resource operators 185
19.3.2 NETCONF call home to shared resource operator O-DUs 185
19.3.3 Enhanced sro-id based NETCONF access control 185
19.3.4 Supervision monitoring between shared O-RU and shared resource operator O-DUs 191
19.4 Shared O-RU interface management 191
19.4.1 VLAN and IP address management 191
19.4.2 Processing element configuration 191
19.4.3 Shared resource operator O-DU verification of C/U-Plane transport connectivity 192
19.5 Shared O-RU C/U-Plane delay management 192
19.5.1 Adaptive delay operation with shared O-RU 192
19.5.2 Measuring transport delay parameters with Shared O-RU 192
19.6 Shared O-RU configuration management 192
19.6.1 Carrier configuration of the shared O-RU 192
19.6.2 Notification of configuration updates to shared resource operators 193
19.7 Shared O-RU performance management 193
19.8 Shared O-RU fault management 193
19.9 Synchronization aspects of shared O-RU 193
19.10 Co-ordinating service impacting procedures 193
19.10.1 Reset operation 193
19.10.2 Locked administrative state 193
19.10.3 Antenna calibration 194
19.11 Partitioning of shared O-RU carrier resources 194
19.11.1 Partitioning of Shared O-RU advertised resources 194

19.11.2 Partitioning of eAxC identities 194
19.11.3 Partitioning of links, endpoints and array carriers 194
19.11.4 Partitioning of static endpoints 194
19.11.5 Shared O-RU beamforming configuration 194
19.11.6 Shared O-RU with antenna line devices 195
19.12 Example shared O-RU carrier configuration and operation procedure 195
19.13 Shared O-RU and LAA operation 196
19.14 Shared O-RU operation in combination with shared cell 197

20 Network energy saving 197
20.1 Introduction 197
20.2 Carrier deactivation for energy saving 197
20.2.1 High level principle of carrier deactivation for energy saving 197
20.2.2 Synchronization aspects for carrier deactivation for energy saving 197

Annex A (normative): Common alarm definition 198
A.1 Introduction 198

Annex B (normative): Counters 206
B.1 Counter definition 206
B.2 Transceiver statistics 207
B.2.1 Transceiver measurements 207
B.2.2 Statistics calculation 208
B.2.3 Frequency table generation 208
B.3 Rx window statistics 209
B.3.1 Rx window measurement 209
B.4 Tx statistics 210
B.5 Energy, power and environmental statistics 210
B.6 Symbol RSSI statistics 210
B.6.1 Statistics calculation 210
B.6.2 Frequency Table Generation 211

Annex C (informative): Optional multi-vendor functionality 211
C.1 Optional multi-vendor namespace 211
C.2 Optional YANG features 212
C.3 Optional features exposed using O-RAN YANG models 216

Annex D (informative): YANG module graphical representation 224
D.1 Introduction 224
D.2 System folder 224
D.2.1 o-ran-supervision.yang module 224
D.2.2 o-ran-usermgmt.yang module 224
D.2.3 o-ran-hardware.yang module 225
D.2.4 o-ran-fan.yang module 225
D.2.5 o-ran-fm.yang module 225
D.2.6 o-ran-ves-subscribed-notifications.yang module 226
D.2.7 o-ran-certificates.yang module 226
D.3 Operations folder 226
D.3.1 o-ran-operations.yang module 226
D.3.2 o-ran-file-management.yang module 227
D.3.3 o-ran-software-management.yang module 227
D.3.4 o-ran-lbm.yang module 228
D.3.5 o-ran-udp-echo.yang module 229
D.3.6 o-ran-ecpri-delay.yang module 229
D.3.7 o-ran-performance-management.yang module 229
D.3.8 o-ran-uplane-conf.yang module 235
D.3.9 o-ran-ald module 244
D.3.10 o-ran-troubleshooting module 244
D.3.11 o-ran-laa-operations module 245
D.3.12 o-ran-trace module 245
D.3.13 o-ran-ieee802-dot1q-cfm module 245
D.4 Interfaces folder 245

D.4.1 o-ran-interfaces.yang module 245
D.4.2 o-ran-processing-elements.yang module 246
D.4.3 o-ran-transceiver.yang module 248
D.4.4 o-ran-mplane-int.yang module 248
D.4.5 o-ran-dhcp.yang module 249
D.4.6 o-ran-externalio.yang module 250
D.4.7 o-ran-ald-port.yang module 251
D.4.8 o-ran-ethernet-forwarding.yang module 251
D.5 Sync folder 251
D.5.1 o-ran-sync.yang module 251
D.6 Radio folder 253
D.6.1 o-ran-module-cap.yang module 253
D.6.2 o-ran-delay-management.yang module 254
D.6.3 o-ran-beamforming.yang module 255
D.6.4 o-ran-laa.yang module 261
D.6.5 o-ran-antenna-calibration.yang module 261
D.6.6 o-ran-shared-cell.yang module 262

Annex E (normative): Corresponding YANG Module Definition 265

Annex F (informative): Out of scope functionality 267
F.1 Out of scope functionality 267

Annex G (informative): o-ran-lbm.yang and o-ran-ieee802-dot1q-cfm.yang co-ordination 268
G.1 Model structure 268
G.2 Mapping between models 269

Change history 271

History 275

Intellectual Property Rights

Essential patents

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

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

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Mobile Standards Group (MSG).

MSG.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Executive summary

This Technical Specification defines the Management Plane for the O-RAN Open Fronthaul based on the selected lower-layer split point as defined within the Open Fronthaul Control Plane, User Plane and Synchronization Plane specification. This Technical Specification is used in combination with a set of associated YANG models to enable operation of an O-RAN alliance defined O-RU.

1 Scope

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

version xx.yy.zz

where:

xx: the first digit-group is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc. (the initial approved document will have xx=01). Always 2 digits with leading zero if needed.

yy: the second digit-group is incremented when editorial only changes have been incorporated in the document. Always 2 digits with leading zero if needed.

zz: the third digit-group included only in working versions of the document indicating incremental changes during the editing process. External versions never include the third digit-group. Always 2 digits with leading zero if needed.

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).

2 References

2.1 Normative 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.

[1] VOID

[2] O-RAN ALLIANCE O-RAN-WG4.CUS.0: "Control, User and Synchronization Plane Specification".

[3] RFC-6241: "Network Configuration Protocol (NETCONF) ".

[4] RFC-7950: "The YANG 1.1 Data Modeling Language".

[5] RFC-6242: "Using the NETCONF Protocol over Secure Shell (SSH) ".

[6] RFC-4252: "The Secure Shell (SSH) Authentication Protocol".

[7] RFC-4253: "The Secure Shell (SSH) Transport Layer Protocol".

[8] RFC-2132: "DHCP Options and BOOTP Vendor Extensions".

[9] RFC-3925: "Vendor-Identifying Vendor Options for Dynamic Host Configuration Protocol version 4 (DHCPv4) ".

[10] RFC-2131: "Dynamic Host Configuration Protocol".

[11] RFC-4862: "IPv6 Stateless Address Autoconfiguration".

[12] VOID

[13] VOID

[14] VOID

[15] RFC-8071: "NETCONF Call Home and RESTCONF Call Home".

[16] Small Form Factor Committee SFF-8472v12.1 (2021): "Management Interface for SFP+ ".

[17] VOID

[18] RFC-862: "Echo Protocol".

[19] VOID

[20] VOID

[21] RFC-5277: "NETCONF Event Notifications".

[22] ITU-T Recommendation G.8275.1 (2016): "Precision time protocol telecom profile for phase/time synchronization with full timing support from the network".

[23] ITU-T Recommendation G.810 (1996): "Definitions and terminology for synchronization networks".

[24] IEEE 1588v2 (2008): "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems".

[25] ITU-T Recommendation Y.1731 (2015): "Operation, administration and maintenance (OAM) functions and mechanisms for Ethernet based networks".

[26] Antenna Interface Standards Group AISG 2.0 (2006): "Control interface for antenna line devices".

[27] 3GPP TS 37.462 v15.2.0: "Iuant interface: Signalling transport".

[28] 3GPP TS 37.466 v15.5.0: "Iuant interface: Application part".

[29] VOID

[30] ITU-T Recommendation X.733 (1992): "Information Technology – Open Systems Interconnection - System Management: Alarm Reporting Function".

[31] RFC-6187: "X.509v3 Certificates for Secure Shell Authentication".

[32] 3GPP TS 36.213v13.6.0, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures".

[33] RFC-4361: "Node-specific Client Identifiers for Dynamic Host Configuration Protocol Version Four (DHCPv4) ".

[34] Small Form Factor Committee SFF-8636v2.9.3 (2019): "Specification for Management Interface for Cabled Environment".

[35] RFC-6470: "Network Configuration Protocol (NETCONF) Base Notifications".

[36] VOID

[37] RFC-8639: "Subscription to YANG Notifications".

[38] RFC-7951: "JSON Encoding of Data Modeled with YANG".

[39] RFC-5246: "The Transport Layer Security (TLS) Protocol Version 1.2".

[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)".

[41] RFC-7589: "Using the NETCONF Protocol over Transport Layer Security (TLS) with Mutual X.509 Authentication", IETF, June 2015.

[42] RFC-8446: "The Transport Layer Security (TLS) Protocol Version 1.3".

[43] RFC-7030: "Enrollment over Secure Transport".

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

[45] IANA Transport Layer Security (TLS) Parameters. https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml.

[46] 3GPP TS 33.210: "Network Domain Security (NDS); IP network layer security".

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

[48] RFC-5288: "AES Galois Counter Mode (GCM) Cipher Suites for TLS".

[49] RFC-7540: "Hypertext Transfer Protocol Version 2 (HTTP/2)".

[50] IEEE Std 1588-2019: "Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems".

[51] 3GPP 33.310: "Network Domain Security (NDS); Authentication Framework (AF) ".

[52] 3GPP 32.509: "Data formats for multi-vendor plug and play eNode B connection to the network".

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

[54] RFC-4217: "Securing FTP with TLS".

[55] RFC-8996: "Deprecating TLS 1.0 and TLS 1.1".

[56] O-RAN ALLIANCE O-RAN-SFG.O: "O-RAN Security Protocols Specifications".

[57] 3GPP TS 28.552: "5G performance measurements".

[58] RFC-8415: "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6".

[59] RFC-8525: "YANG Library".

[60] 3GPP TS 28.532: "Management and orchestration; Generic management services".

[61] RFC-4191: "Default Router Preferences and More-Specific Routes".

[62] RFC-3442: "The Classless Static Route Option for Dynamic Host Configuration Protocol (DHCP) version 4".

[63] RFC-5656: "Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer".

[64] RFC-8341: "Network Configuration Access Control Model".

[65] ISO 13239 (2002): "Information technology — Telecommunications and information exchange between systems — High-level data link control (HDLC) procedures".

[66] 3GPP TS 38.104: "NR; Base Station (BS) radio transmission and reception".

[67] RFC-6022: "YANG Module for NETCONF Monitoring".

[68] RFC-8332: "Use of RSA Keys with SHA-256 and SHA-512 in the Secure Shell (SSH) Protocol".

[69] IEEE 802.1Q (2018): "IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks"

[70] IEEE 802.1X (2020): "IEEE Standard for Local and Metropolitan Area Networks–Port Based Access Control".

[71] O-RAN ALLIANCE O-RAN-SFG.O: "O-RAN Security Requirements Specifications".

[72] 3GPP TS 28.316: "Management and orchestration; Plug and connect; Data formats ".

[73] O-RAN ALLIANCE ORAN-WG9.XPSAAS.0: "Xhaul Packet Switched Architectures and Solutions".

[74] 3GPP TS 36.211 "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation".

[75] 3GPP TS 38.211 "NR; Physical channels and modulation".

2.2 Informative references

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.

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

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[i.2] ONAP 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

[i.3] Metro Ethernet Forum MEF.38 (2012): "Service OAM Fault management YANG Modules".

[i.4] An extensible YANG validator and converter in python, https://github.com/mbj4668/pyang.

[i.5] RFC-1035: "How to Use Anonymous FTP".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms and definitions given in 3GPP TR 21.905 [i.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 [i.1].

Ad-Hoc O-RU Controller: O-RU controllers that are not configured by DHCP when O-RU performs M-plane transport resolution and not configured by an O-RU controller using o-ran-mplane-int YANG model. An O-RU shall not perform call home to an Ad-Hoc O-RU Controller.

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.

Call home O-RU Controller: O-RU controllers that are configured by DHCP when O-RU performs M-plane transport resolution as part of start-up installation procedure or that are provisioned by an O-RU controller in o-ran-mplane-int.yang.

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, for example PNF registration. Event-collectors are configured when O-RU performs M-plane transport resolution as part of start-up installation procedure.

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.

Multi O-DU O-RU: An O-RU that supports the **SHARED-ORU-MULTI-ODU** feature.

Multi-Operator O-RU: An O-RU that supports the **SHARED-ORU-MULTI-OPERATOR** and **SHARED-ORU-MULTI-ODU** features.

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).

PRACH Symbol: A resource in the time domain having the duration of (1/Subcarrier Spacing), following cyclic prefix (excluded) in a PRACH occasion. Cyclic prefix and one or more of such consecutive PRACH Symbols constitute a PRACH preamble or a NPRACH symbol group defined in 3GPP TS 36.211 [72] and 38.211 [73].

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

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.

Shared O-RU Host: The role performed by the NETCONF client associated with the operator of a Multi-Operator O-RU who determines how the resources of a Multi-Operator O-RU are partitioned between Shared Resource Operators.

Shared Resource Operator: The role performed by the NETCONF client associated with an operator that utilizes the carrier resources of a Multi-Operator O-RU. The Shared Resource Operator is allocated a Shared Resource Operator identity by the Shared O-RU Host.

NOTE: A NETCONF client of an operator can simultaneously perform the Shared O-RU Host role and the Shared Resource Operator role. Alternatively, a NETCONF client of an operator can only perform the Shared O-RU Host role and not Shared Resource Operator role, in which case the operator corresponds to a neutral host operator.

Software build: A consistent set of software files used for a specific radio type of a defined HW version. Software build is subject of versioning and maintenance. Software build is contained within a software package.

Software package: An archive delivered by an O-RU vendor. It contains one or multiple builds and can be used for one or several radio products.

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 analogue 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.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [i.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 [i.1].

ALD	Antenna Line Device
AVP	Average Power
BCN	BTS Clock Number
CA	Certificate Authority
CA/RA	Certificate Authority/Registration Authority
CCM	Continuity Check Message
CMP	Certificate Management Protocol
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
EAP	Extensible Authentication Protocol
FHM	Fronthaul Multiplexer
FTPES	File Transfer Protocol Explicit-mode Secure
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
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 clause 3.1)
O-RU	O-RAN Radio Unit
OMA	Optical Modulation Amplitude
PAE	Port Access Entity

PDV	Packet Delay Variation
PLFS	Physical Layer Frequency Synchronization
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

4 General

4.1 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 the present document, the definition of the YANG models shall take precedence.

4.2 Void

4.3 Revision and compatibility handling

The revision statement in the YANG models shall be used to describe future revisions to the models that are backwards compatible, where backwards compatibility changes follow the rules specified in RFC 7950 [4], clause 11. Backwards incompatible changes shall 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 shall include 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 shall also include a description which is used to track the versioning of the YANG model. All revision statement descriptions 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 clause 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

O-RU Controllers that receive YANG library information from the O-RU with a module revision that is a higher version than the module revision currently used by the O-RU Controller can assume that models with the same namespace have been updated to ensure backwards compatibility. The O-RU Controller can continue to use its current module version and any unknown schema nodes received from the O-RU, i.e., those introduced in later revisions, should be ignored by the O-RU Controller.

NOTE: There have been non-backwards compatibility changes made during the development of earlier versions of the present document. This means that the YANG backwards compatibility rules described above do not accommodate these specific changes. In particular:

- "ORAN-WG4.MP.0-v05.00" : Switched the mandatory cipher 3DES-CBC to AES128-CTR.
- "ORAN-WG4.MP.0-v06.00" : Switched from all O-RUs required to support IPv4 to enable IPv6 only O-RUs.
- "ORAN-WG4.MP.0-v12.00" : Switched IEEE 802.1X supplicant functionality support in O-RU from recommended to mandatory.

4.4 Namespace compatibility handling

If backwards incompatible changes have been made, the <model-number> used in the YANG model namespace shall be incremented. Following such changes, an O-RU may include multiple backwards incompatible namespaces in its YANG library, for example "urn:o-ran:"<model-name>":1.0" and "urn:o-ran:"<model-name>":2.0".

The O-RAN Alliance's IPR policy defines terms regarding the modification of O-RAN defined specifications. When such modifications are necessary, the preferred approach for realizing such is for the third-party licensee to publish their own augmentations to the O-RAN defined YANG models and procedures. An O-RU that supports such third-party modifications shall include such model augmentations in its YANG library. Consequently, an O-RU Controller should be prepared to ignore any unknown models, e.g., developed according to such a procedure.

5 High level description

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

5.1.1 Architecture for O-RAN WG4 fronthaul functional split

This O-RAN FH specification addresses the lower layer functional split as depicted in Figure 5.1.1-1. Refer to the O-RAN CUS plane specification [2], clause 4.1 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.

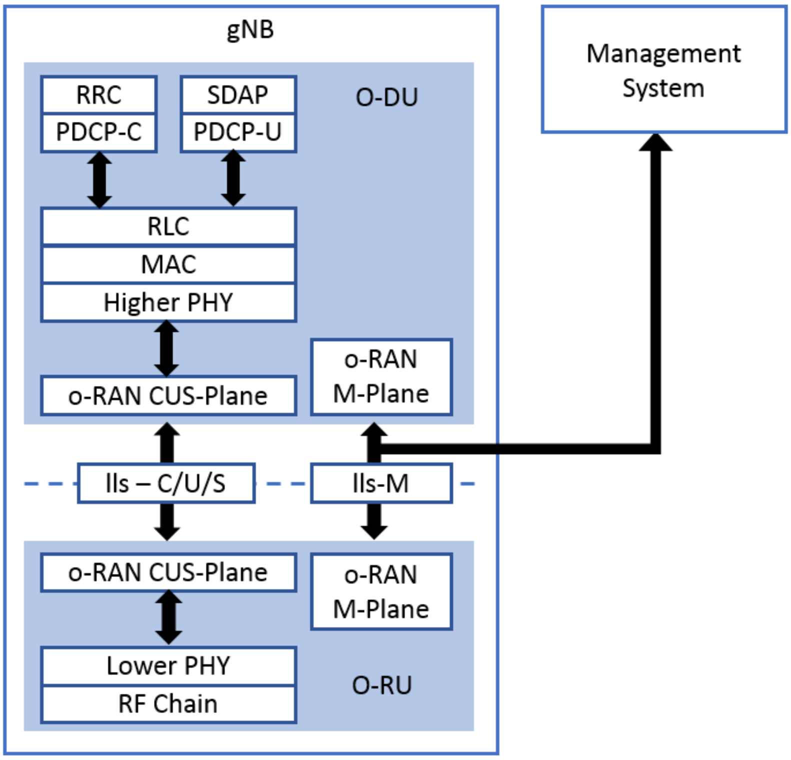


Figure 5.1.1-1: O-RAN WG4 FH functional split

5.1.2 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 5.1.2-1, the O-RU is managed entirely by one or more O-DU(s) using a NETCONF based M-Plane interface.

2) **Hybrid model**. As shown on the right side of Figure 5.1.2-1, 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.

NOTE: In the hybrid model, the NETCONF clients connecting to the O-RU may be of different privilege classes (e.g., "hybrid-odu", "smo", “swm” or “fm-pm”), allowing functions like O-RU software management, performance management, configuration management and fault management can be managed directly by the management system(s).

A Multi-Operator O-RU shall additionally support M-Plane architecture models which involve one or more O-DUs operated by one or more different Shared Resource Operators. Clause 19.2 describes the high level shared O-RU architecture, including hybrid and hierarchical architecture models for Multi-Operator O-RU operation with one or more different Shared Resource Operators.

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 clause 6 for details how O-RU acquires the IP address of O-DU and SMO for the M-plane communication.

As described in clause 6, there is no explicit signalling 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. All O-RUs shall be able to support both hierarchical and hybrid deployment.

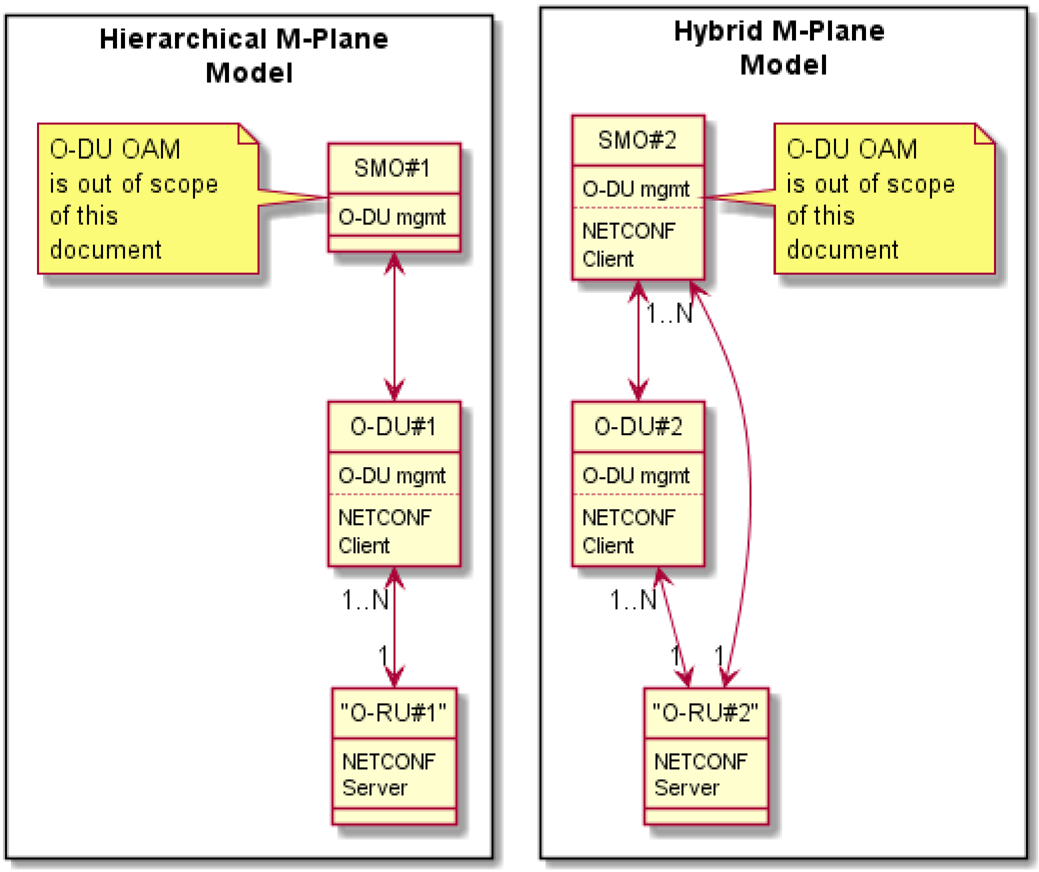


Figure 5.1.2-1: M-Plane Architecture

NETCONF/YANG is used as the network element management protocol [3] and data modelling language [4]. Use of such a standardized framework and common modelling 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.

5.1.3 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 the present document. The O-RU shall support either IPv4 or IPv6 and optionally support dual stack (IPv4 and IPv6).

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

5.1.4 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 start-up, 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 clause 6.

NOTE: The use of "start up" terminology in the present document is distinct from the "start-up" capability used in a NETCONF environment to indicate that a device supports separate running and startup configuration datastores. The present document 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 download is performed using sFTP with SSH or FTPES with TLS as specified in RFC 4217 [54].

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 are used (see clause 10 for more details).

2) **File Upload:** This option uses the file upload procedure defined in clause12. 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 or FTPES connection between O-RU to O-DU/SMO.

5.2 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 in the Figure 5.2-1 below. The transport network layer built on IP transport, SSH/TCP, and 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.

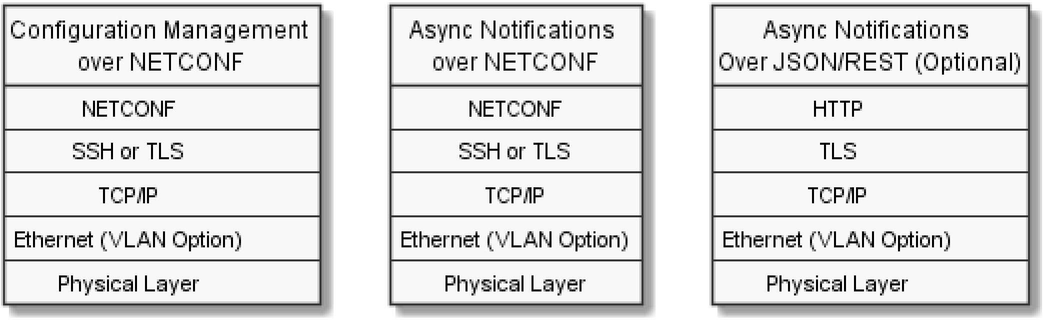


Figure 5.2-1: M-plane protocol stack

5.3 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 clauses, Annex D and the repository of YANG models for more details on each of these modules.

5.4 Security

The M-Plane provides end to end security as a mandatory feature, see Table 5.4-1. M-Plane security shall support NETCONF/SSHv2 as specified in RFC 6242 [5] and NETCONF/TLS 1.2 as specified in RFC 7589 [41]. TLS 1.3 as specified in 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. An O-RU shall support sFTP based file transfer over SSH and FTPES based file transfer over TLS. For the O-DU, the operator may use SSH, TLS, or both. It is recommended that operators use NETCONF/TLS and FTPES in production networks. TLS versions 1.0 and 1.1 have been formally deprecated by the IETF as specified in RFC 8996 [55] and shall not be used.

Table 5.4-1: M-Plane Security

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 NETCONF/SSHv2, as specified in RFC 6242 [5] b) Mandatory support for NETCONF/TLS 1.2, as specified in RFC 7589 [41] c) Optional support for TLS 1.3, as specified in RFC 8446 [42]
Optional support of JSON/REST	Yes	Yes	Yes	HTTPS used for JSON/REST transport

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 specified in RFC 4252 [6]. Clause 6 of the present document describes the authentication approach based on username and password, as specified in clause 8 of RFC 4252 [6], as well as based on X.509 certificates, as specified in clause 7 of RFC 4252 [6].

The SSHv2 transport level security (encryption algorithms, data integrity algorithms) shall be as specified in 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 by the clients, and username/password-based client authentication shall be done by the server as part of the SSH session establishment as specified in RFC 4253 [7]. The O-RU shall support the host key algorithms and key exchange methods for securing the Secure Shell (SSH) transport, as specified in clause 10.1 of RFC 5656 [63].

Host key algorithms:

- ecdsa-sha2-nistp256
- ecdsa-sha2-nistp384
- ecdsa-sha2-nistp521

Key exchange methods:

- ecdh-sha2-nistp521
- ecdh-sha2-nistp384
- ecdh-sha2-nistp256

In addition, the following host key algorithms should be supported as specified in RFC 8332 [68]:

- rsa-sha2-256
- rsa-sha2-512

In order to ensure backwards compatibility with equipment supporting earlier versions of this document, a vendor may provide support for other, optional host key algorithms and key exchange methods.

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 ciphering algorithm and hmac-sha2-256 shall be supported for integrity algorithm.

NOTE 1: 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 as specified in 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 shall support X.509 certificate-based authentication using TLS 1.2 as specified in RFC 7589 [41]. When X.509 based authentication is used, NETCONF server identity is as specified in clause 4 of RFC 6125 [40] and NETCONF client identity is specified in clause 7 of RFC 7589[41].

TLS 1.2 implementations shall support the following TLS Cipher Suites with SHA-256 and AES Galois Counter Mode as specified in clause 9.2 of RFC 7540 [42] and clause 7 of 3GPP TS 33.210 [46]:

- ECDHE_RSA_WITH_AES_128_GCM_SHA256 as specified in clause 3.2 of RFC 5289 [47].
- DHE_RSA_WITH_AES_128_GCM_SHA256 as specified in clause 3 of RFC 5288 [48].

It is mandatory that TLS implementations follow the rules on allowed cipher suites specified in clause 4.2.2 of the O-RAN Security Protocols Specifications [56]. 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 clause 6.2.3 of 3GPP TS 33.210 [46].

Operators may select the authentication mechanism and protocol to use as shown in Table 5.4-2.

Table 5.4-2: Mandatory and Optional Features for O-RU Authentication

Protocol	Certificate lifecycle management	PKIX (Public Key Infrastructure with X.509 Certificates)	Simple Public Key	Password-based Authentication
TLS 1.2	Mandatory to support CMPv2, optional to support vendor certificate lifecycle management	Mandatory to support / Optional to use	Not specified in RFCs 5246/8446	Not specified for use with NETCONF
SSHv2	Optional to support CMPv2, optional to support vendor certificate lifecycle management	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

6 "Start-up" installation

6.1 General

This clause 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. O-RU enables its port(s).
- O-RU controller/NETCONF Client(s) and/or event-collector is/are in operation.
- Physical interface(s) is(are) connected.

Post-condition:

- O-RU is running software from slot with active = TRUE
- O-RU has marked slot with currently used software as running = TRUE
- 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 "sudo" or "hybrid-odu" access privileges can control the carrier configuration of the O-RU/NETCONF server in O-RU.

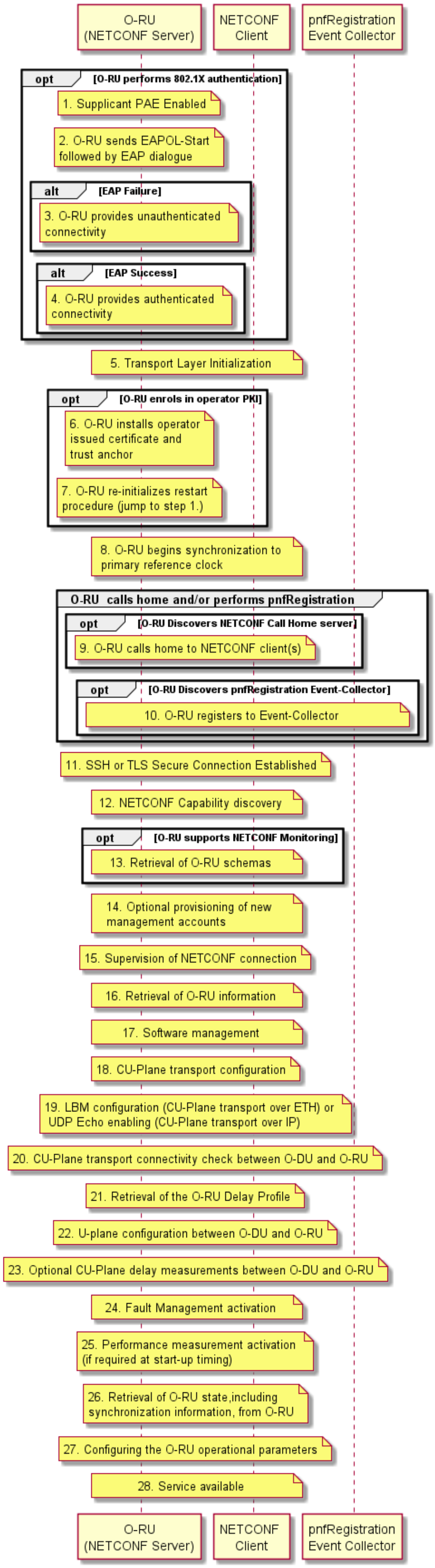


Figure 6.1-1: Overall of Start-Up Installation

At the power-on of O-RU or following an O-RU restart, the following procedures are performed, as illustrated in Figure 6.1-1.

1) (opt) Supplicant PAE is enabled on the port(s)

2) (opt) The O-RU initiates authentication and attempts to perform an EAP authentication dialogue with a peer Authenticator PAE.

NOTE: This does not limit the ability of an Authenticator PAE to initiate the authentication, as defined in clause 8.1 of [70]

3) (alt) EAP failure results in O-RU providing unauthenticated connectivity

4) (alt) EAP Success results in O-RU providing authenticated connectivity

5) 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.

6) The O-RU has not yet enrolled in an operator PKI and O-RU discovers a CA/RA server. O-RU attempts to enrol in operator PKI.

7) After installing the operator issued certificate, the O-RU re-initializes its start-up procedure (jump to step 1)

8) O-RU begins synchronization of the O-RU against a Primary Reference Clock. Step 8 may be in parallel with step 5 for some O-RU implementation.

9) (opt) O-RU performs NETCONF Call Home to discovered O-RU controller(s)

10) (opt) O-RU performs pnfRegistration to discovered event-collector.

11) O-RU controller performs SSH or TLS connection establishment.

12) O-RU and O-RU controller perform NETCONF capability discovery.

13) (opt) O-RU controller retrieves O-RU schemas using <get-schema> RPC [67].

14) O-RU controller performs optional provisioning of new management accounts (typically only performed once during pre-staging). O-RU controller may perform provisioning of certificate-to-NETCONF username mapping information to the O-RU after account provisioning in case certificate-based client authentication is used.

15) O-RU and O-RU controller perform supervision of NETCONF connection.

16) O-RU controller performs retrieval of O-RU information. (opt) O-RU controller retrieves O-RU S-Plane information and if necessary, it updates the O-RU's S-Plane configuration. (This step may be started any time after step 15 but needs to be completed before step 28).

17) O-RU controller performs SW management.

NOTE: If an O-RU is running with factory default software, the O-RU functionality is permitted to comprise a sub-set of a fully operating O-RU. In such scenarios, it is recommended that the O-RU controller triggers a software update to fully functioning O-RU software.

18) O-DU performs CU-Plane transport configuration

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

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

21) O-RU controller retrieves the O-RU delay profile from the O-RU.

22) 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.

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

24) O-RU controller performs Fault Management activation by creating a subscription to the YANG notifications defined in the o-ran-fm.yang model. Additionally, the O-RU controller uses the same YANG model to retrieve the list of O-RU's active alarms. See Clause 11 for details.

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

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

27) O-RU controller configures the O-RU operational parameters.

28) Service available.

Additional Start-up Considerations:

The synchronization procedures started in step 8 needs to be completed before service is available. If the O-RU's power-on S-Plane settings are incompatible with the network's S-Plane design, then at step 16 the O-RU controller should reconfigure the O-RU's S-Plane settings.

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.

The details of the above start-up procedure are covered in sub-clauses 6.1 and 6.3 to 6.7.

Cross Reference to other clauses:

The details of 8, 16 and 26. Synchronization management is described in clause 13.

The method of 14 and 25 retrieval of O-RU information is described in clause 9.

The detail of 17. SW management is described in clause 8.

The detail of 20. C/U-Plane transport connectivity checking between O-DU and O-RU is described in clause 7.

The detail of 21. Retrieval of the O-RU delay profile and 13. C/U-Plane delay measurements are described in clause 7.

The detail of 22. U-plane configuration is described in clause 15, and C/U-Plane transportation configuration is described in clause 7.

The detail of 25. Performance management is described in clause 10.

The detail of 24. Fault management is described in clause 11.

The method of 27. Control to make service available is described in clause 15.

6.2 Management plane transport aspects

6.2.1 Transport establishment

This clause 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 clause. Transport aspects of the C plane and U plane are covered in clause 7.

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, as illustrated in Figure 6.2.1-1:

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 the present document. 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 signalled, 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 signalled, 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 signalled by the DHCP server.

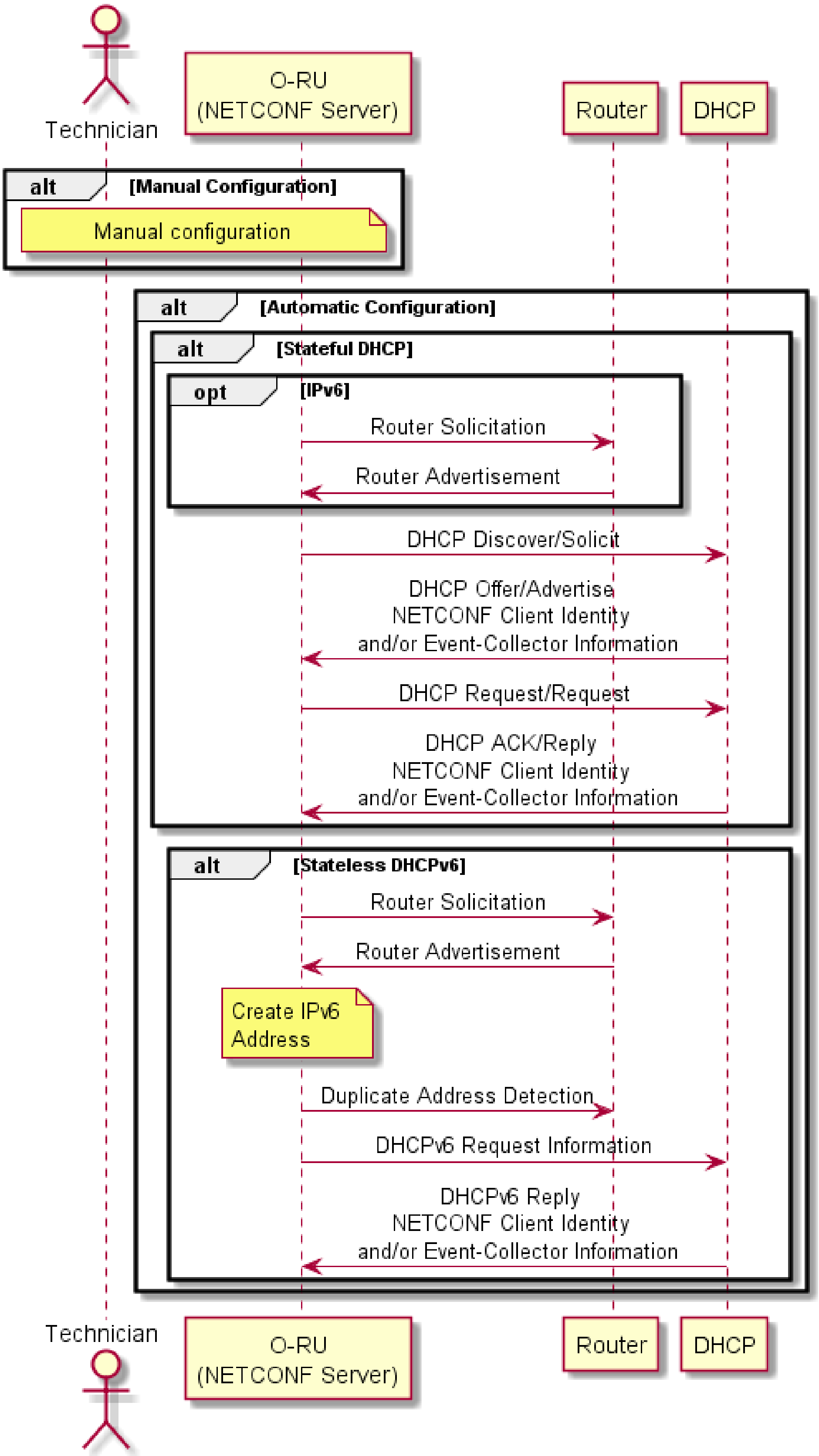


Figure 6.2.1-1: Transport Layer Establishment for M-plane

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 clauses are used:

- O-RU identification in DHCP messages from O-RU (clause 6.2.2).
- VLAN discovery aspect for M-plane (clause 6.2.3).
- IP address assignment to O-RU (clause 6.2.4).
- Discovery of address information of O-RU controller(s) and/or Event-Collector (clause 6.2.5 and/or clause 6.2.7).

6.2.2 O-RU identification in DHCP

The O-RU shall use the DHCP Vendor Class option(s) with the **vendor-class-data** string within the o-ran-dhcp YANG model or vendor identifying vendor class option with enterprise number and vendor-class-data to identify itself as an O-RU to DHCP servers. When the O-RU supports IPv4, it shall identify itself either 1) use option 60 Vendor Class Identifier, as specified in RFC2132 [8], OR 2) use option 124 Vendor Identifying Vendor Class Option, as specified in RFC3925 [9]. When the O-RU supports IPv6, it shall identify itself using the DHCPv6 Vendor Class Option 16.

O-RU can identify itself with both O-RAN registered IANA Enterprise Number and 3GPP registered Enterprise Number when using option 124 to retrieve O-RU controller, event controller and CA/RA information in DHCPv4 server. If an O-RU intends to get only O-RAN domain information, O-RU can use opt 60 or opt 124 with only O-RAN enterprise number.

NOTE: To support DHCP server implementations that are limited in their support for multiple instances of the Vendor-Specific Information Options, an operator and vendor can agree to use the O-RAN registered IANA Enterprise Number in DHCPv4/DHCPv6 messages to identify itself to signal the CA/RA server information.

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"

OR

2) "o-ran-ru2/<vendor>/<product-code>", e.g., "o-ran-ru2/vendorA/ORUAA100"

OR

3) "o-ran-ru2/<vendor>/<product-code>/<serial-number>", e.g., "o-ran-ru2/vendorA/ORUAA100/FR1918010111"

The Vendor Class Identifier should be selected to avoid the likelihood that different vendors select identical strings, e.g., using a vendor namespace registry or ensuring that the identifier includes the <product-code> information.

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 the information when selecting an address pool from which to allocate an IP address to the O-RU or when selecting which management plane O-RU Controller information to configure in the O-RU.

6.2.3 Management plane VLAN discovery aspects

The O-RU is connected to one or more Ethernet ports. The transport systems may be realized such that these Ethernet ports is/are configured either as an access port, where untagged Ethernet frames are used, or as a trunk port, where multiple VLANs are configured. During start up, the O-RU is typically unable 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 shall determine whether it is connected to an access port or a trunk port. In particular, when connected to a trunk port, the O-RU shall 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 clause 6.2.5, clause 6.2.7 or clause 6.2.6.1.

NOTE 1: An O-RU which supports IPv6 can 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.

NOTE 2: Versions prior v12.0 of the present document did not define management plane VLAN discovery based on CA/RA server identity as specified in clause 6.2.6.1. When operating with O-RUs that support an earlier version of this documents, an operator wanting to operate a VLAN that restricts access to production O-RU controllers, e.g., offering sole access to a CA/RA server, can configure a dummy NETCONF client identity to be returned to the O-RU in the DHCP OFFER message.

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

When the O-RU has been previously configured with management plane VLAN information, the O-RU may use this information to optimize its discovery of the VLAN ID(s) used for management plane connectivity. Previously configured management plane VLAN information includes an O-RU that stores the last VLAN(s) used for management plane connectivity, and/or an O-RU which has been previously configured with a range of management plane VLANs by a NETCONF client using the contents of the **searchable-mplane-access-vlans-info** container that have 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 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 the full range of VLAN IDs (1~4094) on all its Ethernet ports.

The individual VLAN search algorithm used by an O-RU should ensure timely activation of the M-Plane while accommodating scenarios whereby there may be an intermittent or temporary connectivity problem between the O-RU and the DHCP server causing no DHCP response to be received on the M-Plane VLAN. The O-RU should repeatedly search using untagged frames and previously configured VLANs whenever it searches across the full range of VLAN IDs. The O-RU controller is able to recommend the maximum interval between repeatedly scanning for M-Plane connectivity on the untagged and configured VLANs using the **scan-interval** schema node.

For example, the default **scan-interval** is 60 seconds. If the O-RU takes 1 second to scan an individual VLAN, then after scanning every 60 out of the full range of VLAN IDs, the O-RU should repeat the scan for M-Plane connectivity on untagged and configured VLANs.

6.2.4 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, as described in RFC2131 [10] enables DHCP servers to configure IPv4 network address(es) on the O-RU. An O-RU implementing IPv4 shall support the behaviour specified in clause 6.1 of RFC 4361 [33], using stable DHCPv4 node identifiers in their dhcp-client-identifier option.

   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.

   An O-RU may indicate that it supports configuration of routing information as specified in RFC 3442 [62], 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.

   A network realized with multiple IPv6-enabled routers that support dynamic address assignment should use the extensions to Router Advertisements as specified in RFC 4191 [61] to configure the preference of the default route prefixes learnt by the O-RU using SLAAC.

3) IPv6 State-full address configuration uses DHCPv6, as specified in RFC 8415 [58] 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 3: The above does not restrict the realization of the DHCP server, which can be integrated with the O-DU, can be provided by the transport system, or can be accessed via a relay.

O-RUs that support more than one network interface should use a different client-identifier on each interface using a combination of Identity Association Unique Identifier and DHCP Unique Identifier, as specified in clause 6.1 of RFC 4361 [33] for DHCPv4 and clause 11 and 12 of RFC 8415 [58] for DHCPv6.

NOTE 4: An O-RU Controller can learn the particular client identifier(s) used by an O-RU by using the o-ran-dhcp YANG model.

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

6.2.5 O-RU controller discovery

This clause 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 as specified in RFC8415 [58], 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.

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

O-RAN defined vendor specific option shall be used to signal all NETCONF client information to the O-RU using option 43 or option 125 for DHCPv4 and option 17 for DHCPv6. The O-RU shall request this option. Multiple instances of NETCONF client information may be signalled, encoded as a sequence of type/length/value fields.

The definition of the types used within the DHCPv4 option 43, option 125/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 is encoded in an option 43 or option 125 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" is encoded in an option 43 or option 125 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

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 as specified in clause 6 of RFC 8071 [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 encodes the option 43 or option 125 TLV as

- Type: 0x86
- Length: 0x01
- Value: 01

NOTE: A previous version of the present document defined the use of OPTION_V4_SZTP_REDIRECT and OPTION_V6_SZTP_REDIRECT DHCP options to allow an O-RU to recover NETCONF Client information. In order to ensure backwards compatibility with equipment supporting earlier versions of this document, an operator and vendor can agree to continue to use OPTION_V4_SZTP_REDIRECT and/or OPTION_V6_SZTP_REDIRECT DHCP options to recover NETCONF Client information.

6.2.6 Certificate handling

6.2.6.0 Trust anchor provisioning

Before an O-RU can establish a mutual TLS connection with a signalling peer, e.g., with an O-RU Controller, an O-RU needs to be able to trace the peer’s certificate path to a valid trust anchor. To validate against a trust anchor, the O-RU shall be able to be provisioned with one or more trust anchor certificates. The O-RU shall ensure that all trust anchor certificates are stored in reset persistent memory and protected from external modification.

The O-RU shall be able to be provisioned with new trust anchors. The O-RU shall be able to have an existing trust anchor replaced, e.g., because it has expired. The O-RU should support ietf-truststore YANG model to enable an O-RU Controller to recover the list of provisioned trust anchors and associated public keys.

The present document defines the use of CMPv2 Initialization Response message to allow the discovered CA/RA server to provision a trust anchor.

NOTE: The CA/RA Server identity is configured by the DHCP server using techniques described in clause 6.2.6.1. The DHCP server is not considered by the O-RU as a trusted source of security bootstrapping data.

When shipped, an O-RU only trusts information that is signed or encrypted using a certificate chain leading to a pre-loaded trust anchor.

6.2.6.1 Certificate enrolment

An O-RU shall support certificate enrolment using CMPv2. 3GPP 32.509 [52] specifies how the O-RU supporting certificate enrolment over IPv4 can be configured with the IP address or FQDN of one or more Certification Authority (CA/RA) servers using DHCP Option 43 as specified in clause 4.2.2 of 3GPP 32.509 [52].

The DHCP Options specified in 3GPP 32.509 [52] do not specify how to signal a CA/RA server identity using DHCPv6. Hence, O-RU certificate enrolment using CMPv2 over IPv6 shall support the signalling of vendor specific options using DHCPv6 option 17. The format of the DHCPv6 option 17 follows the format of the DHCPv4 encoding as specified in clause 4.2.2 of 3GPP 32.509 [52], 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 shall be 53148 (as allocated by IANA to O-RAN Alliance).

3GPP has since published 28.316 [72] which now specifies how to signal a CA/RA server identity with DHCPv6 option 17 messages using the 3GPP registered IANA Enterprise Number (10415). The DHCP Options defined in [52] are a subset of those defined in [72]. An operator and vendor can agree to use the 3GPP registered IANA Enterprise Number in DHCPv6 messages that signal the CA/RA server information instead of the O-RAN registered IANA Enterprise Number.

If an O-RU has a point-to-point connection between the O-RU and the O-DU, or if an O-RU has a bridged connection between the O-RU and the O-DU and the O-DU is operating as the default gateway to provide connectivity between the O-RU and a remote CA/RA server, when the O-RU is required to perform certificate enrolment using CMPv2, the O-DU shall provide means for the O-RU to access the operator CA/RA for the O-RU certificate enrolment at the IP address conveyed as described above. If the FQDN option is used instead, the O-DU shall provide means for the O-RU to access an operator DNS server in addition. Examples of providing such access include:

- An IP forwarding function within the O-DU providing access to the operators PKI.
- A Network Address Translation function within the O-DU providing access to the operators PKI.
- A Registration Authority (RA) function accessible by the O-DU (in the same domain), which is part of the operator PKI.

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

6.2.6.2 CMPv2 based certificate enrolment

This section covers the use case where an O-RU is configured with the identity of a CMPv2 capable CA/RA server that the operator uses for certificate enrolment. The O-RU shall attempt to enroll in the operator-PKI and may be issued an operator-signed certificate.

Clause 9 of 3GPP 33.310 [51] specifies the use of CMPv2 used by base stations to obtain an operator-signed certificate using a secured communication based on the vendor-signed certificate in the base station and a vendor root certificate pre-installed in the CA/RA server. While the approach has been defined for provisioning certificates for use in either IPSec or TLS, the same techniques defined for provisioning TLS certificates are specified to be re-used here to provision certificates for use in securing the SSHv2 based M-Plane connection as specified in RFC 6187 [31]. Hence, the TLS client CA is responsible for issuing certificates to NETCONF clients, irrespective of whether NETCONF is secured using TLS or SSHv2. Similarly, the TLS server CA is responsible for issuing certificates to NETCONF servers, irrespective of whether NETCONF is secured using TLS or SSHv2.

The handling of certificates, including certificate profiles, shall follow the rules defined in 3GPP 33.310 for TLS CA certificates. In addition:

- when an O-RU generates a certificate signing request it shall populate the Subject Distinguished Name field with a string that includes the O-RU manufacturer's name, model and serial number. The exact Subject DN sub-field used is defined in the operator of the CA/RA server's certificate policy.

NOTE 1: In future, an O-RAN defined certificate policy may be defined to normalize the sub-field definition across the O-RAN ecosystem.

NOTE 2: There are various characters that may not be permissible in the Subject Distinguished Name Field, e.g., ":" (colon, hexadecimal character 0x34), "." (period, hexadecimal character 0x2E), "_" (underscore, hexadecimal character 0x5F), "#" (hash, hexadecimal 0x23), "£" (pound, hexadecimal 0xa3), "*" (asterisk, hexadecimal 0x2a) or """ (double quote, hexadecimal 0x22). Manufacturers that include such characters in their name, model and/or serial number should ensure such characters are removed before including in the Subject Distinguished Name Field.

- when transferring messages to the CA/RA server, the O-RU shall use the "port number of the CA/RA server" and the "path to the CA/RA directory" as signalled using the DHCP options as specified in clause 4.2.2 of 3GPP 32.509 [52]. If no DHCP based configuration is received by an O-RU, the O-RU shall use the default port 443 and default directory "/pkix/".
- The CA/RA server shall include the trust anchor for the operator issued certificate and the appropriate certificate chains in the initialization response message.
- The O-RU shall store the operator issued certificate and corresponding certificate chain in reset persistent memory.

NOTE 3: The current document does not define how to provision an O-RU with multiple trust anchors. As a consequence, for Multi-Operator O-RU scenarios operating with TLS, it is assumed that the Shared O-RU Host sign digital certificates for each of the Shared Resource Operators to enable them to mutually authenticate the TLS connection with the Multi-Operator O-RU.

When configured to operate with TLS, an O-RU that has a valid certificate issued by an operator PKI shall use this certificate to establish a mutually authenticated TLS connection for secure signaling with its O-RU Controller(s).

6.2.6.3 Operation with vendor-signed certificates

All O-RUs should be provisioned with a unique device vendor-signed certificate and its entire certificate chain up to the root. If an O-RU fails to enroll in an operator-PKI, as specified in clause 6.2.6.2, an O-RU configured to operate with TLS may use its vendor-signed certificate for setting up mTLS connections to its O-RU Controller(s), for example using the procedures specified in clause 6.2.5 to call home using NETCONF/TLS. The vendor's public trusted certificate and certificate management services should be accessible within the operator network to allow management of vendor-signed certificates. The extended use of unmanaged vendor-signed certificates, such as outside of enrolment, is not recommended.

6.2.7 Event-Collector discovery

This clause 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 clause 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, as specified in RFC8415 [58], 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 supporting PNF Registration 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 or option 125 for DHCPv4 and option 17 for DHCPv6. To achieve that, the O-RU shall request the option 43 or option 125 for DHCPv4 and option 17 for DHCPv6 as defined in clause 6.2.5. If the network provides an Event-Collector for the O-RU to send its pnfRegistration notification to, a DHCPv4 server shall respond with option 43 or option 125 if so requested, and a DHCPv6 server shall respond with option 17 if so requested.

The definition of the types used within the DHCPv4 option 43 or option 125/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.

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 is encoded in an option 43 or option 125 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" is encoded in an option 43 or option 125 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 [i.2].

For example, an Event-Collector expecting the pnfRegistration notification to be signalled using the ONAP defined format is encoded in the option 43 or option 125 TLV as

- Type 0x85
- Length: 0x01
- Value: 00

6.3 NETCONF call home to O-RU controller(s)

The O-RU aims to have NETCONF sessions with all of the call home O-RU Controller(s), either discovered using the DHCP options defined in clause 6.2.5, 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 call home O-RU Controllers that either do not attempt to initiate a NETCONF session with the O-RU, or are prevented from doing so, e.g., because of Network Address Translation limitations, the O-RU shall call home to all call home O-RU Controller identities with which it does not already have an active NETCONF session.

When the O-RU discovers, or is provisioned with, a call home O-RU Controller identity represented as an FQDN, the O-RU shall use DNS to resolve the IP address(es) of the FQDN identity. If as result of the resolution of the O-RU Controller FQDN to IP address, an O-RU receives multiple A and/or AAAA records, the O-RU should cycle through the received IP addresses in the A and/or AAAA records until it is able to establish a NETCONF session with a particular O-RU Controller identity. The O-RU should report the O-RU controller IP address used in the established NETCONF session in the **client-info** container in the o-ran-mplane-int YANG model. As a consequence, the O-RU is expected to have a single M-Plane session established per each O-RU controller's FQDN, regardless how many A/AAAA records were returned by the DNS server for each O-RU Controller FQDN.

An O-RU that supports more than one network interface shall be able to perform O-RU Controller discovery on each of its network interfaces. If an O-RU recovers identical O-RU controller identity information on multiple network interfaces, then it should select which one out of the multiple interfaces to use for its call home operation.

NOTE 1: The operation of an O-RU that discovers different O-RU Controller identities on separate network interfaces is not currently defined in the present document.

If the O-RU is unable to establish a NETCONF session with some of the call home O-RU Controller identities, the O-RU shall use the "re-call-home-no-ssh-timer" to repeatedly re-perform the call home procedure to all call home O-RU Controller identities with which the O-RU does not have an established NETCONF session, cycling through the A and/or AAAA records until it is able to establish a NETCONF session with a particular call home O-RU Controller identity. The same value of timer shall be used, irrespective of whether SSH or TLS is being used to transport the NETCONF session.

NOTE 2: The O-RU can cache the returned A/AAAA records for a period of time according to the DNS time to live. The setting of the DNS time to live is an operator name server configuration parameter. As an example, if an operator knows in advance of a change in the IP address(es) used by its O-RU controller(s), the operator can configure the name server with a minimum time to live value to ensure O-RUs request the new IP address(es) in a timely fashion. Name server configuration is outside the scope of the present document.

If the O-RU is unable to trigger the establishment of NETCONF session with at least one call home O-RU Controller after having repeated the call home procedure a total of **max-call-home-attempts** per O-RU Controller, then the O-RU should perform an autonomous reset.

The O-RU shall call home as specified in clause 4 of RFC 8071 [15] whereby the O-RU (NETCONF Server) initiates a TCP connection to the NETCONF client. When calling home to the NETCONF clients in the container **client-info**, O-RU shall use the port signalled using the RFC 8572 DHCP option [14] or manually configured during installation, and when calling home to the NETCONF clients in the container **configured-client-info**, O-RU shall use the port configured by other NETCONF Client. If no port was signalled or manually configured in the container **client-info**, or not configured in the container **configured-client-info**, O-RU shall use the port configured in **call-home-ssh-port** to indicate that the O-RU uses SSHv2 to secure the NETCONF connection and use the port configured in **call-home-tls-port** to indicate that the O-RU uses TLS to secure the NETCONF connection. If **call-home-ssh-port** doesn't exist, the O-RU shall use the IANA-assigned port 4334 to indicate that the O-RU uses SSHv2 to secure the NETCONF connection, and if **call-home-tls-port** doesn't exist, the O-RU shall use the IANA-assigned port 4335 to indicate that the O-RU uses TLS to secure the NETCONF connection. As illustrated in Figure 6.3-1, when the 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.

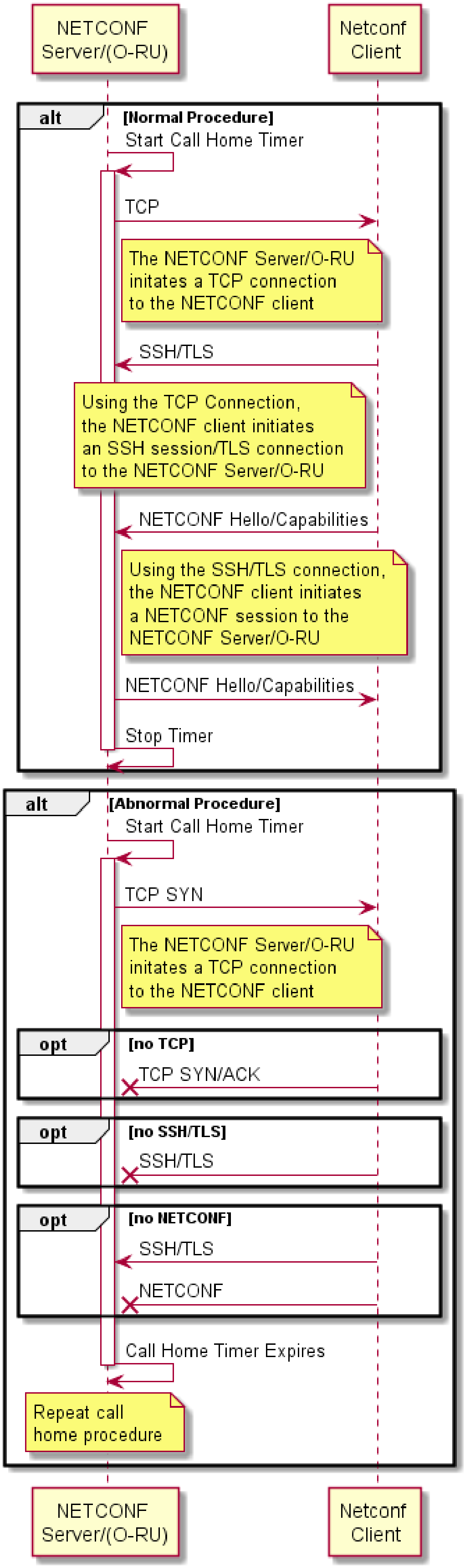


Figure 6.3-1: Outline of NETCONF call home procedure

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 as specified in clause 4.1 of RFC 8071 [15]. The establishment of NETCONF client privileges is covered in clause 6.5.

6.4 NETCONF connection establishment

6.4.0 General

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 as specified in RFC 6187 [31].

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

6.4.1 NETCONF security

As specified in clause 5.4, this version of the O-RU Management Plane Specification uses TLS 1.2, TLS 1.3, or SSHv2 for mutual authentication between the NETCONF server (O-RU) and the NETCONF client (O-DU or SMO).

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

6.4.2 NETCONF authentication

This version of the O-RU Management Plane Specification supports SSHv2 using password authentication method for SSHv2 as specified in clause 8 of RFC 4252 [6] and client authentication based on X.509 certificates as specified in RFC 6187 [31], and TLS 1.2, or TLS 1.3, 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-name** 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 as specified in RFC 7589 [41]. For the purposes of NETCONF server authentication, RFC 7589 [41] specifies server identity as specified in clause 4 of 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". As the default account may be operator specific, this may require that the O-RU provides facilities to configure securely this default account and cert-to-name mapping list 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 maps any authenticated X.509 v3 certificate to this default O-RAN account. The trust anchor for O-RU shall be provisioned automatically with online CA server during O-RU Plug and Play, and it shall 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 clause 6.5 for details of groups/privileges) as it can be used to create other accounts (see clause 6.4.3).

The operator and vendor shall agree how the default account can be determined by a NETCONF client operating in a multi-vendor O-RU environment based on the SSH response message sent by the O-RU. It is recommended that the vendor and operator agree on a particular format of the “SSH-protoversion-softwareversion SP comments CR LF” identification string, as defined in RFC4253 [7] clause 4.2, used by the SSH Server in the O-RU to identify the O-RU vendor and consequently the default account to be used by the NETCONF 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 and TLS 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).

6.4.3 User account provisioning

6.4.3.1 General

The NETCONF client with suitable privileges may provision user accounts on the O-RU, including the accounts (users) name, password, group (see clause 6.5 for details of groups/privileges) and whether a particular account is enabled or disabled.

- The **name** for the user is a string which should be between 3-32 characters. For account-type PASSWORD, the first character should be a lowercase letter. The remaining characters should be lowercase letters or numbers. For account-type CERTIFICATE, the characters should be lowercase letters, numbers, periods (ASCII code 0x2E) or @ sign (ASCII code 0x40).
- 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.
- Whether an account is **enabled.** The YANG model ensures that at least one user account is always enabled on the O-RU.
- When the O-RU supports the **SHARED-ORU-MULTI-OPERATOR** feature, the user account may be associated with zero or more Shared Resource Operator IDs (**sro-id**). Clause 19 describes the operation of the **SHARED-ORU-MULTI-OPERATOR** feature in more detail. When an **sro-id** is configured with a user account with "carrier" group/role privileges, the multi-operator shared O-RU feature is enabled.

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

Each account **name** in o-ran-usermgmt YANG model is identical to the **user-name** used in ietf-netconf-acm YANG model. User account provisioning shall include using the ietf-netconf-acm module to define which **groups** are associated with a particular **user-name** (see clause 6.5 for details of groups/privileges).

If certificate-based client authentication is used no password needs to be provisioned. At time of SSH or TLS 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 clause 6.8. 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 clause 8.8. Any other way to enable the default account is not precluded as O-RU vendor implementation matter.

The security principle defined in this clause 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 clause 6.4.2.

6.4.3.2 Certificates to NETCONF usernames mapping list provisioning

The O-RU controller with suitable privileges may provision certificate-to-NETCONF-username-mappings to the certificate-to-NETCONF-usernames mapping list at the O-RU, each entry of the mapping list containing a certificate fingerprint, a map-type and optional username.

- The certificate fingerprint is a digest of NETCONF client end-entry (EE) certificate or a digest of a trusted certificate authority (CA) which is part of the CA certificate chain of the NETCONF client EE certificate in X.509 binary format. From security consideration, it is recommended to use EE certificate fingerprint instead of CA certificate fingerprint.
- The map-type indicates how the NETCONF username associated with the certificate should be determined. O-RU shall support map-types A and B defined in section 7 of RFC 7589 [41]. Support for other map-types is optional. Support for map-type F, "common-name", is deprecated.
- The username name is a NETCONF account's name which is specified only when map-type is 'specified'.

In case a X.509v3 certificate that is used for client authentication is updated e.g., due to it being close to expiry, the O-RU controller shall update the X.509v3 certificate and if necessary re-provision the certificate-to-username mapping to O-RU.

6.5 NETCONF access control

This clause 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 clause 5, 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 as specified in RFC8341 [64].

Currently seven access control **groups** corresponding to different NETCONF client roles are defined and can be mapped to the user-name for a NETCONF session: "sudo", "smo", "hybrid-odu", "carrier", "nms", "fm-pm", and "swm". Table 6.5-1 maps the group/role **name** to different privileges. Privileges are defined per namespace for read "R", write "W" and execute "X" where "X" indicates access privilege to use RPC operations or to subscribe to Notifications. The NACM module is used to store the association between the **groups** and **user-names** and enables a **user-name** to be associated with multiple different **groups**. The configuration of the NACM module shall be stored in reset-persistent memory in O-RU.

During access control procedure, **groups** associated to the user are determined based on **user-name** that an O-RU controller provided or derived from O-RU controller's certificate. The user's access privileges are the combination (union) of the privileges associated with those **groups**. Please refer to table 6.5-1 for privilege details. O-RU shall support access control enforcement procedure defined in RFC 8341 [64] clause 3.4.

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.

When operating with a NETCONF client with user name user-list entry containing a configured sro-id and with "carrier" privileges, e.g., a client operated by a Shared Resource Operator, a Multi-Operator O-RU shall have its read, write and execute privileges for certain models further refined based on the sro-id associated with the name of the NETCONF client as configured using the o-ran-usermgmt YANG model. Further details are described in clause 19.

Table 6.5-1: Mapping of account groupings to O-RU module privileges

Module Rules	sudo	nms	fm-pm	swm	smo	hybrid-odu	carrier
"urn:o-ran:supervision:x.y"	RWX	---	---	---	RW- (NOTE 4)	RWX	R-X
"urn:o-ran:hardware:x.y"	RWX	RW-	---	---	RWX	R--	R--
"urn:ietf:params:xml:ns:yang:ietf-hardware"	RWX	RWX	R-X	---	RWX	R-X	R-X
"urn:ietf:params:xml:ns:yang:iana-hardware"	R--	R--	R--	---	R--	R--	R--
"urn:o-ran:user-mgmt:x.y"	RWX (NOTE 1)	--X (NOTE 8)	--X (NOTE 8)	--X (NOTE 8)	RWX (NOTE 1)	RWX (NOTE 1)	--X (NOTE 7, 8)
"urn:o-ran:fm: x.y "	R-X	R-X	R-X	---	R-X	R-X	R-X
"urn:o-ran:fan: x.y "	R--	R--	R--	---	R--	R--	R--
"urn:o-ran:sync: x.y "	RWX	RWX	R--	---	RWX	R-X	R-X
"urn:o-ran:delay: x.y "	RW-	R--	R--	---	R--	RW-	R--
"urn:o-ran:module-cap: x.y "	RW-	R--	R--	---	R--	RW-	R--
"urn:o-ran:udpecho: x.y "	RW-	R--	---	---	RW-	R--	R--
"urn:o-ran:operations: x.y "	RWX	RW-	R--	---	RWX	RWX	R--
"urn:o-ran:uplane-conf: x.y "	RWX	RWX	R--	---	R--	RWX	RWX (NOTE 6)
"urn:o-ran:beamforming: x.y"	R-X	R-X	R--	---	R--	R-X	R--
"urn:o-ran:lbm: x.y "	RW-	RW-	R--	---	RW-	R--	R--
"urn:o-ran:software-management: x.y "	R-X	R-X	R--	R-X	R-X	R--	R--
"urn:o-ran:file-management: x.y "	R-X	R-X	R-X	---	R-X	---	---
" urn:o-ran:message5: x.y "	RW-	R--	R--	---	R--	RW-	R— (NOTE 6)
"urn:o-ran:performance-management: x.y "	RWX	RWX	RWX	---	RWX	R-X	R-X (NOTE 6)
"urn:o-ran:transceiver: x.y "	RW-	RW-	R--	---	RW-	R--	R--
"urn:o-ran:externalio: x.y "	RWX	RWX	---	---	RWX	R--	R--
"urn:o-ran:ald-port: x.y "	RWX	RWX	---	---	RWX (NOTE 3)	RWX	---
"urn:o-ran:interfaces: x.y "	RWX	RWX	R--	---	RWX	R--	R--
"urn:ietf:params:xml:ns:yang:ietf-ip"	RW-	RW-	R--	---	RW-	R--	R--
"urn:ietf:params:xml:ns:yang:ietf-interfaces"	RW-	RW-	R--	---	RW-	R--	R--
"urn:o-ran:processing-elements: x.y "	RW-	RW-	R--	---	RW-	RW-	R-- (NOTE 6)
" urn:o-ran:mplane-interfaces: x.y "	RW-	RW- (NOTE 2)	R--	---	RW-	R--	R--
"urn:o-ran:dhcp: x.y "	R--	R--	R--	---	R--	R--	R--
"urn:o-ran:ald: x.y"	--X	---	---	---	--X (NOTE 3)	--X	---
"urn:o-ran:troubleshooting: x.y"	R-X	R-X	R-X	---	R-X	---	---
" urn:o-ran:trace: x.y"	R-X	R-X	R-X	---	R-X	---	---

Module Rules	sudo	nms	fm-pm	swm	smo	hybrid-odu	carrier
"urn:o-ran:laa: x.y "	RW-	RW-	---	---	R--	RW-	R--
"urn:o-ran:laa-operations: x.y "	R-X	---	---	---	---	R-X	---
"urn:o-ran:antcal: x.y "	RWX	R--	---	---	R--	RWX	R-X
"urn:ietf:params:xml:ns:yang:ietf-netconf-acm"	RW-	R--	R--	R--	RW-	RW-	R--
"urn:ietf:params:xml:ns:yang:ietf-yang-library"	R-X	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	R-X
"urn:ietf:params:xml:ns:yang:ietf-netconf-notifications"	R-X	R-X	R-X	R-X	R-X	R-X	`R-X
"urn:o-ran:shared-cell:x.y"	RW-	RW-	---	---	R--	RW-	R-- (NOTE 6)
"urn:o-ran:ethernet-fwd:x.y"	RW-	RW-	---	---	RW-	R--	R--
"urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications"	---	---	---	---	RWX	---	---
"urn:o-ran:ves-sn:x.y"	---	---	---	---	RW-	---	---
"urn:ieee:std:802.1X:yang:ieee802-dot1x"	RW-	RW-	R--	---	RW-	R--	R--
"urn:ieee:std:802.1Q:yang:ieee802-dot1q-cfm"	RW-	RW-	R--	---	RW-	R--	R--
"urn:o-ran:o-ran-ieee802-dot1q-cfm"	RW-	RW-	R--	---	RW-	R--	R--
"urn:ietf:params:xml:ns:yang:ietf-system"	RW- (NOTE 5)	RW- (NOTE 5)	R— (NOTE 5)	---	RW- (NOTE 5)	R-- (NOTE 5)	R— (NOTE 6)
"urn:ietf:params:xml:ns:yang:ietf-truststore"	R-- (NOTE 9)	R-- (NOTE 9)	---	---	R-- (NOTE 9)	R-- (NOTE 9)	R-- (NOTE 9)

NOTE 1: The rule list for "urn:o-ran:user-mgmt:1.0" shall additionally deny reading of the password leaf by any NETCONF client

NOTE 2: 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 clause 14.4.

NOTE 4: The rule list for "urn:o-ran:supervision:x.y" shall additionally deny writing of the **cu-plane-monitoring** container for NETCONF sessions with "smo" group privileges.

NOTE 5: The rule list for "urn:ietf:params:xml:ns:yang:ietf-system" shall additionally deny write access to the **clock** container and the **authentication** container and deny read access to the **system-state** container for all group privileges.

NOTE 6: Clause 19 describes further details of how carrier privileges are refined based on **sro-id** for O-RUs that support the **SHARED-ORU-MULTI-OPERATOR** feature.

NOTE 7: The carrier rule list prohibits read and write to o-ran-usermgmt YANG model, but allows access to the **sro-id** leaf instance defined in o-ran-usermgmt from different data structures by reference.

NOTE 8: Execution access privilege applies to chg-password RPC only.

NOTE 9: The present document only supports CMPv2 as the method to configure a trust anchor. An O-RU controller can use the ietf-truststore model to read the operator and/or manufacturer installed trust anchors.

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**.

6.6 NETCONF capability discovery

The O-RU shall advertise its NETCONF capabilities in the NETCONF Hello message. The Hello message sent by the O-RU shall include the <session-id> element containing the session ID for the NETCONF session and shall provide 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 are specified in clause 8 of RFC 6241 [3] and include the following capability items:

- 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

O-RUs shall support the "XPATH capability".

O-RUs shall support "Notifications".

O-RUs shall support at least one of the following capabilities:

- "writable-running capability"
- "candidate configuration capability and associated Commit operation".

In addition, an O-RU that supports "writable-running capability" should support the "rollback on error capability" and an O-RU that supports "candidate configuration capability and associated Commit operation" should support the "confirmed commit capability".

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 clause 9 for more information on NETCONF based configuration management.

In order to avoid interactions between the operation of supervision watchdog timer (see clause 6.7) and the confirmed commit timer (default value set to 600 seconds in RFC 6241), when using the NETCONF confirmed commit capability, a NETCONF 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).

6.7 Monitoring NETCONF connectivity

This clause provides description of NETCONF connectivity monitoring for persistent NETCONF session. Additional procedures for O-RUs that support the optional NON-PERSISTENT-MPLANE feature to monitor the communication path between the O-RU and Event-Collector are defined in clause 18.6.

When having a session with a NETCONF client that has subscribed to receive the **supervision-notification**, the O-RU operates watchdog timers (supervision timer and notification timer) to ensure that the session to the NETCONF client is persistent, as illustrated in Figure 6.7.1. The O-RU provides NETCONF Notifications to indicate to remote systems that its management system is operational.

For the O-RU that supports feature SUPERVISION-WITH-SESSION-ID, these **supervision-notification** messages shall also indicate the NETCONF **session-id** associated with the subscription to the event notification. When subscribing to receive the **supervision-notification**, a NETCONF client may use its own NETCONF session-id in the subscription filter criteria to indicate to the O-RU which **supervision-notification** events the O-RU shall forward to the NETCONF client. The session-id is provided by the NETCONF server to the NETCONF client in the initial Hello exchange, see clause 6.6.

An O-RU controller that has subscribed to the supervision-notification is expected to use the <supervision-watchdog-reset> RPC to indicate to O-RU the O-RU controller is operational.

NOTE 1: This supervision is intended to be used with the NETCONF client associated with the operation of the peer to the O-RU's lower layer split and clause 6.5 describes which NETCONF clients have privileges to subscribe to the **supervision-notification**.

A NETCONF server shall support the operation of individual supervision watchdog timers for each NETCONF client which has subscribed to **supervision-notification**.

The privileged NETCONF client is responsible for automatically enabling the operation of the watchdog timers by creating supervision-notification subscription. After operation of watchdog timers is enabled - the timers are considered as running.

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 value: 10s)

  Operation: The O-RU identifies supervision failure operation when the timer expires. To avoid supervision timer expiration, a NETCONF client who has subscribed to receive the **supervision-notification** should repeatedly reset this supervision timer. Such supervision timer reset is considered by O-RU as confirmation that NETCONF connectivity to the O-RU Controller is operational.

The O-RU enables dedicated watchdog timers for specific NETCONF client when it receives a <create-subscription> RPC from a NETCONF client with required privileges. The notification timer shall be started 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 who has subscribed to the **supervision-notification** shall be prepared to receive the notification at any time when the watchdog timers are running.

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 2: **next-update-at** is just informative.

In the **supervision-watchdog-reset** RPC, the NETCONF client may configure new values for the watchdog timers using RPC parameters "supervision-notification-interval" and "guard-timer-overhead. When the O-RU receives the **supervision-watchdog-reset** RPC, it is responsible for resetting its supervision timer and notification timer. When the watchdog timers are running, the O-RU shall be prepared to receive supervision-watchdog-reset RPC at any time - also within supervision timer period.

The NETCONF client can set new value of watchdog timers without receiving **supervision-notification** from the O-RU. The new values are 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.

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 shall enter "supervision failure" condition, as described in clause 14. If all NETCONF sessions to NETCONF clients with "sudo" privileges are closed, the O-RU shall immediately disable operation of the supervision timer.

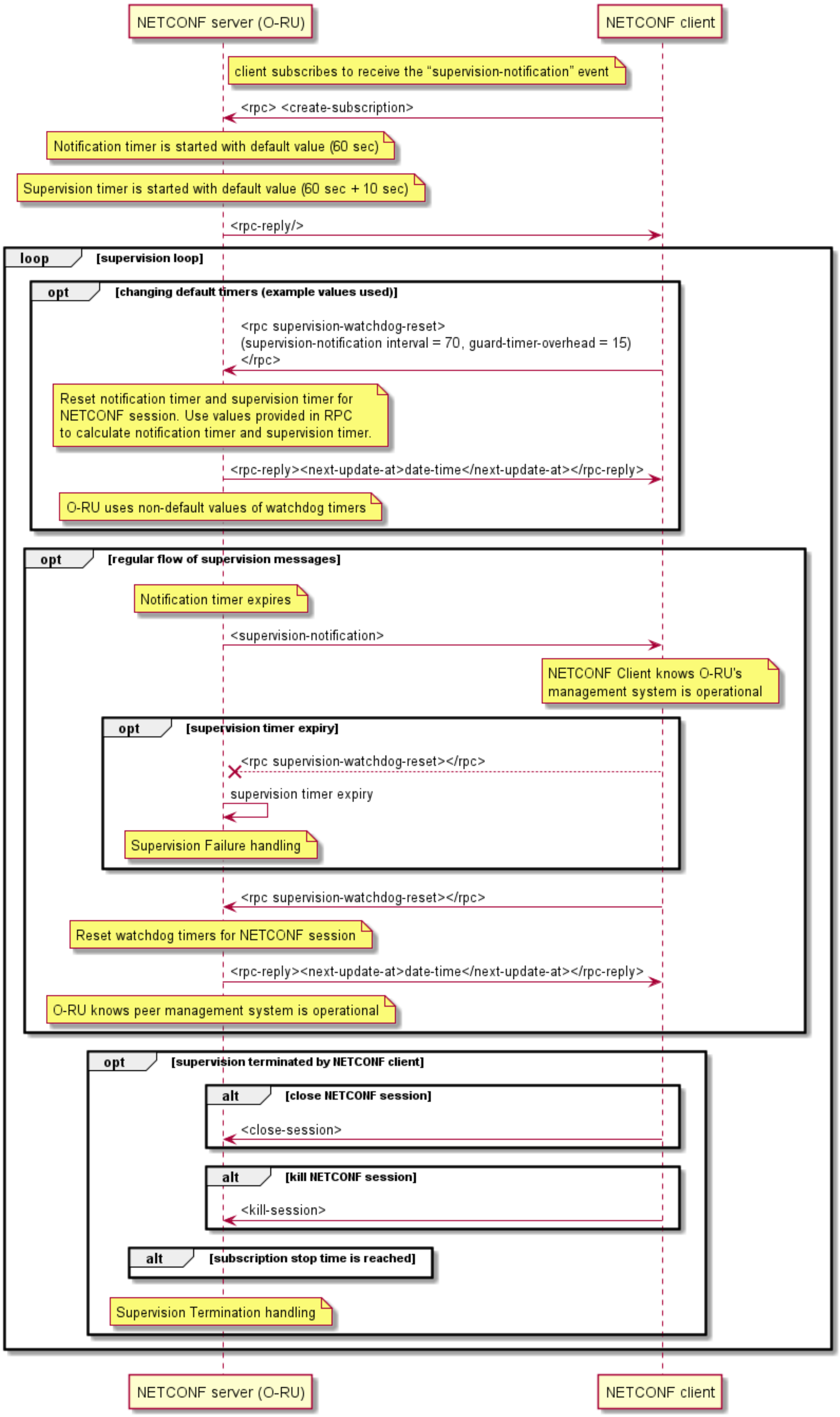


Table 6.7-1: Monitoring NETCONF Connectivity

NOTE 3: A NETCONF client can use the **create-subscription** for the single **supervision-notification** event stream, or alternatively subscribe to the default event-stream using additional filter criteria to identify the supervision notification. In order to subscribe to multiple notifications, the appropriate **create-subscription** message is required. Please refer to clause 11.3 for the appropriate example of **create-subscription** of multiple notifications.

The figure illustrates the O-RU ceasing supervision operation triggered by two options:

1) The supervision timer expires. In such case the O-RU performs Supervision Failure handling as described in clause 14.1.1.

2) The NETCONF client terminates the subscription to the **supervision-notification.** The NETCONF client can either close the subscription session, terminate the NETCONF session or wait for the subscription stop time to be reached. In such case the O-RU performs Supervision Termination handling as described in clause 14.1.2

6.8 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. If the NETCONF client is a Call home O-RU controller, the O-RU shall then re-commence call home procedures, as described in clause 6.3.

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 for the NETCONF session. NETCONF clients associated with other privilege groups are not required to operate using persistent NETCONF sessions.

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.

6.9 PNF registration

6.9.1 Introduction

The support by an O-RU of PNF Registration to a discovered Event-Collector is optional and hence clause 6.9 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 clause 18.6.

6.9.2 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 clause 6.2.7. 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. These procedures are illustrated in Figure 6.9.2-1.

An O-RU that is performing the PNF registration procedure whilst simultaneously performing the call home procedure described in clause 6.3, shall be able to determine that the SMO has established a NETCONF session with the O-RU. This is identified by the O-RU analysing 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 call home O-RU Controller(s) to which the O-RU is simultaneously performing the call home procedure.

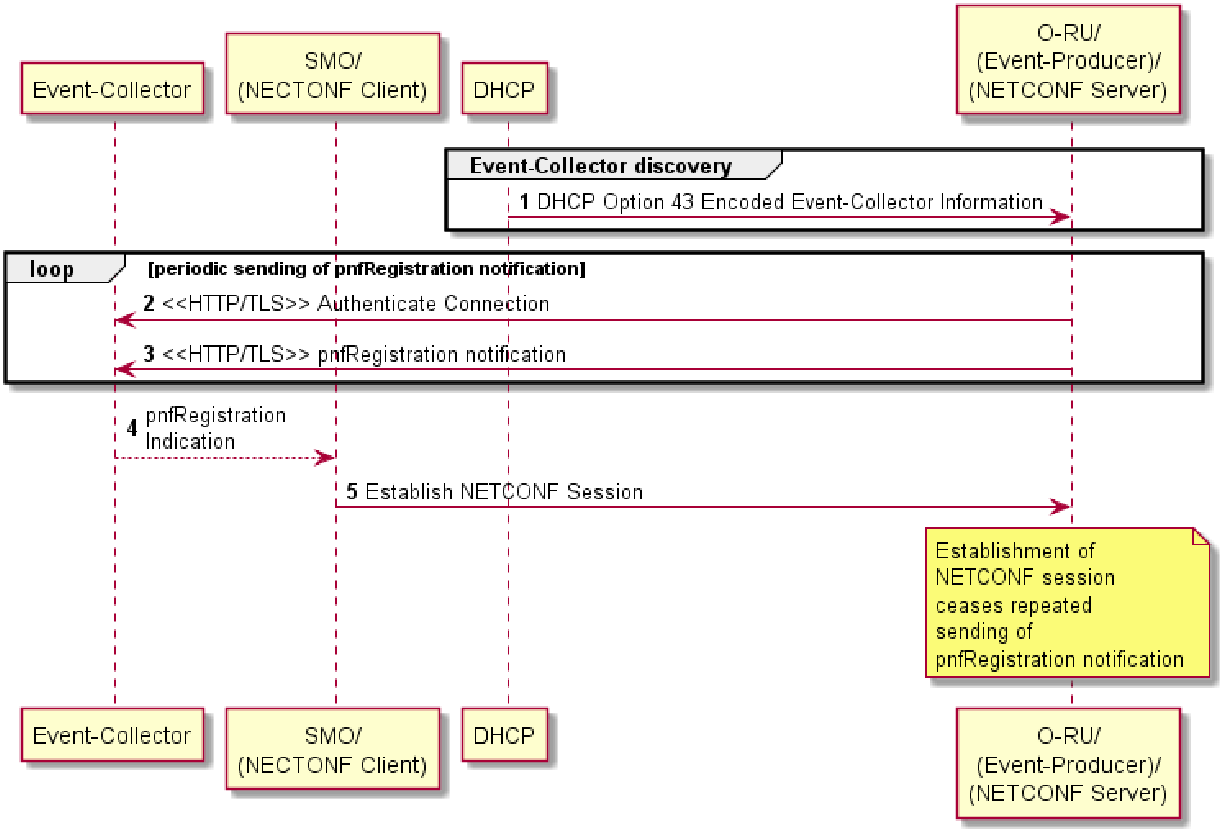


Figure 6.9.2-1: PNF Registration Procedure

6.9.3 Encoding of PNF registration notification

In this version of the specification, the encoding of the pnfRegistration notification follows the ONAP definition [i.2].

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 using Table 6.9.3-1. 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.

Table 6.9.3-1: Mapping from O-RU's Operational Data to PnfRegistration fields

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

7 O-RU to O-DU interface management

7.1 O-RU interfaces

An O-RU has a number of network interfaces, including Ethernet, VLAN and IP interfaces. This section describes the management of these network 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. Additionally, for O-RUs that include the ieee802-dot1x and ietf-system models in their YANG library the /if:interfaces/if:interface/dot1x:pae/dot1x:port-capabilities/dot1x:supp schema node enables an O-RU Controller to configure operation of IEEE 802.1X Port based Access Control, as specified in [70].

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. The O-RU shall provide operational state associated with the layer 3 configuration of these interfaces provide additional detail of the layer 3 configuration, including prefix(es), using the ietf-ip YANG model, and domain name servers and default gateway addresses, using the o-ran-dhcp YANG model. An example of such operational state for the ietf-ip YANG model is shown in Figure 7.1-1.

```
<ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
  <enabled>true</enabled>
  <address>
    <ip>10.10.0.17</ip>
    <netmask>255.255.255.0</netmask>
    <origin>dhcp</origin>
  </address>
</ipv4>
```

Figure 7.1-1: Example operational data related to an IP interface

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.

7.2 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**. Interfaces accessible through the transceiver are provided by the **interface-name**, **interface-names** and **additional-multi-lane-reporting/interface-names** leaves in the o-ran-transceiver YANG module.

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 clause 9 for more details.

With QSFP form factor, the optical links may be multi-wavelengths (4xTx & 4xRx) and/or multi-fibres (MPO - Multifibre Parallel Optic). The QSFP digital diagnostic interface as specified in SFF-8636 [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 start-up. 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 clause 12. 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: File Upload procedure requesting non-existing file will 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.

7.3 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 shall be enabled on all interfaces. This permits an O-RU to autonomously discover that it is connected to a trunk port, as described in clause 6.2.3.

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 shall 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.

7.4 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 clause 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 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 as specified in RFC 2131 [10].

and when the O-RU supports IPv6:

2) IPv6 Stateless Address Auto-Configuration (SLAAC) as specified in RFC 4862 [11].

OR

3) IPv6 State-full address configuration uses DHCPv6 as specified in RFC 8415 [58].

7.5 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, as specified in [2] clause 5.4 or delay measurement operation, as specified in [2] clause 4.4.3.3. 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.

If the O-RU reports the capability of supporting feature MULTIPLE-TRANSPORT-SESSION-TYPE, which indicates the O-RU supports multiple transport-session-type, the NETCONF client may configure **processing-elements** with more than one **transport-session-type** to the O-RU, using the list **additional-transport-session-type-elements** with the key **transport-session-type**. When a processing element configured by the list **additional-transport-session-type-elements** is referenced by other modules, the **transport-session-type** shall also be configured as a key to the list **additional-transport-session-type-elements**.

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.

7.6 O-DU verification of C/U-Plane transport connectivity

7.6.1 C/U-Plane transport connectivity verification

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 C/U-Plane end-to-end transport connectivity between the O-DU and O-RU, the O-RU shall support C/U-Plane connectivity verification capabilities using a request/reply function, as illustrated in Figure 7.6.1-1.

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

- During O-RU configuration, to validate the transport configuration
- At runtime to regularly perform bi-directional network connectivity verification

The periodicity for bi-directional connectivity verification is usually between 1 and 60 seconds.

Two different network protocols are defined for performing the bi-directional transport connectivity verification procedure:

- For C/U sessions over Ethernet: Loop-back Protocol (LB/LBM) as specified in IEEE 802.1Q-2018 [69].
- When the O-RU is configured to support C/U sessions over IP: UDP echo, as specified in RFC 862 [18].

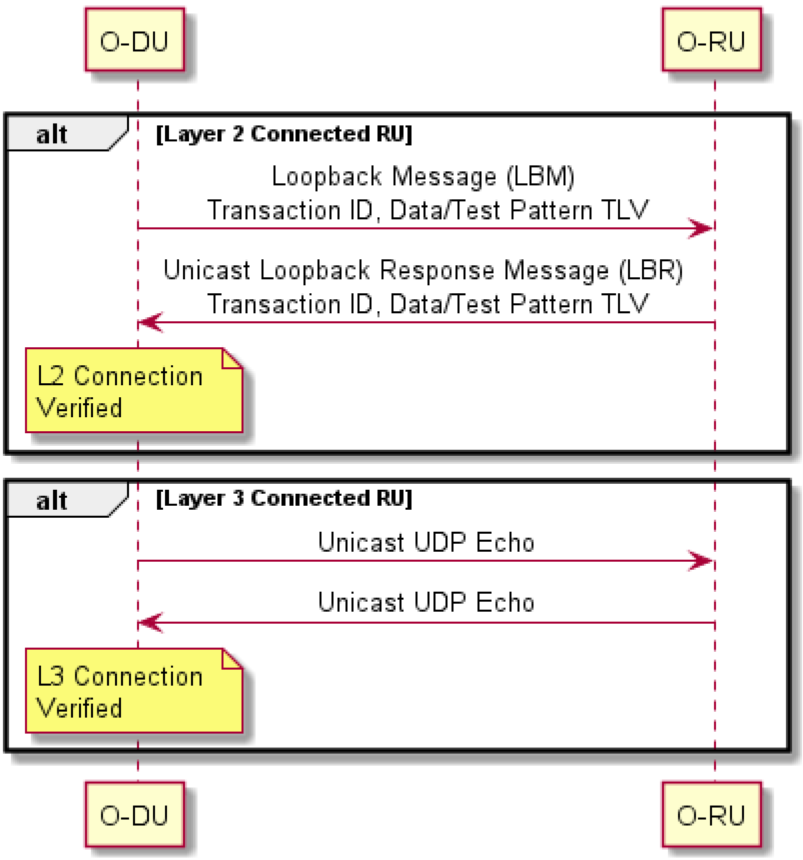


Figure 7.6.1-1: C/U Plane Transport Connectivity Verification

7.6.2 Ethernet connectivity monitoring procedure

7.6.2.1 Transport connectivity verification monitoring Procedure

When the O-RU and O-DU are operating their C/U sessions transported over Ethernet, the transport connectivity verification checks operate at the Ethernet layer. The protocol for Ethernet connectivity verification shall use the Loop-back Protocol as defined by IEEE 802.1Q-2018 [69].

For the purpose of regular connectivity verification, all C/U -plane messaging endpoints in the fronthaul network shall be configured to be part of the same Maintenance Entity (ME). The O-RU Controller shall configure each O-RU, assigning them 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 interface of the O-RU responder. If administratively initiated, the O-DU shall send the LBM request. When sending the LBM request, support for setting the destination MAC address as the unicast MAC address of the specified O-RU Ethernet interface as per IEEE 802.1Q-2018 [67] is mandatory while the support for setting the destination MAC address as the group destination MAC address 01-80-C2-00-00-3x as specified in ITU-T Y.1731 [25] clause 7.2.2, where the LSB 4 bits are derived from the md-level configuration in o-ran-lbm.yang, is optional. See also clause 10.1 of ITU-T Y.1731 [25].

An O-RU that supports LBM configuration using the ieee802-dot1q-cfm YANG model shall include the ieee802-dot1q-cfm model in its YANG library. Such an O-RU can have a common LBM configuration datastore and operational data, exposed by both ieee802-dot1q-cfm and o-ran-lbm YANG models.

For example, an O-RU can ensure the value of **replies-transmitted** in o-ran-lbm YANG model to be set as identical to that value of **mep-lbr-out** in the ieee802-dot1q-cfm YANG model.

In case the coordination of LBM configuration happens external to the O-RU, when LBM configuration is inconsistent between two models, the LBM behavior of the O-RU is not specified by the present document.

O-RUs that supports ieee802-dot1q-cfm.yang shall support the Notification of Updates to its Configuration Datastore functionality, as specified in clause 9.4. Hence, any NETCONF clients connected to the O-RU may configure subscriptions to receive notifications of modifications to ieee802-dot1q-cfm.yang and/or o-ran-lbm.yang configuration changes.

7.6.2.2 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.

The O-RU shall be able to respond to Loopback Messages received from different remote Maintenance Association Endpoints, when the destination MAC address is the O-RU's MAC address. The O-RU should respond to Loopback Messages received from different remote Maintenance Association Endpoints, when the destination MAC address is a group destination MAC address as defined in clause 10.1 of [25].

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.

7.6.2.3 Monitor network connectivity

After the procedure described in clause 7.6.2.2 has been executed successfully, a further procedure may be executed continuously to maintain the connectivity status. To achieve this the O-DU can continuely 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.

Optionally, for those O-RUs that advertise support for the ieee802-dot1q-cfm.yang, the O-DU can enable O-RU sending of Connectivity Check Messages (CCM) by configuring **ccm-enabled** to "true" on the O-RU. The O-DU can monitor network connectivity by received CCM message from the O-RU. These CCM messages are sent to a multicast destination address which encodes the maintenance domain level as specified in Table 8-18 of IEEE 802.1Q[67].

7.6.2.4 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 clause 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 [i.3]. 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-2018 [69].

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

7.6.3 IP connectivity monitoring procedure

7.6.3.1 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, as specified in 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.

7.6.3.2 Managing IP connectivity monitoring procedure

This clause 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.

7.7 C/U-Plane delay management

7.7.1 Introduction

The Intra-PHY lower layer fronthaul split has the characteristic of a stringent bandwidth and tight latency requirement. The CUS-Plane specification [2] clause 4.4 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 clause describes the procedures that are used to manage the delay parameters for the fronthaul split.

7.7.2 Delay parameters

The reference points for delay management are specified in the CUS-Plane Specification [2], clause 4.4 and clause 4.7. 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. Other parameters related to delays to external antennas may vary according to specific tx/rx-array-carriers. 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.
- Tda: Corresponding to the time difference between the output of DL signal at the antenna connector of O-RU and the transmission over the air.

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.
- Tau: Corresponding to the time difference between the reception over the air and the input of UL signal at the antenna connector of O-RU.

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 start-up phase.

7.7.3 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.3 for information on reception window counters.

7.7.4 External antenna delay control

An O-RU may optionally support the external antenna delay control by indicating that it supports the **EXT-ANT-DELAY-CONTROL** feature in its o-ran-wg4-features YANG model. Such an O-RU uses the **ext-ant-delay-capability** schema node in o-ran-module-cap YANG model to indicate to the O-RU Controller the type of external delay configuration supported by the O-RU:

- PER-O-RU: The O-RU only supports a single value of t-da-offset and a single value of t-au-offset across all tx-array-carriers and rx-array-carriers respectively.
- PER-ARRAY: The O-RU supports separate values of t-da-offset and t-au-offset across individual tx-arrays and rx-arrays respectively.
- PER-ARRAY-CARRIER: The O-RU supports separate values of t-da-offset and t-au-offset across separate tx-array-carriers and rx-array-carriers respectively.

7.8 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 procedure operated by the O-DU or by other techniques. This clause 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 clause 9.2 and Annex C indicating that it supports the ADAPTIVE-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

As per [2] clause 4.7, the O-RU Controller shall configure values of **t12-max** and **t34-max** as if there is no external delay, i.e., Tau = Tda = 0 (zero). 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 should 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 shall include the newly adapted timing values when signalling its delay parameters to a NETCONF client.

7.9 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. If O-RU supports measured transport delay, it shall implement the protocol as described in the CUS-Plane Specification [2] clause 4.4.3.3.

An O-RU that supports the eCPRI based delay measurement capability, shall be able to support the operation of delay measurements whenever any processing element has been configured as described in clause 7.5. For each processing element configured, the O-RU shall be able to respond to any messages when received and keep a record of the number of responses, requests and follow-up messages transmitted by the O-RU.

7.10 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.

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 is advisable for the NETCONF client to initially disable the operation of the timer before carrier activation. Such an approach avoids 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 NETCONF client can re-configure the timer with the desired value and hence activate monitoring of the C/U plane connectivity by the O-RU.

7.11 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.

7.12 IEEE 802.1X port based access control

7.12.1 Configuring port based access control

The O-RU shall support IEEE 802.1X port based access control supplicant functionality.

NOTE 1: IEEE 802.1X port based authentication is useful in those deployment scenarios where the O-RU is connected to an Ethernet bridged network or IP routed network used to transport the fronthaul traffic, for example as typically used in hybrid deployment scenarios.

NOTE 2: A previous version of this document did not mandate IEEE 802.1X port-based network access control supplicant functionality in the O-RU. Operators that enable IEEE 802.1X authenticator PAE functionality in the fronthaul need to accommodate legacy O-RUs that do not support supplicant PAE functionality.

NOTE 3: The present document does not define operation of an authenticator PAE in the O-RU. When operating an O-RU in shared cell configuration (i.e., in FHM or cascade mode), operators wanting to benefit from the port based access control supplicant functionality in an O-RU will need to rely on the transport network equipment to implement the authenticator PAE functionality. Clause 17.4 of O-RAN Xhaul Packet Switched Architectures and Solutions [73], requires each port connection from the transport network equipment towards the O-RU supports IEEE 802.1X authenticator functionality.

O-RUs indicate they support IEEE 802.1X port based access control by including the ieee802-dot1x and ietf-system models in their YANG library. An O-RU Controller can use the **/sys:system/dot1x:pae-system/dot1x:system-access-control** parameter to enable or disable operation of IEEE 802.1X port based access control. Once configured, an O-RU shall store the system-access-control parameter in reset persistent memory. This means that, unless the O-RU has performed a factory reset procedure as described in clause 8.8, the O-RU shall always use the last configured value of system-access-control during its next start-up procedure.

NOTE 4: The leaf **sys:system/dot1x:pae-system/dot1x:system-access-control** in ieee802-dot1x YANG model does not have a default value and hence IEEE 802.1X operation will normally be disabled in factory software. An operator and vendor can agree that an O-RU uses a default value for this leaf in factory software.

The configuration parameters and operational data related to IEEE 802.1X are defined in the **pae** container defined in the ieee802-dot1x YANG model defined augmentation of ietf-interfaces YANG model. An O-RU uses the port-capabilities container to indicate to an O-RU controller its PAE capabilities on a per-port basis. An O-RU indicates it supports a supplicant on a particular port by setting the supp leaf to true. When an O-RU indicates it supports a supplicant, an O-RU controller can use the supplicant container to configure authentication timers and re-try behaviour. The O-RU shall store this supplicant configuration in reset persistent memory.

7.12.2 EAP authentication

The operation of a successful EAP authentication is described O-RAN Security Requirements Specifications [71], clause 3.2.5.5.3. An O-RU that supports IEEE 802.1X and which has its supplicant functionality enabled shall support EAP-TLS authentication. An O-RU that has a valid operator-signed certificate shall use the certificate in the EAP-TLS authentication. An O-RU that does not have a valid operator-signed certificate shall use its manufacturer installed certificate in the EAP-TLS authentication.

All O-RUs that have a supplicant enabled shall set their unauthAllowed variable to AuthFail, as specified in IEEE 802.1X, clause 12.5.1 of [70]. Accordingly, the O-RU's PAE controlled port state machine shall enable unauthenticated connectivity on a port with supplicant enabled only after the EAP authentication procedure has failed.

NOTE: The PAE authenticator can decide to permit unauthenticated connectivity from an O-RU based on operator determined policy, e.g., based on matching the source Ethernet MAC address used by the Supplicant PAE against a register of O-RU MAC addresses. Such policies and their enforcement are outside the scope of the present document.

7.12.3 Certificate time validation

Operation of the EAP-TLS method requires the O-RU verify the certificate chain presented by the authenticator PAE, including confirming the certificate's validity periods. O-RUs using IEEE 802.1X port based access control shall be able to verify the certificate validity periods during the EAP-TLS exchange, e.g., by employing a persistent clock or using GNSS-based time synchronization .

If the O-RU cannot verify the certificates validity, e.g., because a persistent clock has failed, the EAP authentication procedure will fail. According to clause 7.12.2, this will result in the O-RU providing un-authenticated connectivity. If the O-RU uses a persistent clock for certificate validity checking and detects a persistent clock failure, it shall, as soon as possible, set its clock to the current time recovered from available external sources, e.g., PTP or GNSS.

NOTE: The scenario corresponding to a failed clock source and resulting O-RU operation using un-authenticated connectivity, is identical to the O-RU operation prior to the introduction of IEEE 802.1X port based access control in version 10 of the present document.

8 Software management

8.1 General

8.1.1 Introduction

The Software Management function provides a set of operations allowing the desired software package or build to be downloaded, files to be installed and slot containing installed software to be activated at O-RU.

There are two types of software management processing. Some O-RUs expect a software package (in form of a zip file) to be downloaded to O-RU, while other O-RUs expect individual files in a build to be downloaded.

The O-RUs that do not support download of an archived package and require downloading of individual files in a build can be distinguished by the existence of data node **build-content-download**. For such O-RU, the O-RU controller shall provide remote-file-path of file in the desired build as input for **software-download** RPC. Refer to clause 8.5 for details.

If the **build-content-download** is not exposed by an O-RU, the O-RU controller shall provide remote-file-path of the software package zip file as input for **software-download** RPC.

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.

The software management functions involve the O-RU controller subscribing to receive particular YANG notifications from the O-RU. All O-RUs support the NETCONF Create-Subscription method, enabling those notifications to be transported using NETCONF notifications. In addition, those O-RUs that support the optional NON-PERSISTENT-MPLANE feature, the O-RU Controller can create a configured subscription from the O-RU, enabling those notifications to be transported over HTTPS to an Event-Collector as described in clause 18.

It is also Vendor's responsibility to handle SW Build / package / file integrity check. The O-RU controller shall be able to obtain information contained in the manifest.xml.

8.1.2 Software slots

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. If the read-only slot is supported by an O-RU, the software that is present in the read only slot is termed factory default software and can be used in factory reset procedure. Refer to clause 8.9 for detailed information. 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.

8.1.3 Software management procedures

Software management procedures are described in clause 8.4, clause 8.5 and clause 8.6. Procedures used in Software Management are covered by o-ran-software-management.yang module.

8.2 Software build

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. Whether a file is included in a build is defined in manifest.xml. The manifest.xml enables mapping from radio product type to files. Revision and name of the build in each slot is provided as result by the inventory procedure to ensure visibility of radio software installation status.

The use of compression and ciphering for the content of the software build is left to vendor implementation.

8.3 Software package

8.3.1 Software package name

The name of a software package shall conform to the following format:

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

Where:

- *Vendor Code* is a mandatory part which either has two capital characters or 1-5 digits. The vendor code prefix in the software package file name is used to avoid conflict between file names of SW packages provided by different vendors,
- *Vendor Specific Field* is any set of characters allowed in filename. The value shall 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 software package is split into multiple files, e.g., when a zipped file is split into multi-part zip file – number after "#" indicates the part number of a split file. Numbering starts from 1 and shall be continuous,
- *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. <EXT > in the file name shall be ".zip". The O-RUs that do not expose **build-content-download** shall support ZIP functionality. For O-RUs that expose **build-content-download**, support of ZIP functionality is implementation dependent.

8.3.2 File server storage of software package

The operator needs to manage and control which O-RU files will be stored and used in the file server. The software package can be stored on the file sever as one or more archived file(s) or as multiple files that are extracted from a software package zip file. The O-RU controller triggers the O-RU to download software file from a file server and should ensure that all the files necessary for the O-RU are transferred from the file server to the O-RU.

8.3.3 Software package content and manifest.xml file

Each software package includes:

- manifest.xml
- software files

NOTE: Those software files are intended to be installed on software slot. Those software files are intended to be used for one or several O-RU products.

An example of archive structure for the software files is:

```
../o-ru-sw/version1.0/xxxx
../o-ru-sw/version1.0/yyyy
../o-ru-sw/version2.0/yyyy
../o-ru-sw/version1.5/zzzz
```

The content of the manifest.xml file allows to maintain software update process correctly in terms of compatibility between O-RU hardware and software files to be downloaded. The content of the manifest.xml file enables the O-RU controller to trigger the installation of files of a software build designed for device based on different product code. The manifest.xml file is contained in a software package and shall not be ciphered. The format of the manifest.xml file is illustrated below using example data:

```
<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 – SHALL 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 and corresponds to attribute build-Id in the tag
"product".
        /// @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)
            /// @fileVersion may be used as a handle to identify a specific path used when archiving
different file versions
            /// @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 check 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>
```

Keywords in manifest.xml example are in **bold**, the keywords shall be strictly followed.

Correspondence between content of manifest.xml tags, their attributes and content of o-ran-software-management.yang is:

- XML tag "product", attribute "vendor" corresponds to content leaf "vendor-code",
- XML tag "product", attribute "code" corresponds to content of leaf "product-code",
- XML tag "product", attribute "build-Id" corresponds to content of leaf "build-id",
- XML tag "build", attribute "id" corresponds to leaf "build-id",
- XML tag "build", attribute "bldName" corresponds to content of leaf "build-name",
- XML tag "build", attribute "bldVersion" corresponds to content of leaf "build-version",
- XML tag "file", attribute "fileName" corresponds to content of leaf "name" in list "files",
- XML tag "file", attribute "fileVersion" corresponds to content of leaf "version" in list "files"

8.4 Software inventory

Pre-condition:

- M-Plane NETCONF session established.

Post-condition:

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

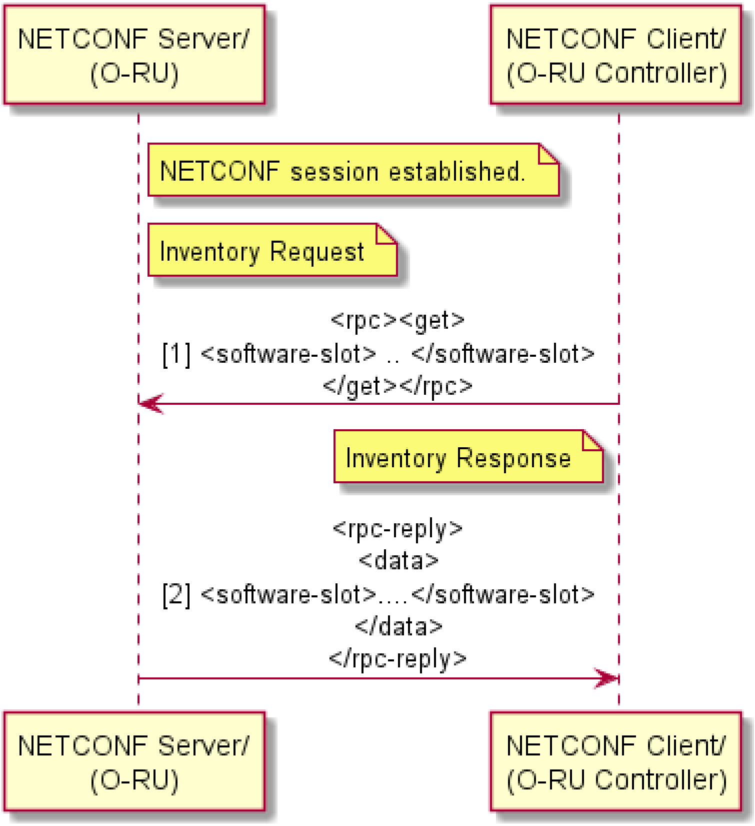


Figure 8.4-1: Inventory fetch call flow

Figure 8.4-1 illustrates the operation where 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 it's 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 slot. Status of the software in the slot can be

   - **VALID** - Slot contains a software build considered as proven valid.
   - **INVALID** - software build in the software slot cannot be used by O-RU. The software is considered by the O-RU as damaged (e.g., wrong CRC). Failed software install operation can cause a slot status to change to "Invalid".

     NETCONF client shall not activate a software slot that is invalid. Activation of a software slot containing an invalid software build shall be rejected by O-RU.

   - **EMPTY** - software slot does not contain any software files. NETCONF client shall not activate an empty software slot. Activation of an empty software slot shall be rejected by O-RU.

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

   - **True** - software slot is activated. Active::True can be assigned only for slots with status "Valid". At any time, only one slot in the O-RU shall be marked as active::True. The O-RU shall reject activation for software slots with status "Empty" and "Invalid".
   - **False** - software slot is not active.

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 the software slot

- **name** – name of one particular file
- **version** – version of the file
- **local-path** - complete path of the file on local file system
- **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 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.

The empty slot parameters shall be as follows

name: up to vendor, not empty

status: "EMPTY"

active: parameter does not exist

running: parameter does not exist

access: READ_WRITE

product-code: up to vendor

vendor code: up to vendor

build-name: null (empty string)

build-version: null (empty string)

files: empty list

8.5 Download

Pre-condition:

- M-Plane NETCONF session established.

- O-RU Controller has subscribed to receive **download-event** notifications.
- O-RU controller has validated presence of data node **build-content-download** at O-RU.

Post-condition:

- O-RU has downloaded the file specified and has successfully stored the downloaded file in the O-RU's file system.

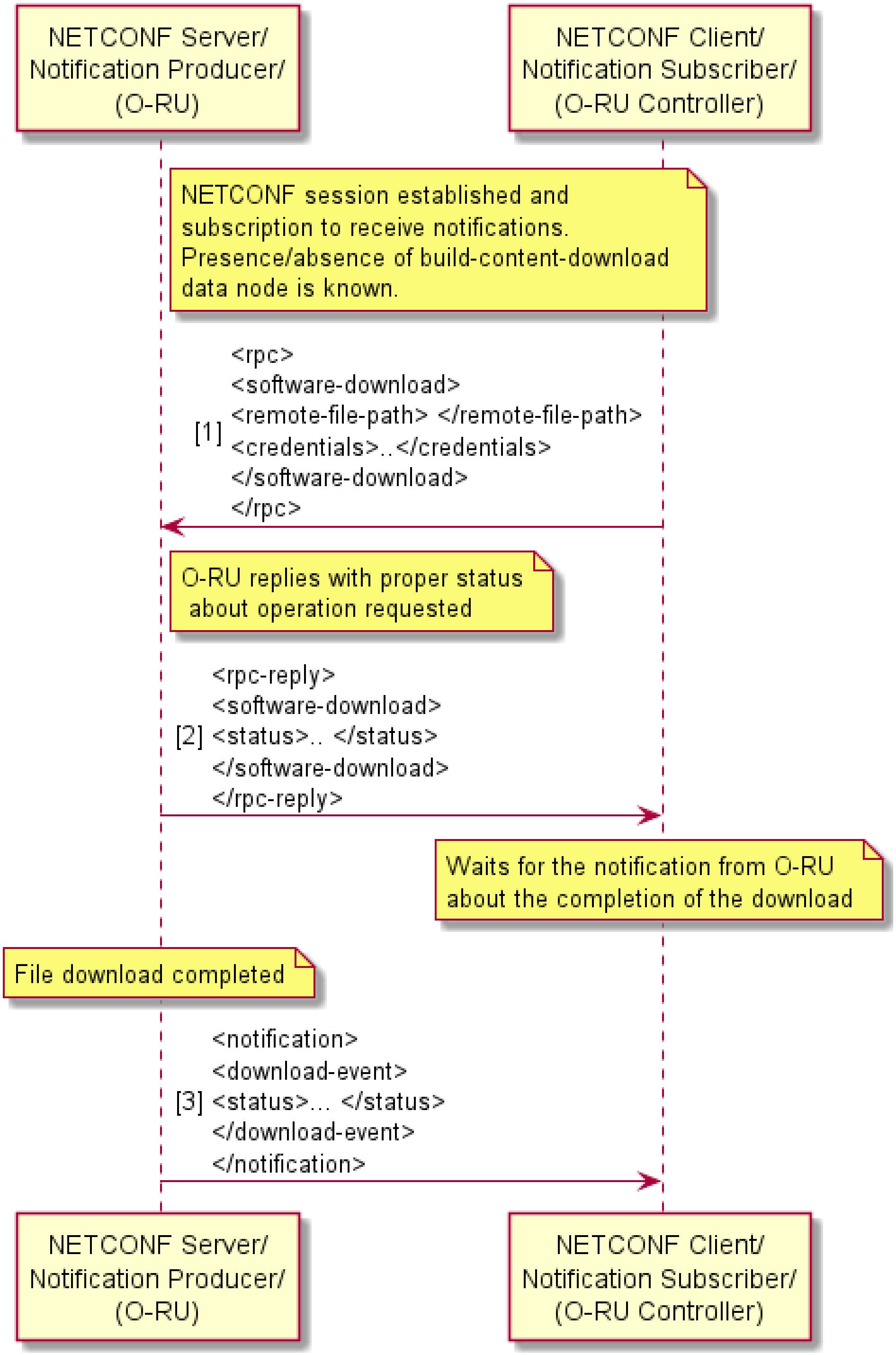


Figure 8.5-1: Software download call flow

Figure 8.5-1 illustrates the software download call flow. The following types of authentications shall be supported for **software-download**:

a) password for RU authentication and list of public keys (see clause 5.4 in the present document) for sFTP server authentication

Following types of authentications may be supported for **software-download**:

b) X.509 certificate for authentication of FTPES client (O-RU) and FTPES server

c) certificate for both O-RU and sFTP server authentication

The **software-download** RPC is used to trigger the downloading of software files to the O-RU. The download shall be performed using either sFTP or FTPES. Based on presence of data node **build-content-download**, O-RU controller provides either path to a file in a build or path to a software package file as input to **software-download** RPC. The RPC specifies the URI of the remote location of the software files using the remote-file-path leaf, where the URI scheme is used to signal whether to use sFTP or FTPES. An O-RU Controller shall only trigger software-download using FTPES if it is using NETCONF/TLS to configure the O-RU.

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

The O-RU downloads the file from file server. When the O-RU completes the software download or software download fails, the O-RU shall send the **download-event** notification with one of the following statuses:

a) COMPLETED

b) AUTHENTICATION_ERROR - source available, wrong credentials

c) PROTOCOL_ERROR – sFTP or FTPES 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)

g) INTEGRITY_ERROR – file is corrupted.

An O-RU that supports integrity check at download shall expose YANG feature “INTEGRITY-CHECK-AT-SW-DOWNLOAD”. O-DU can enable the feature by configuring type empty parameter **integrity-check-at-download-enabled.** O-RU may send download-event with status INTEGRITY_ERROR when **integrity-check-at-download-enabled** is presented.

NOTE: O-RU can obtain information of expected checksum either from manifest.xml or from information embedded in the file.

O-RU controller shall repeat the above procedure until all files which are required by the O-RU have been downloaded to the O-RU. To determine all files needed by O-RU, the O-RU Controller uses content of manifest.xml and relies on presence of data node **build-content-download**.

8.6 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 all files required by the O-RU are available in O-RU.
- O-RU Controller has subscribed to receive **install-event** notifications.

Post-condition:

- O-RU software file(s) is/are installed in the specified target **software-slot**.

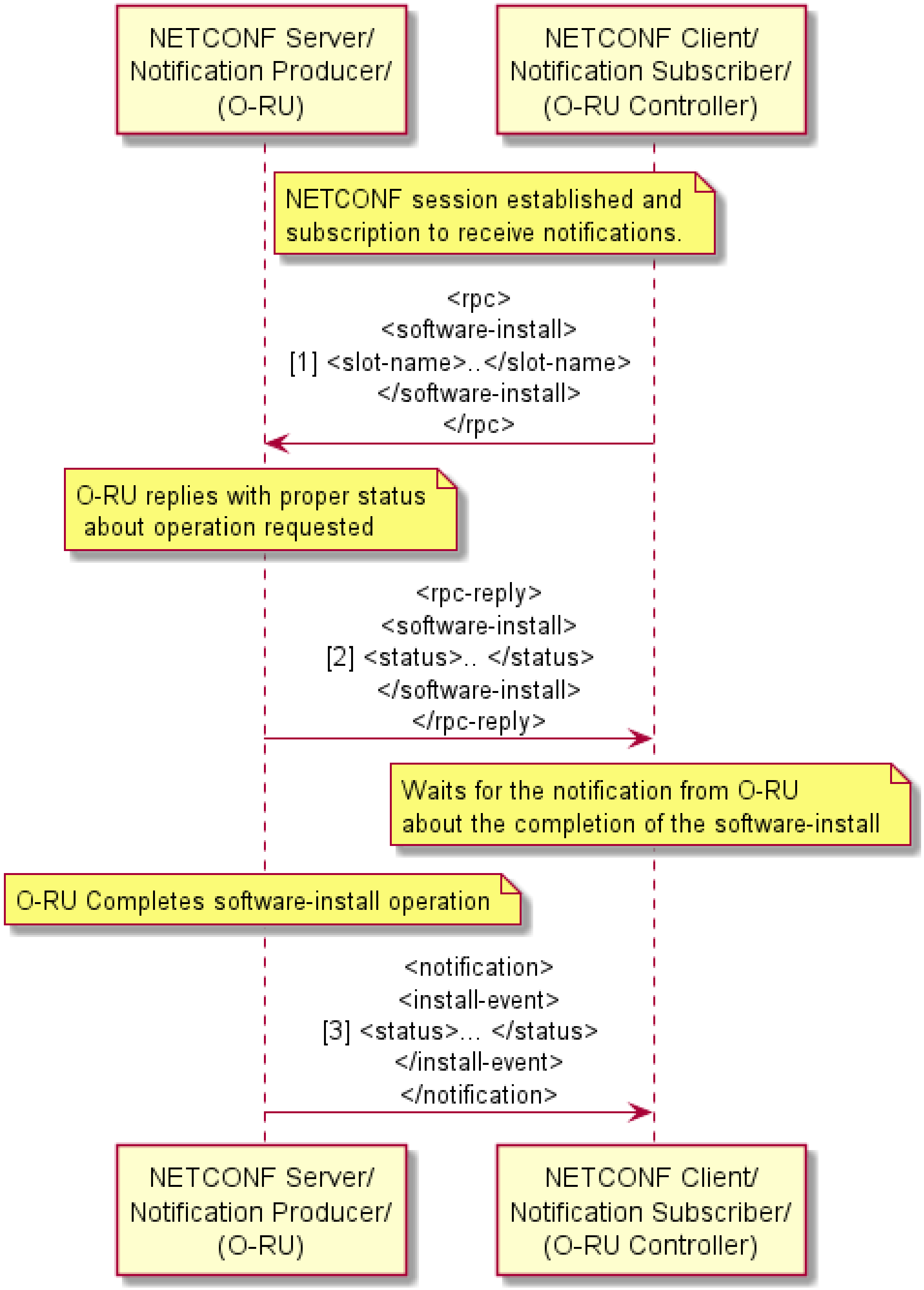


Figure 8.6-1: Software install call flow

Figure 8.6-1 illustrates the operation where the NETCONF **software-install** RPC is used to install the previously downloaded software (i.e., all necessary) to the specified target **software-slot** on O-RU**.**

The list of file-names in install-input shall not be empty. This slot shall 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 shall send the **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

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. O-RU shall not change the status of the slot when the install procedure is ongoing or when it is interrupted (e.g., by spurious reset operation).

8.7 Bringing software into operation

8.7.1 Procedure

Two steps are required to bring O-RU software into operation. The first step uses RPC **software-activate** to activate software in a slot. The second step, which may be delayed as per operator's decision until a suitable time, uses a **reset** RPC to take the activated software into operation.

8.7.2 Software 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:

- For activated slot the parameter **active** is set to TRUE. At the same time the parameter **active** for previously operational slot is set to FALSE by the O-RU. Parameter running is not changed for any of O-RU's software slots. This indicates that O-RU is still running software from previously active slot.

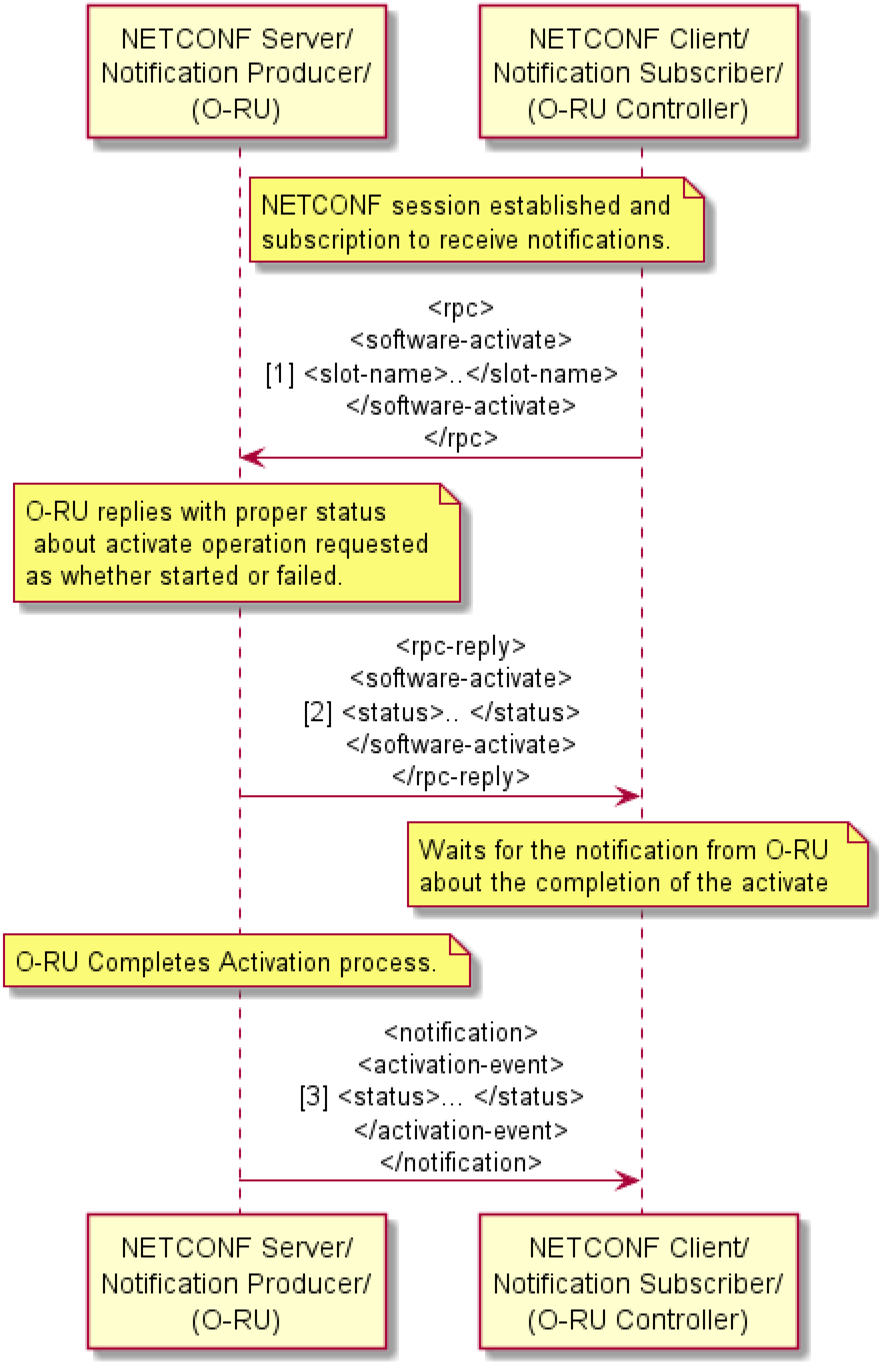


Figure 8.7.2-1: Software activation call flow.

Figure 8.7.2-1 illustrates the operations where the 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 **activation-event** notification with the status of activation. The following status is returned in the **activation-event** notifications.

a) COMPLETED - Activation procedure is successfully completed. O-RU can 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 sets **active** to true for the slot that was provided in the RPC and O-RU sets **active** to false for the previously active slot.

8.7.3 Reset

A O-RU reset is required to take the activated software into operation. See clause 9.5 for details of how an O-RU controller can trigger reset operation.

8.8 Software update scenario

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

The present document does not distinguish between a software upgrade and a downgrade.

4) 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.

5) NETCONF client validates whether the O-RU has data node **build-content-download**.

6) In case the O-RU does not expose **build-content-download**, NETCONF client, knowing target SW version, O-RU product type and manifest.xml content, determines the software package zip file(s) to be downloaded.

7) In case the O-RU exposes data node **build-content-download**, NETCONF client, knowing target SW version, O-RU product type and manifest.xml content, determines the desired build name, build version and software files in the build to be downloaded.

8) NETCONF client using the **software-download** RPC triggers the O-RU to download a file (if several files are required, steps 5-7 need to be performed repeatedly until all files needed by O-RU have been downloaded)

9) O-RU sends RPC response that download was started

10) O-RU finishes downloading the file(s) and reports this by sending the **download-event** notification

11) 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.

NOTE: list of files to be installed cannot be empty.

12) O-RU sends RPC response that installation was started

13) O-RU sets installation slot status to INVALID

14) O-RU installs the software and after successful installation (with checksum control) changes status of the slot to VALID

15) O-RU notifies the notification subscriber that the installation is finished using **install-event** notification

16) NETCONF client requests the O-RU to activate the slot that contains the newly installed software using the **software-activate** rpc

17) O-RU sends RPC response that activation was started

18) For requested slot, O-RU changes **active** to true and at the same time sets **active** to false for previously active slot

19) O-RU notifies the notification subscriber about activation finished using the **activation-event** notification

20) NETCONF client restarts the O-RU forcing it to use the newly installed and activated software. Refer to 9.5 O-RU reset for detailed information.

8.9 Factory reset

An O-RU that support a read only slot can perform this procedure. 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. Factory default software may have limited functionality and O-RU that runs on factory default SW will typically require a software update before it can be fully functional (for example carry traffic). Factory default software shall support all functionality needed to perform a software update.

9 Configuration management

9.1 Baseline configuration

9.1.1 NETCONF operations

The O-RU shall use the ietf-yang-library mode, as specified in RFC 8525 [59] to signal the namespaces of the YANG models supported by the NETCONF Server. When the O-RU signals it supports the YANG module for NETCONF Monitoring, as specified in RFC 6022 [67], an O-RU Controller shall be able to retrieve the models from the O-RU using the <get-schema> operation.

Clause 9 describes NETCONF standard operation (edit-config/get-config/get) as specified in clause 7 of RFC 6241 [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.

The following scenarios are feasible for Configuration Management purposes.

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

As described in clause 6.6, the NETCONF client can use sub-tree based <filter> and XPath based <filter> to recover particular sub-trees from the O-RU. When using XPath filtering, the O-RU returns a node-set which matches the criteria expressed in the XPath. An example of a get operation with XPath based filtering is shown in figure 9.1.1-1.

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
     <source>
       <running/>
     </source>
     <!-- get all critical alarms -->
     <filter xmlns:o-ran-fm="urn:o-ran:fm:1.0"
             type="xpath"
             select="/o-ran-fm:active-alarm-list/o-ran-fm:active-alarms[o-ran-fm:fault-
severity='CRITICAL']"/>
  </get>
</rpc>
```

Figure 9.1.1-1: Example of filtering based on XPath

9.1.2 Retrieve state

O-RU Controller is able to retrieve state which is defined in o-ran-hardware by using NETCONF <get> procedure, as illustrated in Figure 9.1.2-1.

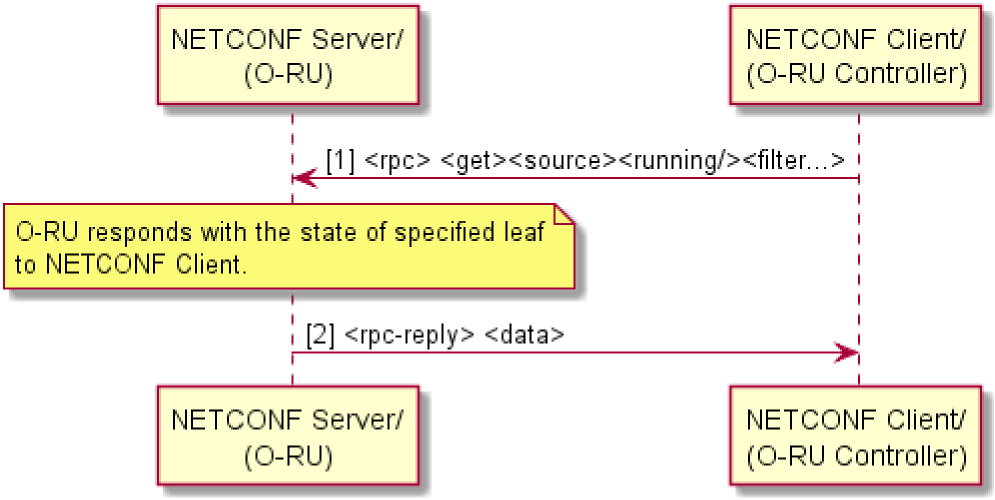


Figure 9.1.2-1: Retrieve Resource State

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.

9.1.3 Modify state

For O-RUs that support the optional hardware-state feature defined in ietf-hardware, the O-RU Controller is able to change state which can be configurable by using NETCONF <edit-config> procedure without reset, as illustrated in Figure 9.1.3-1.

The configurable state is admin-state defined in o-ran-hardware YANG model. In case of a failure, an error shall be returned. Please refer to RFC6241 Appendix A for error codes. The vendor can define the behaviour after the error occurred.

The power-state is read-only state defined in o-ran-hardware.yang. This state is only exposed by the O-RUs supporting ENERGYSAVING feature and is used by O-RU to inform if unit is in energy saving state, not in energy saving state or in transition between energy saving state and non-energy saving state.

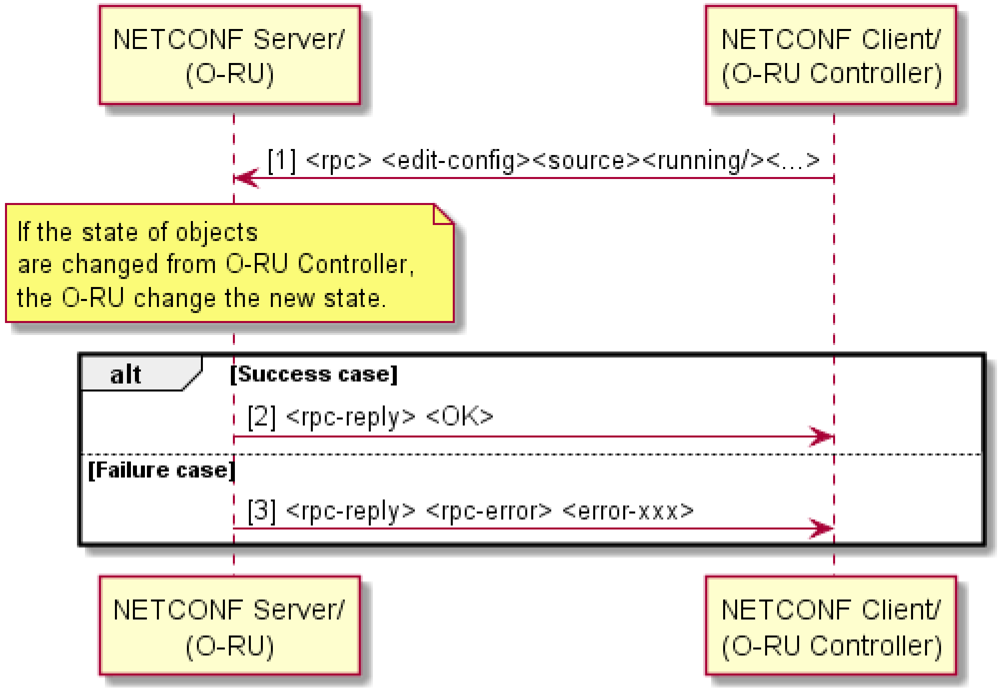


Figure 9.1.3-1: Modify Resource State without reset

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 9.1.3-2.

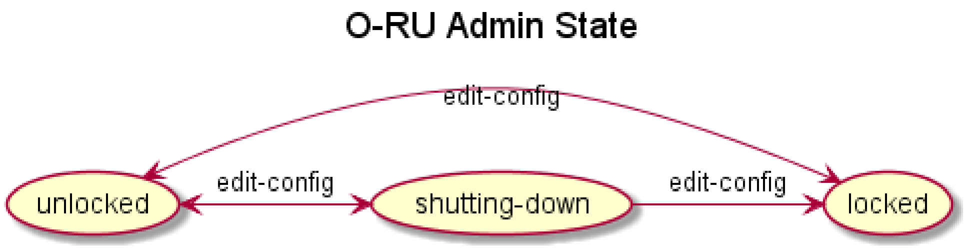


Figure 9.1.3-2: Admin State

- 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 is 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 the figure below. This state can be indirectly controlled by editing the parameters **energy-saving-enabled** and [tr]x-array-carrier::**active**, as illustrated in Figure 9.1.3-3.

Figure 9.1.3-3: Transition diagram for power-state node

- AWAKE: This value of power-state node indicates that the O-RU is operating normally, i.e., not in energy saving mode. AWAKE is the initial value of power-state node after the reset of the O-RU. AWAKE is power-state of O-RU in case energy-saving-enabled is FALSE or when at least one carrier is active.
- SLEEPING: This value of power-state node 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 autonomously stopped by the O-RU if there is no active carrier and value of energy-saving-enabled is TRUE - to reduce energy consumption.
- UNKNOWN**:** This value of power-state node can be exposed by the O-RU e.g., in case the O-RU does not know its power-state value is AWAKE or SLEEPING. This value of power-state node 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**., as illustrated in Figure 9.1.3-4. 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**.

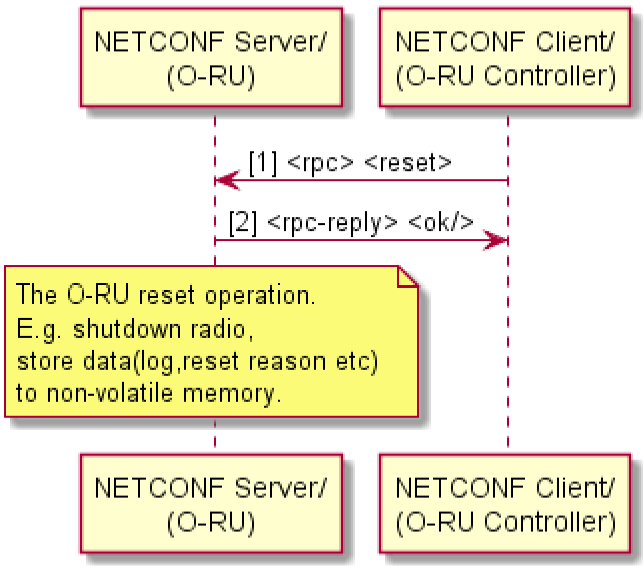


Figure 9.1.3-4: Modify Oper State (reset)

The oper-state transition diagram for O-RU is presented in Figure 9.1.3-5.

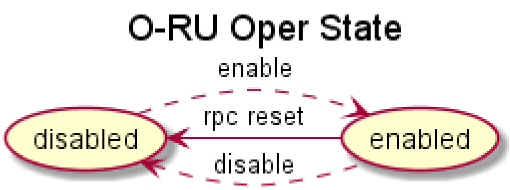


Figure 9.1.3-5: Oper State

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 9.1.3-6.

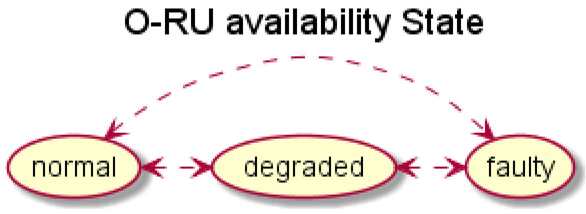


Figure 9.1.3-6: Availability State

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 9.1.3-7.

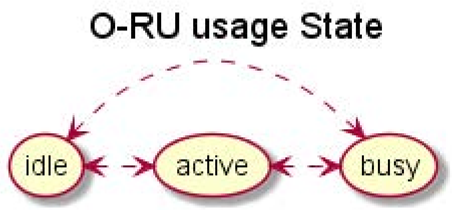


Figure 9.1.3-7: usage State

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.

9.1.4 Retrieve parameters

O-RU Controller is able to retrieve parameters of the YANG module by using NETCONF <get> or <get-config> procedure, as illustrated in Figure 9.1.4-1.

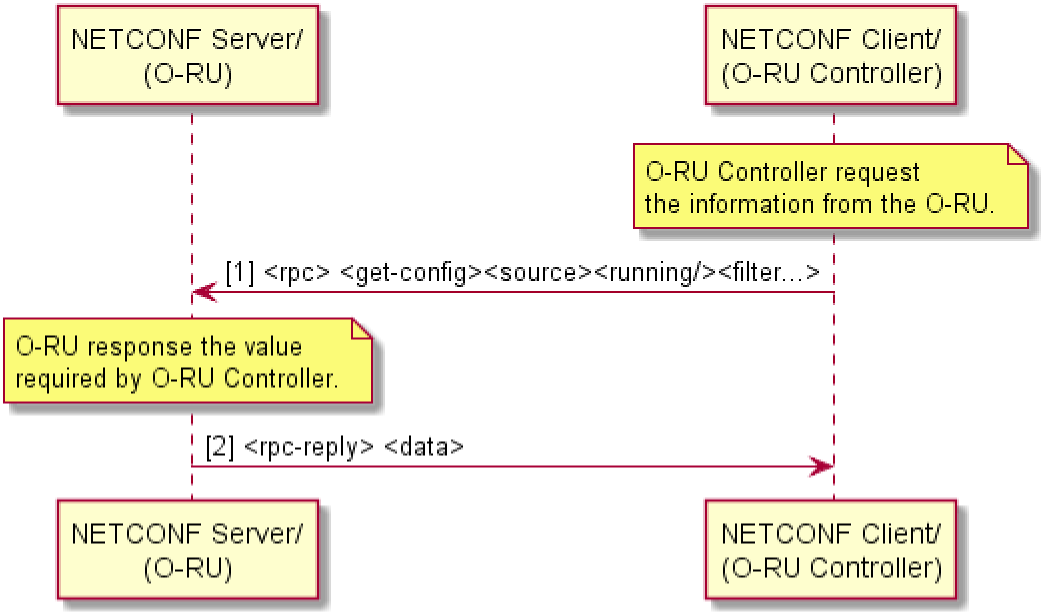


Figure 9.1.4-1: Retrieve Parameters

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.

9.1.5 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 shall respond with a NETCONF error indicating that the requested lock is denied. 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, as illustrated in Figure 9.1.5-1.

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 shall be returned. Please refer to RFC6241 Appendix A for error codes.

The vendor can define the behaviour after error occurred.

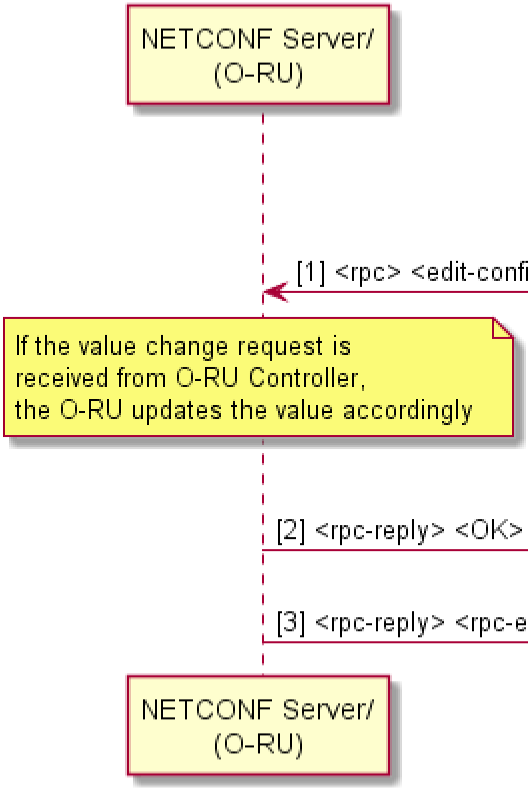


Figure 9.1.5-1: Modify Parameters

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.

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.

9.1.6 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 shall respond with a NETCONF error indicating that the requested lock is denied. 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, as illustrated in Figure 9.1.6-1.

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.

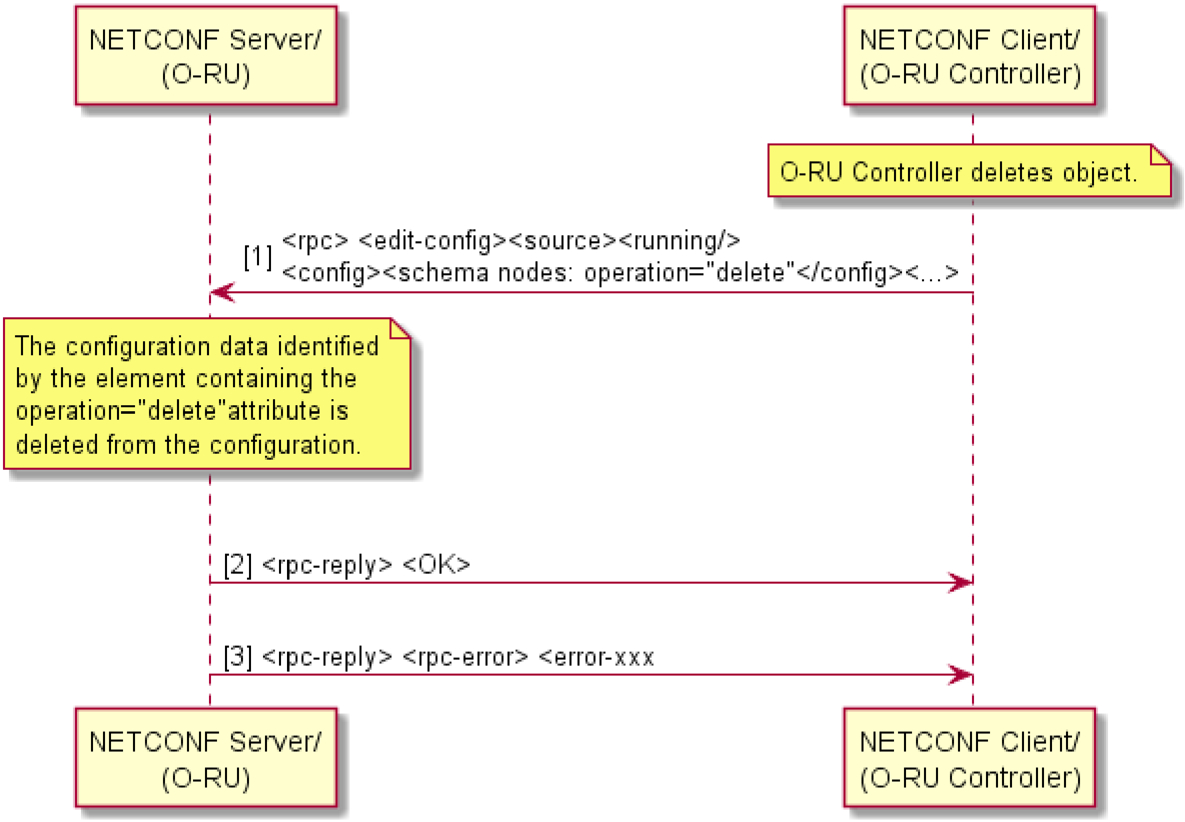


Figure 9.1.6-1: Delete Parameters

Preconditions:

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

9.1.7 Notification framework

9.1.7.1 General

YANG notifications are used to signal event information to notification subscribers. The present document support two types of subscriptions for receiving YANG notifications:

- dynamic, which is supported by all O-RUs, and
- configured, which is supported by O-RUs that support the NON-PERSISTENT-NETCONF feature.

9.1.7.2 Event streams

All O-RUs shall support the default “NETCONF” event stream, as specified in clause 3.2.3 of RFC 5277 [21]. In addition to the default event stream, O-RUs should support the following event streams:

- “fault-management” used to subscribe to YANG notification events defined in o-ran-fm YANG model.
- “measurement-result-stats” used to subscribe to YANG notification events defined in o-ran-performance-management YANG model.
- “supervision-notification” used to subscribe to YANG notification events defined in o-ran-supervision YANG model.

9.1.7.3 Event stream discovery and subscription

An O-RU controller discovers the O-RU's ability to process and send event notifications via capability exchange, as defined in clause 3.1 of RFC5277 [21].

The O-RU shall support event stream discovery as defined in clause 3.2.5 of RFC5277. Prior to initiating a dynamic subscription to an event stream, an O-RU controller can discover the O-RU’s supported event streams. Refer to clause 3.2.5 of RFC5277 for detail.

If the O-RU supports the NON-PERSISTENT-NETCONF feature, the O-RU shall maintain the **streams** container as defined in clause 3.1 of RFC8639 [37]. An O-RU controller supporting RFC 8639 operating with an O-RU that supports the NON-PERSISTENT-NETCONF feature may use the **streams** container to recover the list of available streams supported by the O-RU.

The O-RU supported event streams can be subscribed using dynamic subscriptions and, for those O-RUs that support the NON-PERSISTENT-NETCONF feature, using configured subscriptions. These are transported using NETCONF and VES respectively.

- When no <stream> parameter is provided at creation of subscription, the default “NETCONF” stream is subscribed.
- The "NETCONF" default event stream can be subscribed to by all O-RU controllers. Events defined in the YANG modules to which the O-RU controller does not have “execute” access privileges, i.e., modules without an “X” in table 6.5-1, shall be excluded before the events which are sent to the subscriber.
- For non-default streams, the NACM “execute” privileges defined in clause 6.5 shall be used by the O-RU to determine whether an O-RU controller has privileges to establish a subscription to a particular non-default event-stream.

9.2 Framework for optional feature handling

This clause 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.

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.

Both the NETCONF Server and NETCONF Client shall use the ietf-yang-library mode, as specified in RFC 8525 [59] to signal the namespaces of the models supported by the NETCONF 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/NETCONF Server 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.

9.3 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.

9.4 Notification of updates to configuration datastore

9.4.1 Introduction

This clause 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 as 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.

9.4.2 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 may subscribe to the **netconf-config-change** notification.

An example event notification is shown in Figure 9.4.2-1, where a notification indicates that the configuration of the O-RU's timezone offset has been modified by a second O-RU controller.

```
<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 9.4.2-1: Example of a netconf-config-change notification

9.5 Resetting O-RU

9.5.1 O-RU reset procedure

An O-RU reset may be triggered by an O-RU controller or autonomously triggered by an O-RU, e.g., as a result of supervision failure specified in clause 14.1.1.

9.5.2 O-RU controller triggered O-RU reset operation

The reset procedure can be used by an O-RU controller to trigger the reset of the O-RU. The reset of an O-RU triggers the O-RU to perform a re-start and follow the procedures defined in clause 6.

9.5.3 O-RU controller triggered O-RU reset procedure

O-RU Controller is able to trigger the reset of an O-RU by using the **reset** rpc, as illustrated in Figure 9.5.3-1.

Pre-condition:

- M-Plane connection is established and running between O-RU and O-RU controller

Post-condition:

- O-RU performs reset followed by start-up sequence described in clause 6

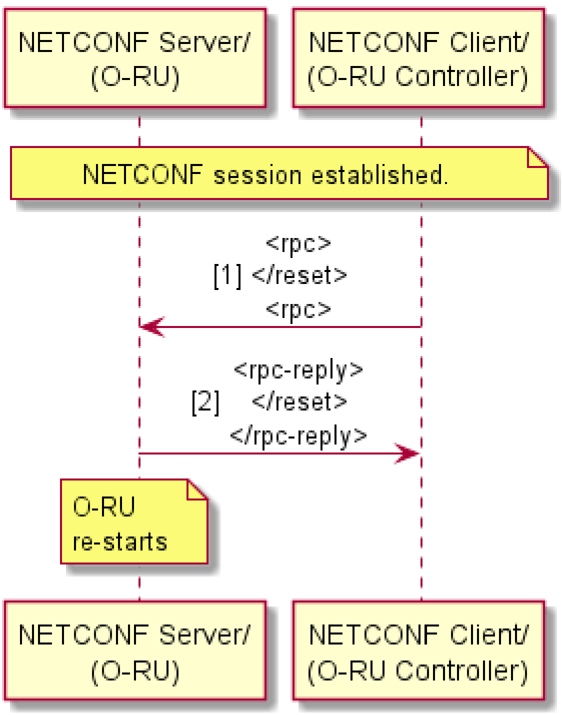


Figure 9.5.3-1: O-RU Reset Procedure

10 Performance management

10.1 General

This clause 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.

10.2 Measurement activation and de-activation

The measurement activation at the start-up installation is also allowed as described in clause 6.

Pre-condition:

M-Plane is operational.

Post-condition:

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

This clause 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**, **epe-measurement-object**s and **symbol-rssi-measurement-ob**jects.
- **measurement-interval**: measurement interval for the **measurement-objects** 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-uni**t may be configurable Identifier **object-unit-id** means e.g., physical port number when object unit set to physical port number.
- **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, as illustrated in Figure 10.2-1. 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 details see the following illustration.

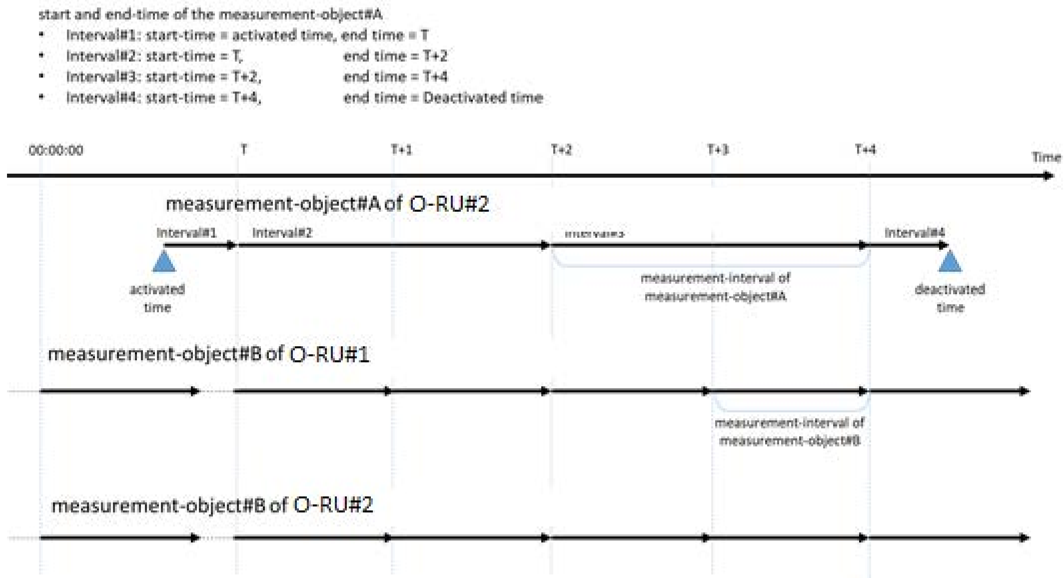


Figure 10.2-1: synchronization of measurement-interval.

The modification of the configurable parameters (except **active**) for the measurement and removal of measurement objects (e.g., **transceiver-measurement-objects**, **rx-window-measurement-objects**, etc.) shall be allowed only while **active** for the corresponding **measurement-object** has value FALSE.

O-RU shall support those measurements described as mandatory in clause 9.1 of the O-RAN CUS plane specification [2].

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.

10.3 Collection and reporting of measurement result

10.3.0 Introduction

This clause 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 to transfer 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.

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.

10.3.1 NETCONF process for dynamic subscriptions

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 <**create-subscription**> 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 Figure 10.3.1-1.

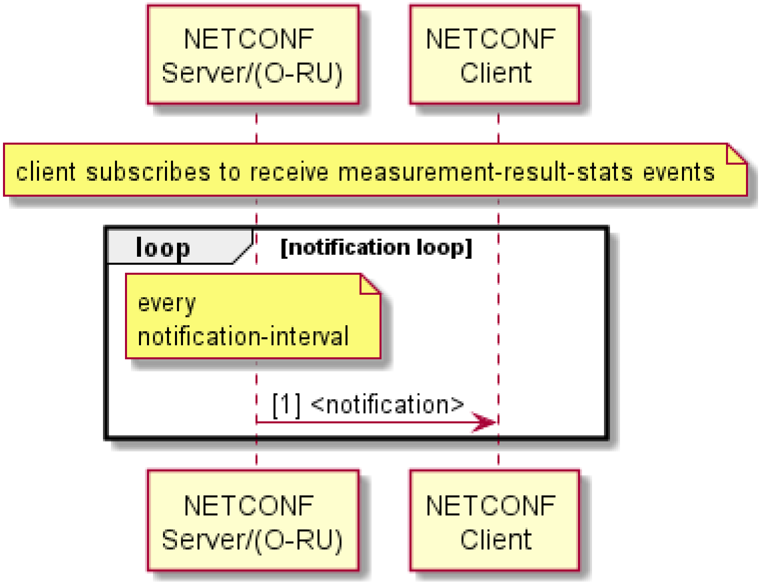


Figure 10.3.1-1: NETCONF process of Measurement Result Collection

NOTE 1: A NETCONF client may use the **create-subscription** for the single event-stream "**measurement-result-stats**", or alternatively subscribe the default event-stream using additional filter criteria to identify those mesaurements of interest to the client. In order to subscribe multiple notifications, the appropriate **create-subscription** message is required. Please refer to clause 11.3 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 or configuring the <stopTime> parameter when the subscription was created. If NETCONF session is terminated by <**kill-session**>, the subscribed notification is terminated as well.

This procedure is described in Figure 10.3.1-2.

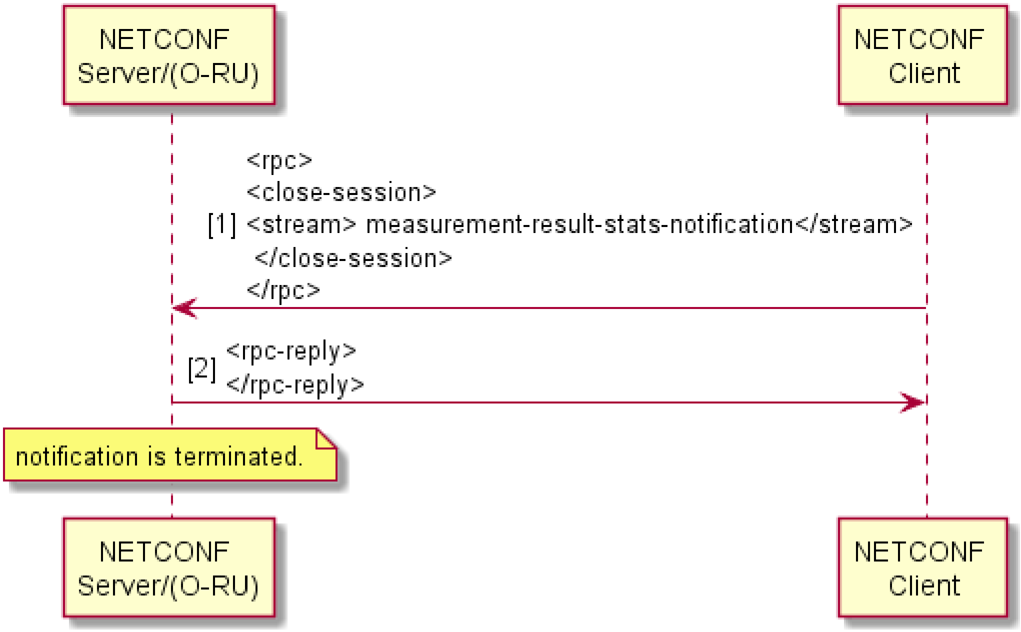


Figure 10.3.1-2: NETCONF process of Measurement Result Collection to end

When **notification-interval** is larger than **measurement-interval**, O-RUs implementing v07.00 or later of this document may use a single notification that contains multiple stats included in the list such as **multiple-transceiver-measurement-result**, **multiple-rx-window-measurement-result**, **multiple-tx-measurement-result**, **multiple-epe-measurement-result** or **multiple-symbol-rssi-measurement-result** which have consecutive periods indicating **start-time** and **end-time** for the measurement.

NOTE 2: When multiple measurements stats corresponding to the same measurement-group are available in an O-RU that implements an earlier version of this document, the O-RU is only able to include a single statistic in the notification. In such circumstances, the O-RU should report the latest available statistic in the notification.

NOTE 3: An O-RU Controller that implements an earlier version of this document can ensure that the configuration of **notification-interval** and **measurement-interval** results in only a single measurement statistic corresponding to a specific measurement-group being reported by the O-RU in any one notification. In other cases, the O-RU Controller can use the rules defined in clause 4.3 to ignore the newly introduced schema-nodes for reporting multiple measurements.

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.

10.3.2 File management process

NETCONF client needs to configure a parameter of performance measurement YANG module '**enable-file-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-file-upload-path**, **credentials** information of the file server and **enable-random-file-upload** as configurable parameters.

Following types of authentications shall be supported for **performance file upload**:

a) Password for RU authentication and list of public keys (see clause 5.4 in the present document) for sFTP server authentication

Following types of authentications may be supported for **performance file upload**:

b) X.509 Certificate for FTPES client (O-RU) and FTPES server

c) Certificate for both O-RU and sFTP server authentication

When the parameter **enable-file-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/ or o-ran/pm/. Every **file-upload-interval**, O-RU pushes the latest file to upload to the **remote-file-upload-path** of the configured SFTP/FTPES servers if **enable-file-upload** is set to TRUE. Otherwise, the performance measurement file is not created and uploaded. The O-RU shall use the URI scheme of the **remote-file-path** leaf to determine whether to use sFTP or FTPES for the file upload. 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. An O-RU Controller shall only trigger file upload using FTPES if it is using NETCONF/TLS to configure the O-RU

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**.

If the parameter **enable-random-file-upload** is set to TRUE, the O-RU shall randomize the timing to upload SFTP or FTPES 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
- :
- 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 or 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 clause 12.

10.3.3 Configured subscription process

This optional process requires the O-RU to support configured subscriptions, as described in clause 18. The structure of the process follows the NETCONF process described in clause 10.3.1. 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 shall send to a subscription state change notification indicating that the subscription has ended to the Event-Collector.

NOTE: Unlike the NETCONF process described in clause 10.3.1, the subscription to the subscribed notifications is not terminated when the NETCONF session used to establish the subscription is terminated.

11 Fault management

11.1 Introduction

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 clause 18, 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 the **active-alarm-list**. Alarms with **fault-severity** set to 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**.

Optionally, the NETCONF Server is also responsible for managing the **historical-alarm-list**. Alarms with **fault-severity** set to WARNING are excluded from this list. When an alarm is cleared, it is added to this list with **is-cleared** set to "true".

The NETCONF Client can read " active-alarm-list" by **get** RPC operation, as illustrated in Figure 11.1-1.

Figure 11.1-1: Read Active Alarms

11.2 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, as illustrated in Figure 11.2-1.

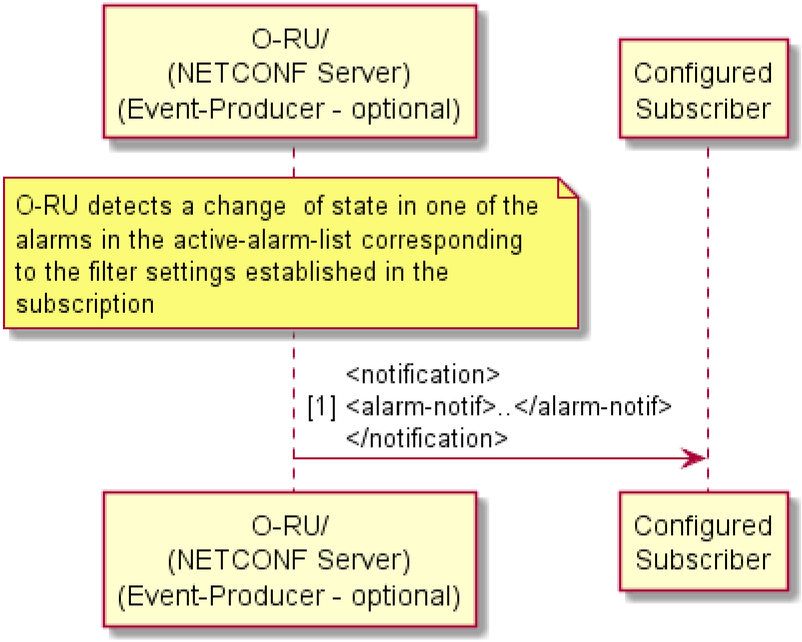


Figure 11.2-1: Alarm Notification

11.3 Manage alarms request to NETCONF clients

The NETCONF Client can "subscribe" to Fault Management Element by sending **create-subscription**, as specified in clause 2.1 of 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>
      </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 in Figure 11.3-1. 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.

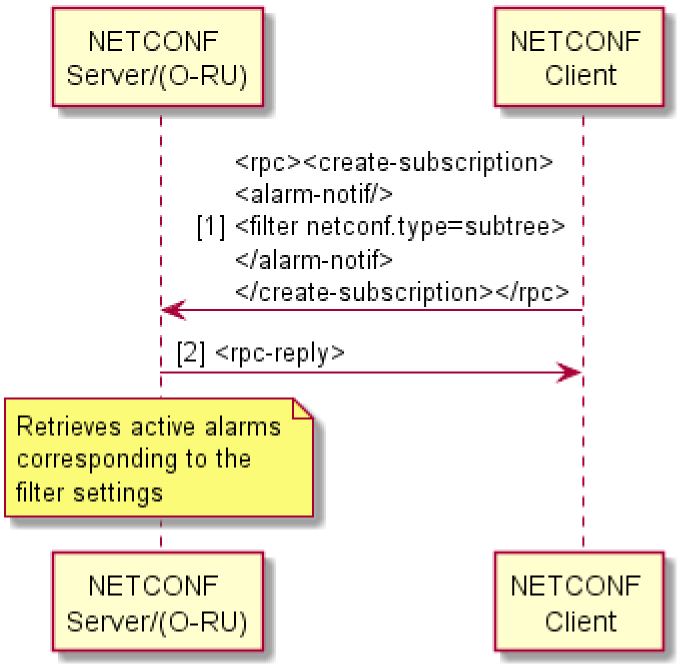


Figure 11.3-1: Manage Alarms Subscription Request

To terminate the subscription, the NETCONF client shall send a <close-session> operation from the subscription's session, as illustrated in Figure 11.3-2.

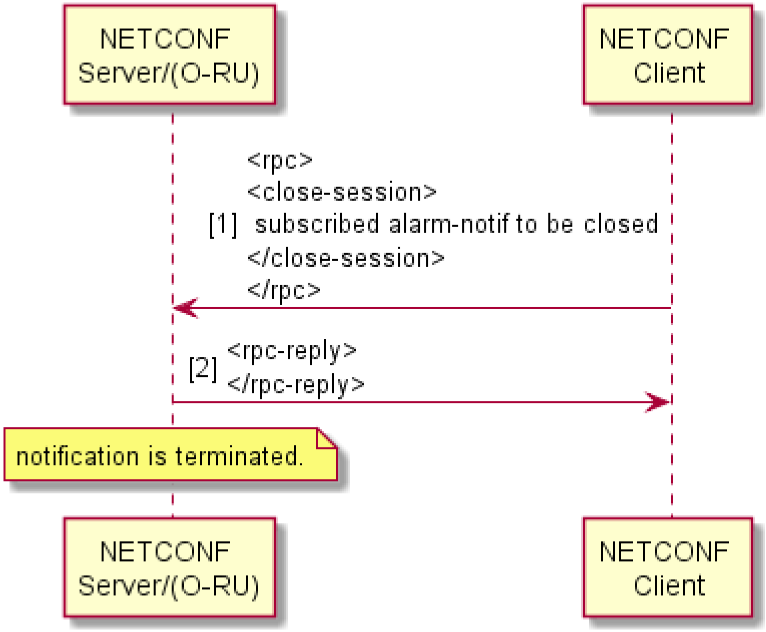


Figure 11.3-2: Terminating an Alarm Subscription

11.4 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 derived from 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 derived from element name.

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

Value of fault-source may identify the O-RU external resource; 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 numbers may 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 do not function 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 do not function correctly due to the fault are reported via "affected-objects".

11.5 Manage alarms request to event-collector

This optional capability requires the O-RU to support configured subscriptions, as described in Clause 18. The structure of the process follows the process described in clause 11.3. 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 shall send to a subscription state change notification indicating that the subscription has ended to the Event-Collector.

11.6 Historical Alarms List

The O-RU may optionally support the **historical-alarm-list** container, enabling alarm events with **is-cleared** status is "true" to be captured. The procedures for aging out entries in the **historical-alarms** list based on time and/or memory is left to vendor specific implementation of the O-RU.

NOTE: The persistence of **historical-alarms** list entries on O-RU upon reset is left to vendor specific implementation

The NETCONF Client can read **historical-alarm-list** by **get** RPC operation, as illustrated in Figure 11.6-1.

Figure 11.6-1: Read Historical Alarms

In case of multiple NETCONF clients and/or Event-Collectors, the O-RU shall report the same notification to all subscribed NETCONF clients/Event-Collectors.

12 File management

12.1 Introduction

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

- upload (see clause 12.3)

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

- retrieve file list (see clause 12.4)

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

- download (see clause 12.5)

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

NOTE 1: file-download has different purpose with software-download specified in clause 8.5. For example, file-download can be used for Beamforming configuration in clause 15.4.

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

a) Password for RU authentication and list of public keys (see clause 5.4 in the present document) for sFTP server authentication

b) X.509 Certificate for TLS between O-RU and FTPES file server. In addition to the X.509 certificate, the O-RU shall use any configured application password, **appl-password**, to authenticate to the FTPES server for the associated username defined in the **remote-file-path**.

NOTE 2: The **appl-password** is optional for FTPES server authentication to support scenarios where the username embedded in the **remote-file-path** corresponds to an anonymous account as defined in RFC 1635 [i.5].

Following types of authentications may be supported for **file management**:

c) Certificate for both O-RU and sFTP server authentication

Following other clauses are related with File Management.

- clause 10.3.2: File Management process (can be used for on demand file upload purpose, since clause 10.3.2 covers periodic file upload)
- clause 14.2: Log Management
- clause 14.3: Trace
- clause 15.4.3: Beamforming configuration update

NOTE 3: The file management functions involve the O-RU controller subscribing to receive particular YANG notifications from the O-RU. All O-RUs support the NETCONF Create-Subscription method, enabling those notifications to be transported using NETCONF notifications. In addition, those O-RUs that support the optional NON-PERSISTENT-MPLANE feature, the O-RU Controller can create a configured subscription from the O-RU, enabling those notifications to be transported over HTTPS to an Event-Collector as described in clause 18.

12.2 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 clause. 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. Prior to version 10.0, the following standardized folders were defined:

- O-RAN/log/
- O-RAN/PM/
- O-RAN/transceiver/

And for those O-RU's supporting beamforming

- O-RAN/beamforming/

The folders listed above are inconsistent with the folders used in figures elsewhere in the present document. Because of such inconsistencies, the use of the folders listed above is not preferred. These folders may be subject to removal in a future version of the present document.

To avoid interoperability issues between case-sensitive and non-case sensitive file systems, the following additional standardized folders are defined and are preferred in the present version of this document:

- o-ran/log/
- o-ran/pm/
- o-ran/transceiver/

And for those O-RUs supporting beamforming

- o-ran/beamforming/

The O-RU may additionally support vendor defined folders which are out of scope of the present document.

12.3 File management operation: upload

This clause describes file upload method from O-RU to the fileserver. sFTP or FTPES 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 FTP connection between O-RU to the fileserver. If the O-RU has a limitation with regard to the ability to upload files simultaneously, it is allowed that O-RU reports failure notification for the simultaneous upload request which exceeds its 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. Figure 12.3-1 shows the file upload sequence diagram.

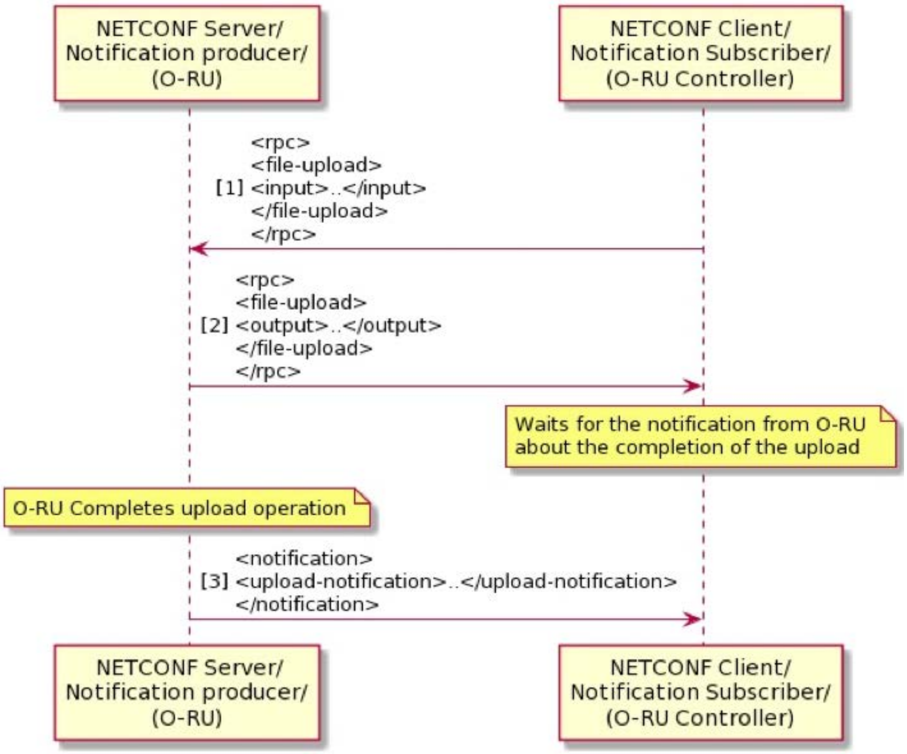


Figure 12.3-1: File Upload Sequence

12.4 File management operation: retrieve file list

This clause 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

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 12.4-1 shows the retrieve file list sequence diagram.

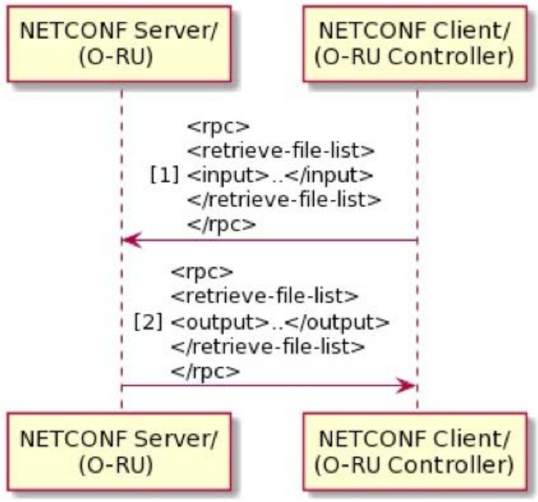


Figure 12.4-1: Retrieve File List Sequence

12.5 File management operation: download

This clause describes the file download method from O-RU Controller to O-RU. sFTP or FTPES 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 FTP 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. Figure 12.5-1 below shows the file download sequence diagram.

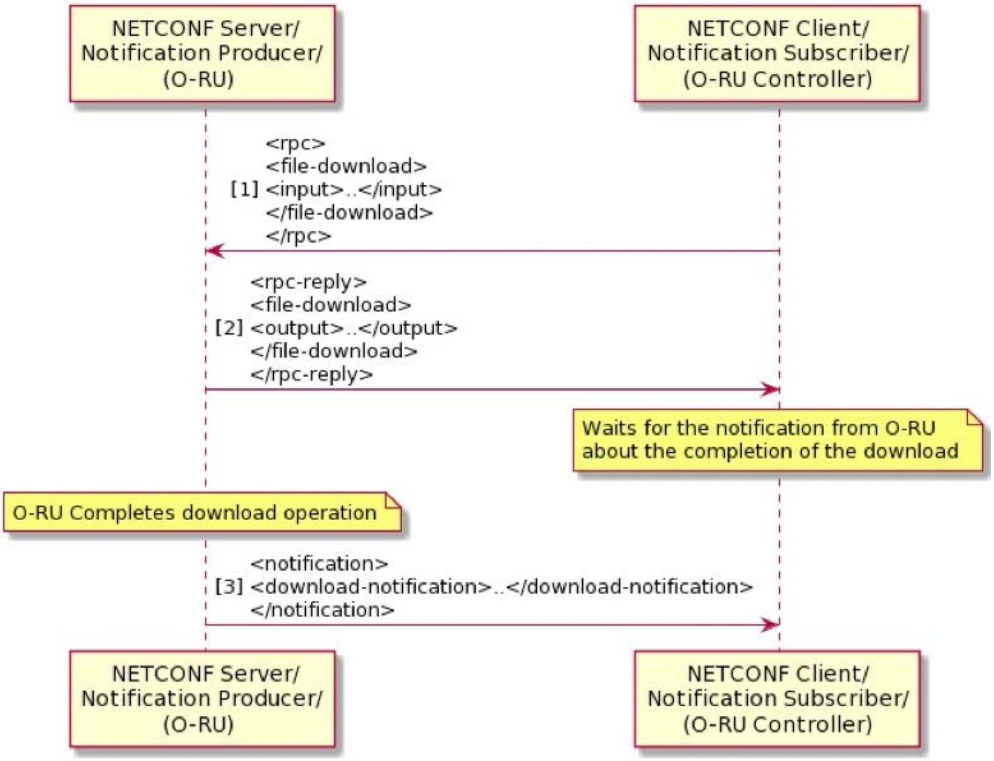


Figure 12.5-1: File Download Sequence

13 Synchronization aspects

13.1 Introduction

This clause describes the YANG model (**o-ran-sync.yang**) containers that are used for 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. Additional information regarding the interactions between M-Plane and S-Plane are described in the current specification's clause 15.3.3 and in the CUS-Plane specification [2], clause 11.4.

The **o-ran-sync.yang** YANG model defines configuration, status and notification containers which are used to enable O-RU controller to recover the state of the O-RU's S-Plane. Before going into the details of each of these an overview of these objects is provided here.

The YANG model defines several configuration containers. These containers comprise read/write parameters that can be read or set by the O-RU controller. The O-RU controller can read them (to determine O-RU configuration) or set them (to change the O-RU's configuration). These are:

- PTP Configuration – **ptp-config:** O-RU controller can set or read PTP parameters like PTP Full/Partial timing support profiles, accepted clock classes, and set network address among other parameters

- PLFS/SyncE Configuration – **synce-config:** This container provides SyncE information such as the acceptable list of SSMs.

- GNSS (Global Network Satellite System) Configuration - **gnss-config:** This container comprises leaves which define whether GNNS can be used, and if so which satellite constellation to use.

- Sync Capability – **sync-capability:** This is a read-only container indicates to the O-RU controller the time accuracy of the T-TSC or T-BC that is integrated into the O-RU.

The O-RU should store the above synchronization configuration in reset persistent memory.

During the start -up installation process (see clause 6.1 in this document) the O-RU's startup configuration settings may allow the O-RU’s S-Plane to become operational without any configuration by the O-RU controller. The O-RU uses its configuration settings to execute the PTP and/or PLFS/SyncE protocols, or GNSS settings to complete the synchronization process. During the start-up installation flow the startup settings may be changed. This may be necessary if the default synchronization settings are incompatible with the network implementation. If the startup configuration requires changes, the O-RU controller first retrieves the synchronization configuration and status from O-RU, and then the O-RU controller modifies the configuration. The O-RU then executes the synchronization process using the updated settings. Having these settings stored in reset persistent memory typically simplifies subsequent startup flows by enabling O-RU to start up without reconfiguration.

Once the O-RU is operational, the O-RU controller can subscribe to several synchronization notifications which report to the O-RU controller the status of the O-RU. The following notifications are defined in the YANG model.

- **synchronization-state-change** – conveys the Sync Status (**sync-state)** container information
- **ptp-state-change –** conveys the PTP Status (**ptp-status)** container
- **synce-state-change –** If SyncE or another PLFS is used, this conveys the PLFS/SyncE (**synce-status)** container
- **gnss-state-change** – If a O-RU supports GNSS capability, this conveys the GNSS Status (**gnss-status)** container.

The above notifications when triggered send their respective status container to the subscribed O-DU/O-RU controller.

- Sync Status – **sync-state:** Overall synchronization status of O-RU.
- PTP Status – **ptp-status:** Status of PTP synchronization process
- PLFS/SyncE Status – **synce-status:** Status of PLFS/SyncE frequency synchronization process
- GNSS Status **– gnss-status:** Synchronization status of the GNSS clock.

During normal operation, the O-DU should monitor the operational state of the O-RU’s S-Plane. This is accomplished by subscribing to receive the synchronization-state-change notification. If the O-DU is notified that the O-RU’s state has changed to UNLOCKED, the O-DU shall stop sending data to the O-RU and shall ignore any data from that O-RU. Sub-clause 15.3.3 describes in more detail the interactions between S-Plane and M-Plane when synchronization is lost and when synchronization is recovered.

NOTE: Except for possible configuration during the start-up installation process, the synchronization start-up and reference clock selection are accomplished via S-Plane protocols independent of M-Plane.

The remainder of this clause describes in more detail the main objects within the configuration, status, and notification containers. However, this clause does not describe all of the leafs/objects within the containers. Some items for example **time-error, frequency-error, supported-reference-type** are not mentioned in this clause, but are defined and described in o-ran-sync.yang.

13.2 Sync status object

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

For an O-DU that is communicating with an O-RU, the **synchronization-state-change** notification is the primary mechanism by which the O-DU knows whether the O-RU is synchronized and is ready to transmit/receive when fully configured. This notification is the primary mechanism to transfer status information required to implement the flows described in section 12.3.3.

The state diagram of O-RU synchronization is shown in Figure 13.2-1, and is indicated by the following allowed values:

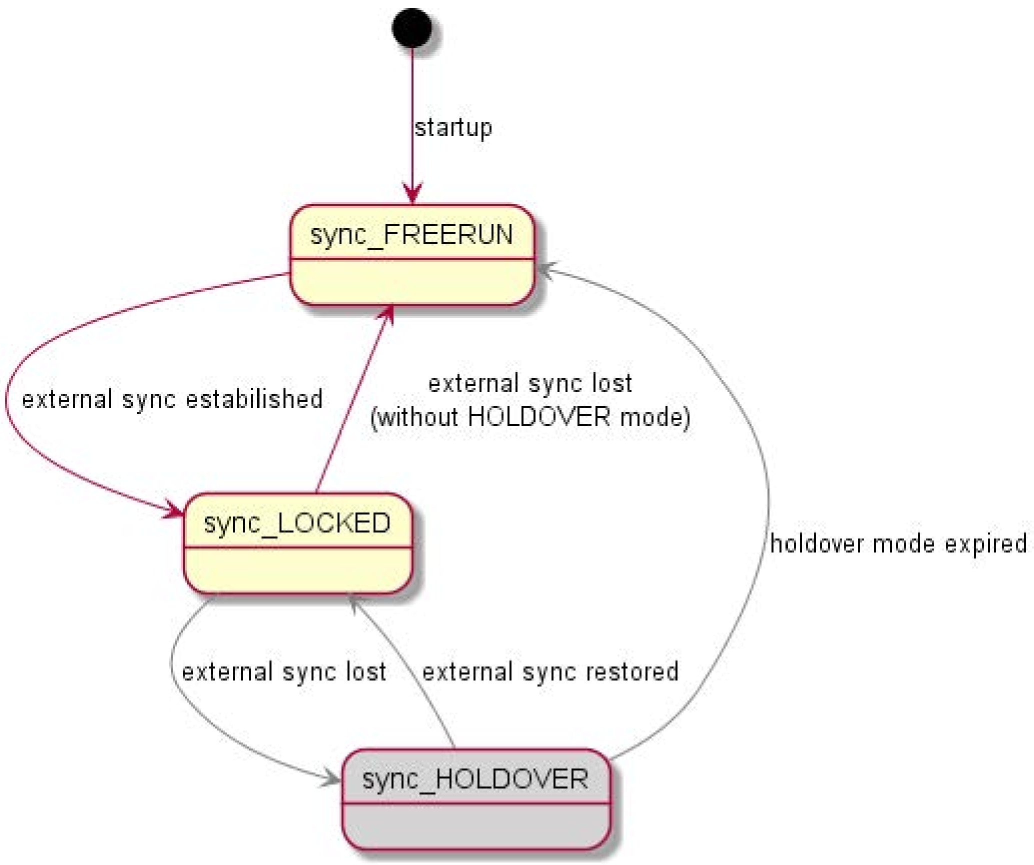
- **LOCKED:** O-RU is in the locked mode, as defined in ITU-T G.810.
- **HOLDOVER:** O-RU clock is in holdover mode.

- **FREERUN:** O-RU clock isn't locked to an input reference and is not in the holdover mode.

Figure 13.2-1 illustrated the state transitions.

The **sync-state** container allows the O-RU to list via an array of the synchronization sources, called supported-reference-types, which it can support. The allowed values are:

- GNSS
- PTP
- SYNCE



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

Figure 13.2-1: Allowed sync state transitions

13.3 Sync capability object

The module's synchronization capability is provided via the sync-t-tsc object. This object indicates the accuracy of the O-RU's derived Telecom Subordinate Clock (T-TSC). For details on the actual capability levels, see clause 11.3 of the O-RAN WG4 CUS plane specification [2]. There are two enumerations possible:

- CLASS_B: Regular accuracy (previously referred to class B) for synchronization is supported by the device as per IEEE802.1CM clause 6.4.1, Case 1.1.
- ENHANCED: Enhanced accuracy for synchronization is supported by the device as per IEEE802.1CM clause 6.4.1, Case 1.2.

Another leaf in this container is a Boolean called **boundary-clock-supported**, which indicates whether the O-RU supports T-BC profiles defined in ITU-T 8275.1 [22].

13.4 PTP configuration

13.4.1 Introduction

The **ptp-config** container contains information about the O-RU's configuration of Precision Time Protocol. The below describes the definitions of the items contained within ptp-config.

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

- 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.

default: 24.

accepted-clock-classes: Contains the list of PTP acceptable Clock Classes, sorted in the descending order.

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

d) 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;

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

default: 7, 6

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

- 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-RU. The error is defined in units of nanoseconds in the range ±10 000 ns. This is a single global value that is common for all the O-RU's ports. As specified in ITU-T G.810 [23] and IEEE1588 [24] and [50], the sign of the parameter shall be 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

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

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.

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

13.4.2 G.8275.1 specific parameters

This container within **ptp-config**, called **g-8275-1-config,** is used when **ptp-profile** is set to G_8275_1. When enabled it instantiates **multicast-mac -address.** The parameter defines the destination MAC address, used by the O-RU in the egress PTP messages. This is a common configuration parameter for all G.8275.1 compliant ports of the O-RU.

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)

default: FORWARDABLE.

13.4.3 G.8275.2 specific parameters

This container, **g-8275-2-config**, within **ptp-config** contains G.8275.2 specific parameters. It is used with **ptp-profile** is set to G_8275_2. The full list of fields within this container and their meaning are listed directly in the YANG model.

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: This 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)

13.5 PTP status

The PTP Status container is used to collect operational status information of the PTP 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 clause 13.2. 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 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 clock is synchronizing to the reference, recovered from PTP flow.
- **UNLOCKED:** The integrated 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: This parameter contains several parameters describing the characteristics of PTP sources of the clock, controlled by the O-RU. Descriptions for these can be found in o-ran-sync.yang.

state: This parameter indicates status of the PTP port:

- **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**, **clock-class**, 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 additional information on the contents of the ptp-status container.

13.6 PLFS/SyncE configuration

The CUS-Plane Specification allows for different methods of PLFS (Physical Layer Frequency Synchronization), however only the use of SyncE is defined in the present document. The **synce-config** container defines the configuration of SyncE. The following list includes the related parameters of this container.

acceptance-list-of-ssm: The parameter contains the list of SyncE acceptable Synchronization Status Messages (SSM).

NOTE: The extended SSM TLV in not supported in the present document, and YANG definitions.

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.

13.7 PLFS/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 clause 13.2. The following list summarizes 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 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** YANG Model for the full details.

13.8 GNSS configuration

The **gnss-config** container defines the configuration of Global Navigation Satellite System (GNSS). The following list summarizes the related parameters of this container.

enable: This parameter defines if GNSS receiver shall be enabled or not. Allowed values: true/false;

default: false.

satellite-constellation-list: This parameter defines list of constellations to be used to acquire synchronization.

Allowed values:

- GPS
- GLONASS
- GALILEO
- BEIDOU

polarity: This parameter defines pulse polarity

Allowed values:

- POSITIVE
- NEGATIVE

default: POSITIVE.

cable-delay: This parameter is used to compensate cable delay. Allowed values: 0 ~ 1000

default: 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: false.

13.9 GNSS status

An O-RU supporting GNSS capability uses the **gnss-status** 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 clause 13.2. 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

14 Operations use cases

14.1 Supervision failure handling and supervision termination handling

14.1.1 Supervision failure handling

This clause clarifies Supervision Failure Handling and Supervision Termination Handling.

When Supervision Failure is detected by O-RU, the O-RU immediately disables operation of the watchdog timers for the corresponding NETCONF session. O-RU assumes NETCONF session related to failed supervision is no longer valid. O-RU terminates this invalid NETCONF session by closing underlying SSH or TLS connection. Then O-RU starts performing the call home procedure towards the NETCONF client, using the **re-call-home-no-ssh-timer** to repeat the call home attempts. This activity is repeated by the O-RU until, either:

- New NETCONF session is established by the original NETCONF client, or
- The original NETCONF client is no longer a "call home O-RU Controller" as defined in clause 6.3, e.g., when reperforming DHCP configuration, the O-RU Controller identity corresponding to the NETCONF client is no longer signalled by the DHCP server, and/or the NETCONF client was previously configured using the **configured-client-info** container and this configuration has been deleted.

Case #1: After entering Supervision Failure handling, the O-RU is still having at least one running and valid NETCONF session with a NETCONF client that has subscribed to receive the **supervision-notification** and where the **per-odu-monitoring** container is not present in the O-RU’s configuration.

The O-RU remains operational and performs periodical Call Home towards call home O-RU controllers as described in clause 6.3.

Case #2: After entering Supervision Failure handling, the O-RU does not have running NETCONF session with any NETCONF client that has subscribed to receive the **supervision-notification**.
The O-RU ceases all radio transmission and performs autonomous reset.

When an O-RU indicates it supports the **SHARED-ORU-MULTI-ODU** feature, an O-RU Controller can enable supervision operation on a per O-DU identity basis by configuring one or more **odu-id** parameters in the **per-odu-monitoring** container in o-ran-supervision YANG model. When an **odu-id** has been configured in the **per-odu-monitoring** container, the O-RU shall enable operation of watchdog supervision timers (supervision timer and notification timer) on a per **odu-id** basis. When enabled, an O-RU controller that has subscribed to the **supervision-notification** is expected to configure one of the **odu-id** values in the **supervision-watchdog-reset** rpc. Operation of watchdog supervision timers on a per **o-du** basis does not obviate the O-RU from performing autonomous reset according to the scenario described in case #2 above.

NOTE: The format of the **odu-id** string is not defined and not interpreted by the shared O-RU.

Case #3: If the supervision timer associated with an **odu-id** expires, the O-RU shall set the **state** leaf of the following array carriers to **DISABLED** and the **active** leaf of the same carriers to **INACTIVE:**

- Any **tx-array-carriers** list entry including a list of odu-ids containing only expired **odu-id**.
- Any **rx-array-carriers** list entry including a list of odu-ids containing only expired **odu-id.**

O-RU shall raise an alarm. The specific alarm #35 "Lost O-DU ID based Supervision " is described in Annex A. The O-RU sends **tx-array-carriers-state-change** and **rx-array-carriers-state-change** notifications to any notification subscribers. O-RU shall not perform an autonomous reset unless the requirements described in case #2 are met.

14.1.2 Supervision termination handling

If NETCONF session used for the supervision subscription is terminated by NETCONF client, the O-RU disables operation of the watchdog timers for terminated NETCONF session and starts performing Call Home procedure towards call home O-RU Controllers, following specification in clause 6.3.

Before terminating its NETCONF session, a NETCONF client that has subscribed to receive supervision notification, should at least de-activate all carriers previously configured by this NETCONF client. Optionally, such a NETCONF client can also remove (full or partial) configuration applied by this NETCONF client to the O-RU.

If the **per-odu-monitoring** container is not present in the O-RU’s configuration, in case when entering Supervision Termination handling, if the O-RU does not have running NETCONF session with any NETCONF client that has subscribed to receive the **supervision-notification**, the O-RU ceases all radio transmission.

If the **per-odu-monitoring** container is present in the O-RU’s configuration, in case when entering Supervision Termination handling, the O-RU deactivates the array carriers associated with the **odu-id** value in its **supervision-watchdog-reset** rpc as specified in clause 14.1.1.

14.2 Log management

14.2.1 Introduction

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.

14.2.2 Troubleshooting log management

By requesting trouble shooting log files an O-RU controller is able to obtain collected log data files that can be used for troubleshooting purposes.

The O-RU can provide 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) - assuming no additional upload restrictions from connectivity bandwidth or file server implementation. It is also recommended to provide more useful files first.

NOTE: The detailed O-RU controller behaviour is not specified. An O-RU controller can 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 collects log information. The RPC <start-troubleshooting-logs> triggers the O-RU to start generating troubleshooting log files containing troubleshooting logs, as illustrated in Figure 14.2.2-1. Completed generation of files is indicated by NETCONF server to NETCONF client in form of a notification.

The notification <troubleshooting-log-generated> is signalled to the notification subscriber after the O-RU has finished generation of all troubleshooting log files, indicating to the subscriber that the troubleshooting log files are ready to be uploaded.

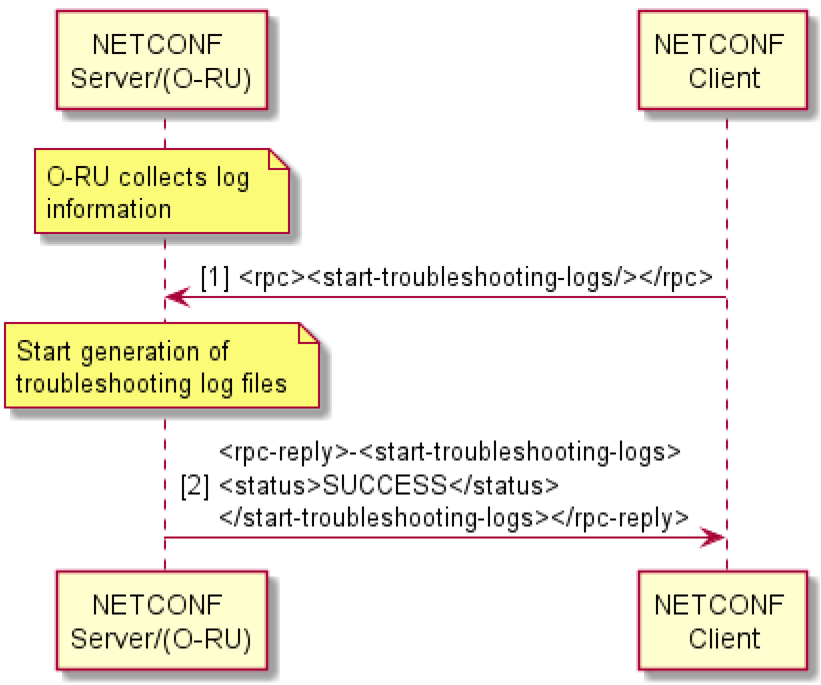


Figure 14.2.2-1: Start Generation of Troubleshooting Log Files

The O-RU shall include file names for troubleshooting log files in the <troubleshooting-log-generated> notification, with illustrative notification contents shown in Figure 14.2.2-2.

After sending the notification, the O-RU shall stop generating further troubleshooting log files unless a new RPC <start-troubleshooting-logs> is triggered. The file transfer for the generated troubleshooting log files shall be handled by the file management defined in Clause 12.

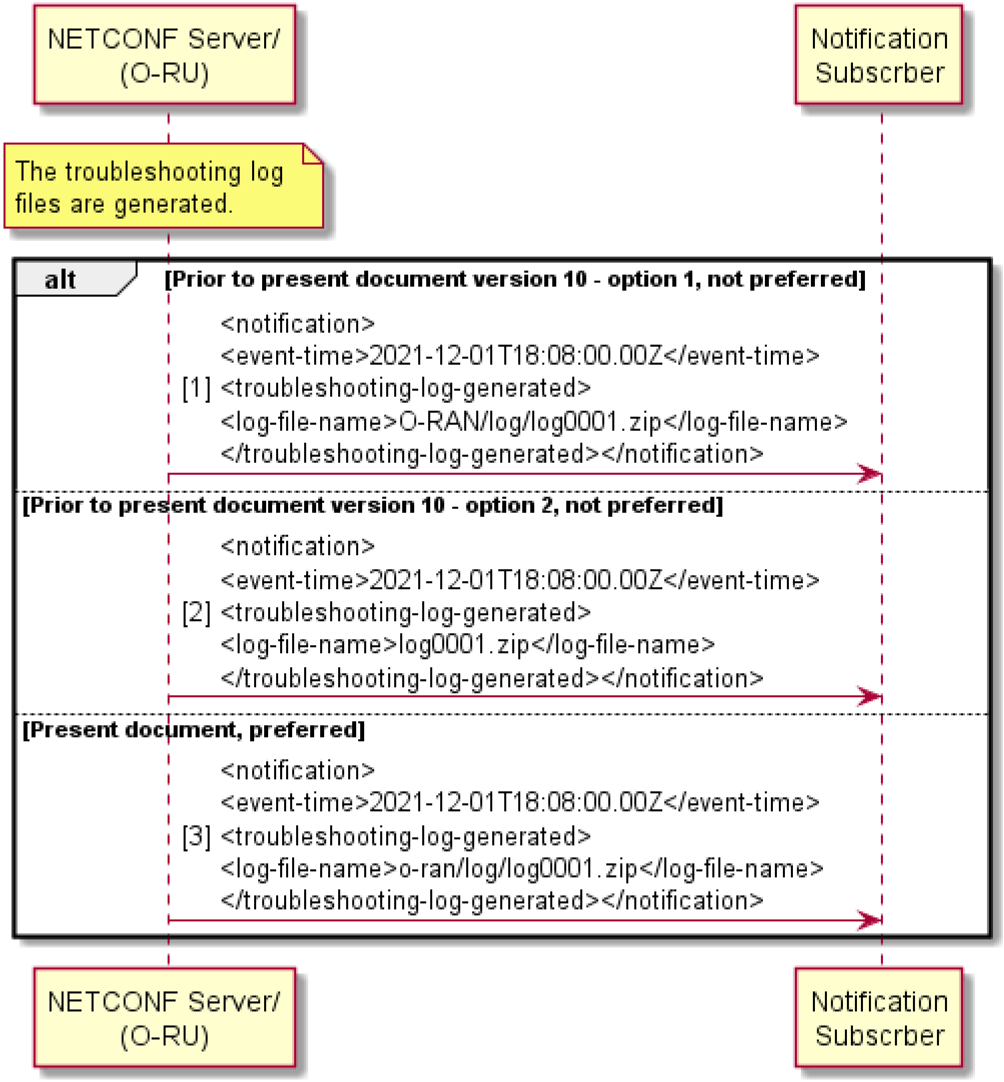


Figure 14.2.2-2: Notification Troubleshooting Log Generated

NOTE: Prior to version 10 of the present document, this document contained inconsistent definition of the **log-file-name** string between Figure 14.2.2-2 and associated YANG models. In present document previously permitted content of node **log-file-name** is considered as not preferred. Not preferred file formats may be subject to removal in a future version of the present document.

NETCONF client cancels generation of troubleshooting logs files using RPC <stop-troubleshooting-logs>. The NETCONF client is no longer interested in troubleshooting log files and O-RU does not need to send <troubleshooting-log-generated> notification, as illustrated in Figure 14.2.2-3.

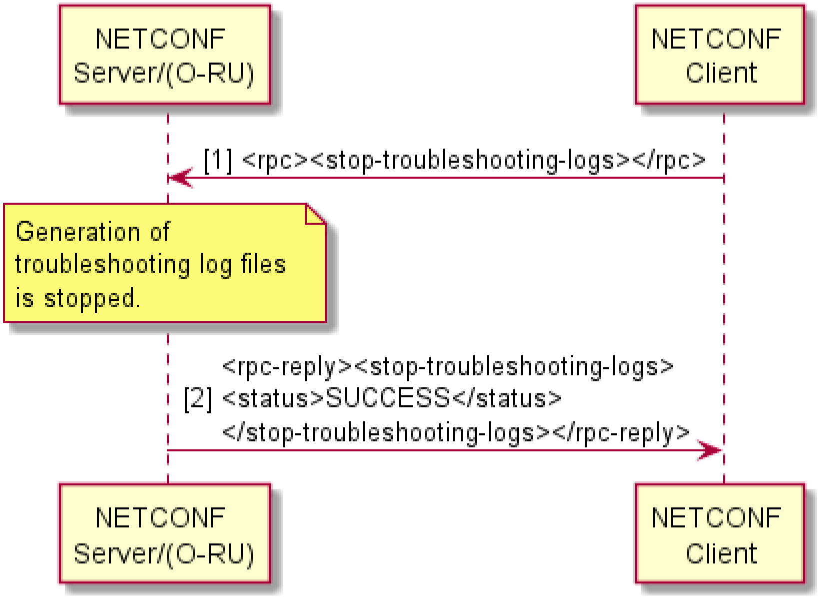


Figure 14.2.2-3: Stop Generation of Troubleshooting Logs

The overall troubleshooting log behaviour is illustrated in Figure 14.2.2-4. It contains 2 cases, successful notification case with illustrative notification contents and no notification as abnormal case.

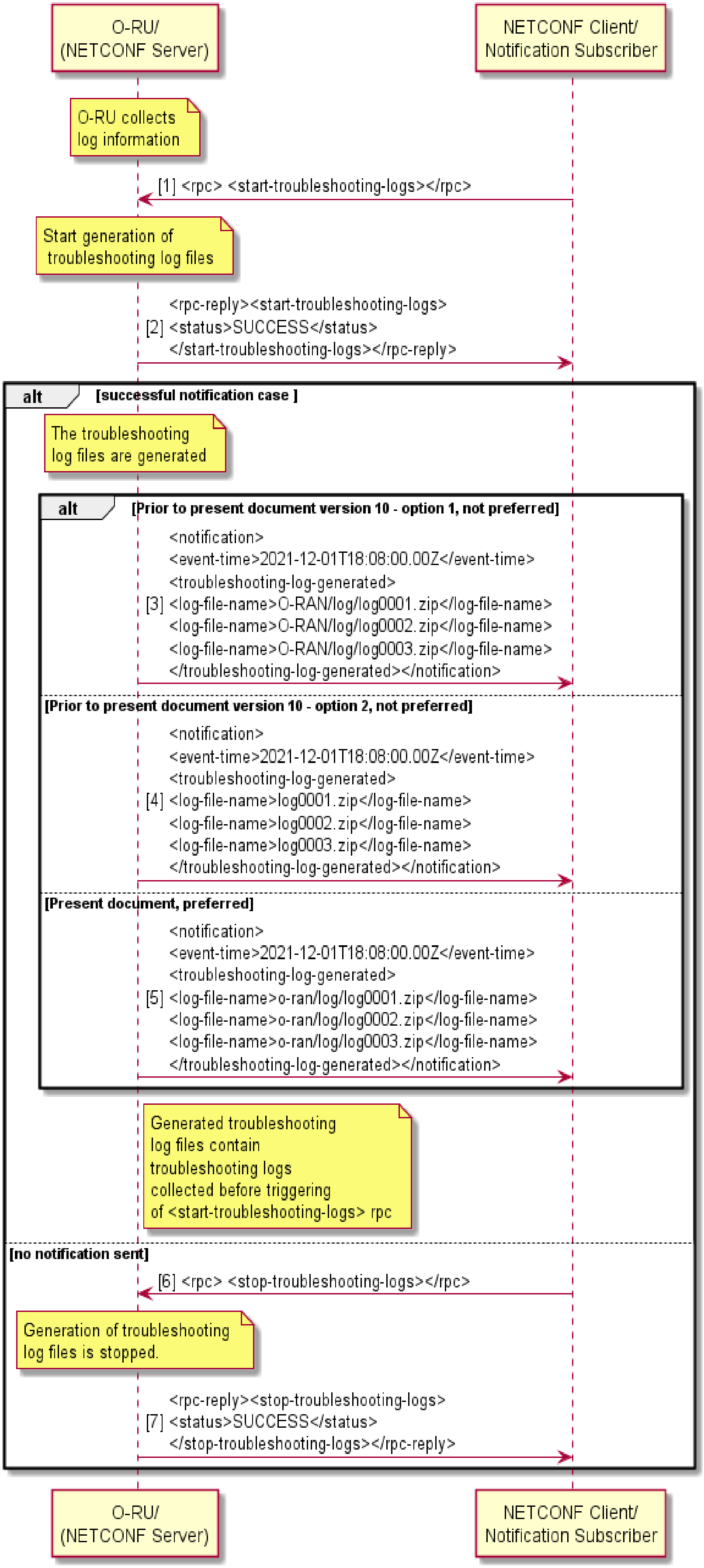


Figure 14.2.2-4: Overall Troubleshooting Log behaviour

NOTE: Prior to version 10 of the present document, this document contained inconsistent definition of the **log-file-name** string between Figure 14.2.2-4 and associated YANG models. In present document previously permitted content of node **log-file-name** is considered as not preferred. Not preferred file formats may be subject to removal in a future version of the present document.

14.3 Trace

By requesting trace log files an O-RU controller is able to get them. Those trace log files contain collected log data files that can be used for trace purposes.

The O-RU can provide 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 sent to O-RU controller whenever generated trace log files is(are) ready. File names 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) - assuming no additional upload restrictions from connectivity bandwidth or file server implementation.

NOTE: Timing of creating new group of trace log files is up to O-RU implementation.

After <stop-trace-logs> rpc received from O-RU controller, O-RU is mandated to stop collecting trace logs and start generating trace log files which contain log data already collected after previous <trace-log-generated> notification.

The overall procedure is shown in Figure 14.3-1, with illustrative notification contents. The figure shows the O-RU last <trace-log-generated> notification with the is-notification-last::‘true’ and file names of trace log files being signalled to the notification subscriber.

The file transfer mechanism for the created trace log files shall be handled by the file management described in clause 12 File Management.

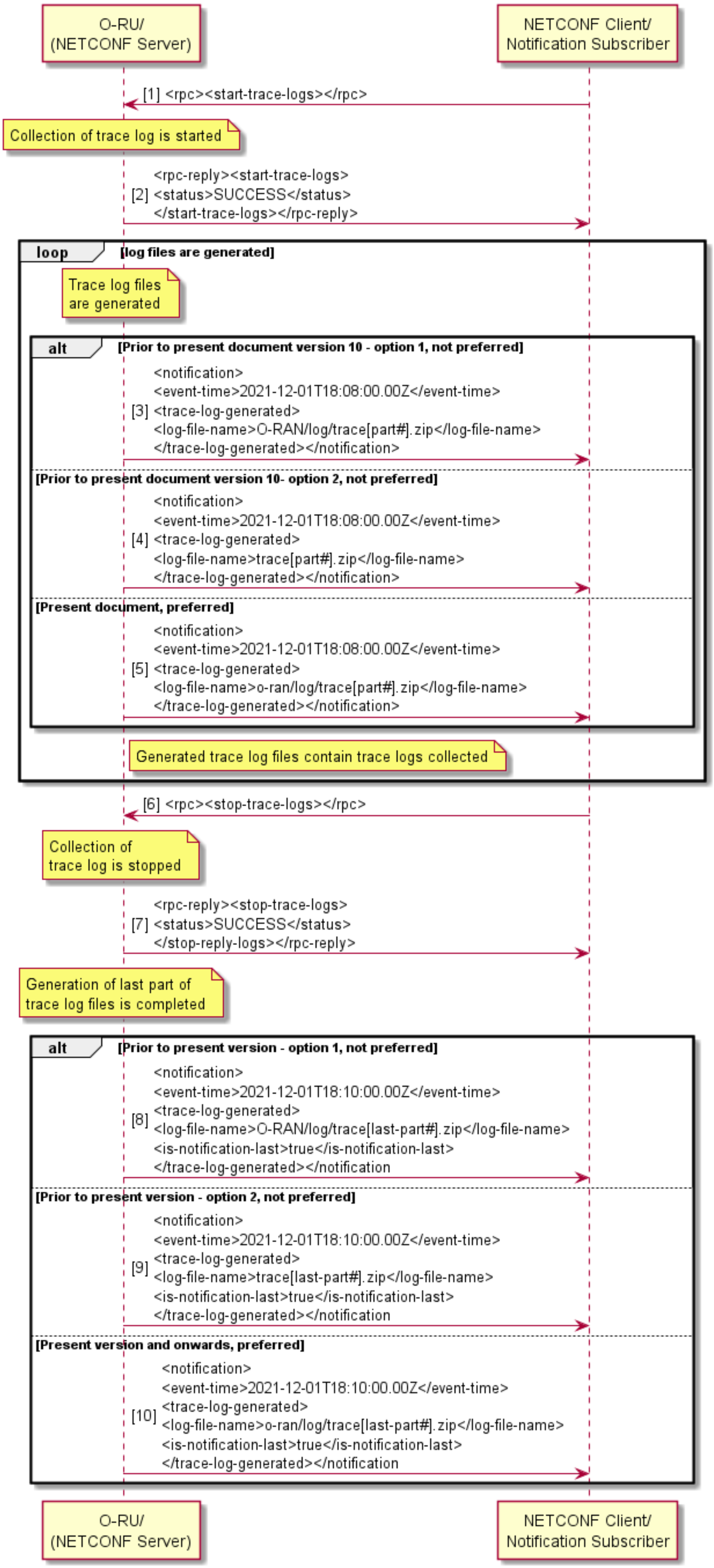


Figure 14.3-1: Overall Trace Log behaviour

NOTE: Prior to version 10 of the present document, this document contained inconsistent definition of the **log-file-name** string between Figure 14.4-1 and associated YANG models. In present document previously permitted content of node **log-file-name** is considered as not preferred. Not preferred file formats may be subject to removal in a future version of the present document.

14.4 Operational aspects of antenna line devices

14.4.1 Introduction

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 as specified in ISO/IEC 13239 [65]. Detailed information can also be found in TS 37.462 [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 37.466 [28].

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).

This clause describes the communication mechanisms based on AISG 2.0 protocol as specified in AISG 2.0 [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.

14.4.2 HDLC interworking

HDLC protocol is standardized by ISO/IEC 13239 [65]. Detailed information can also be found in TS 37.462 [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 as specified in TS 37.466 [28].

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 Figure 14.4.2-1 defines the reference architecture with functional split is defined:

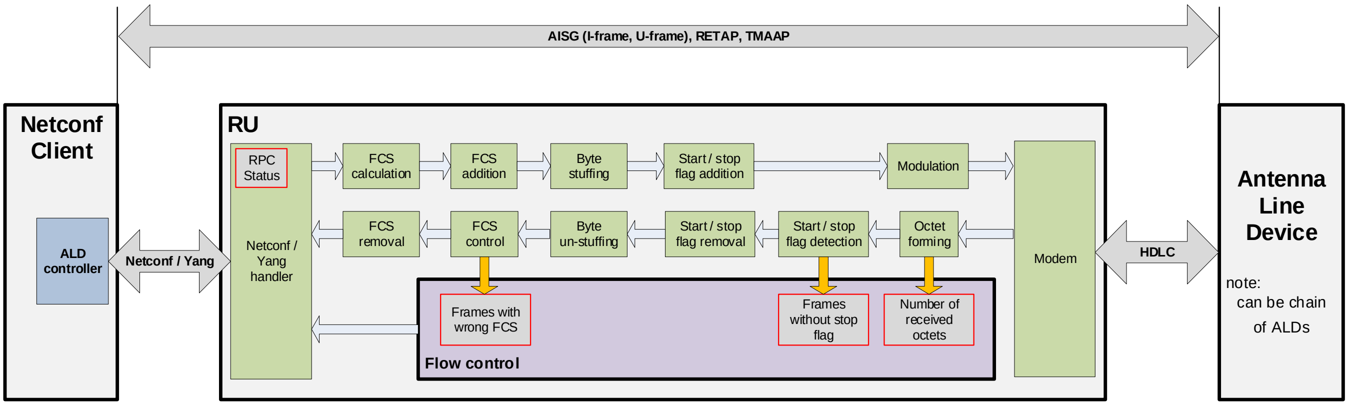


Figure 14.4.2-1: ALD Reference Architecture

The result of the above architecture is that below mentioned parts of HDLC message are processed by entities as illustrated in the Figure 14.4.2-2.

		AISG message				
Flag 1 octet	ADDR 1 octet	Control 1 octet	INFO N octets	FCS 2 octets	Flag 1 octet
0x7E	Secondary station address	Control bits	Variable length	CRC	0x7E
HDLC frame					

RU	Netconf Client	RU

Figure 14.4.2-2: Component's responsibility split.

14.4.3 ALD operations

Figure 14.4.3-1 illustrates the ALD transfer procedure. 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 37.462 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 1: For details of HDLC transmission and reception algorithm, please see TS 37.462, clause 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 2: 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 37.462, clause 4.5 "Message timing" for details.

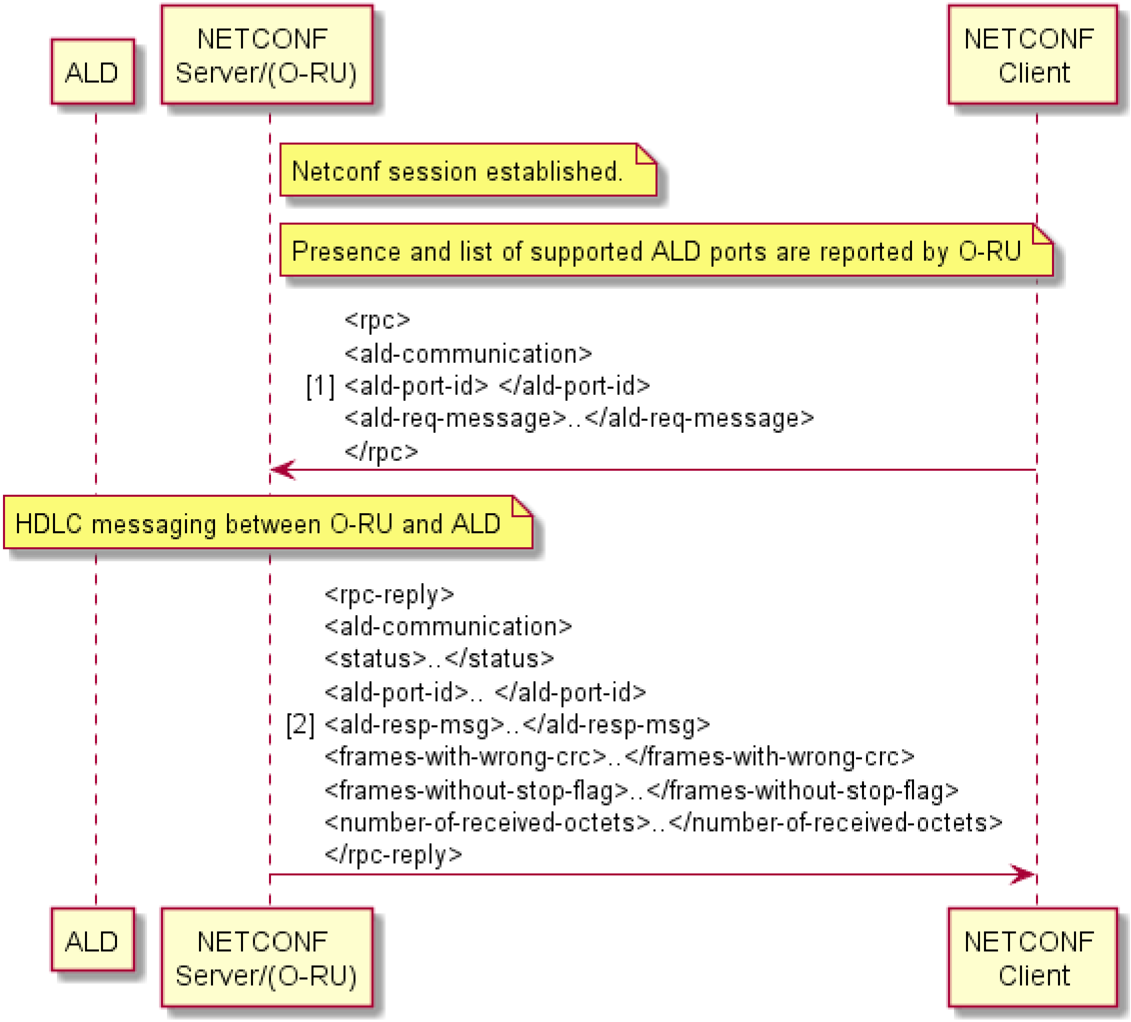


Figure 14.4.3-1: ALD Message Transfer

General scenario

Precondition:

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.

4) NETCONF Client determines presence of HDLC Secondary Devices.

5) NETCONF Client assigns HDLC addresses to desired HDLC Secondary Devices.

6) NETCONF Client initiates HDLC layer for secondary devices by sending SNRM command.

7) NETCONF Client starts polling procedure for every HDLC-addressed Secondary Device.

Postcondition:

Detected and addressed HDLC Secondary Devices are available for configuration.

14.5 Operational aspects of external IO

14.5.1 Introduction

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 includes 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.

14.5.2 External input

This clause 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, as illustrated in Figure 14.5.2-1.

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.

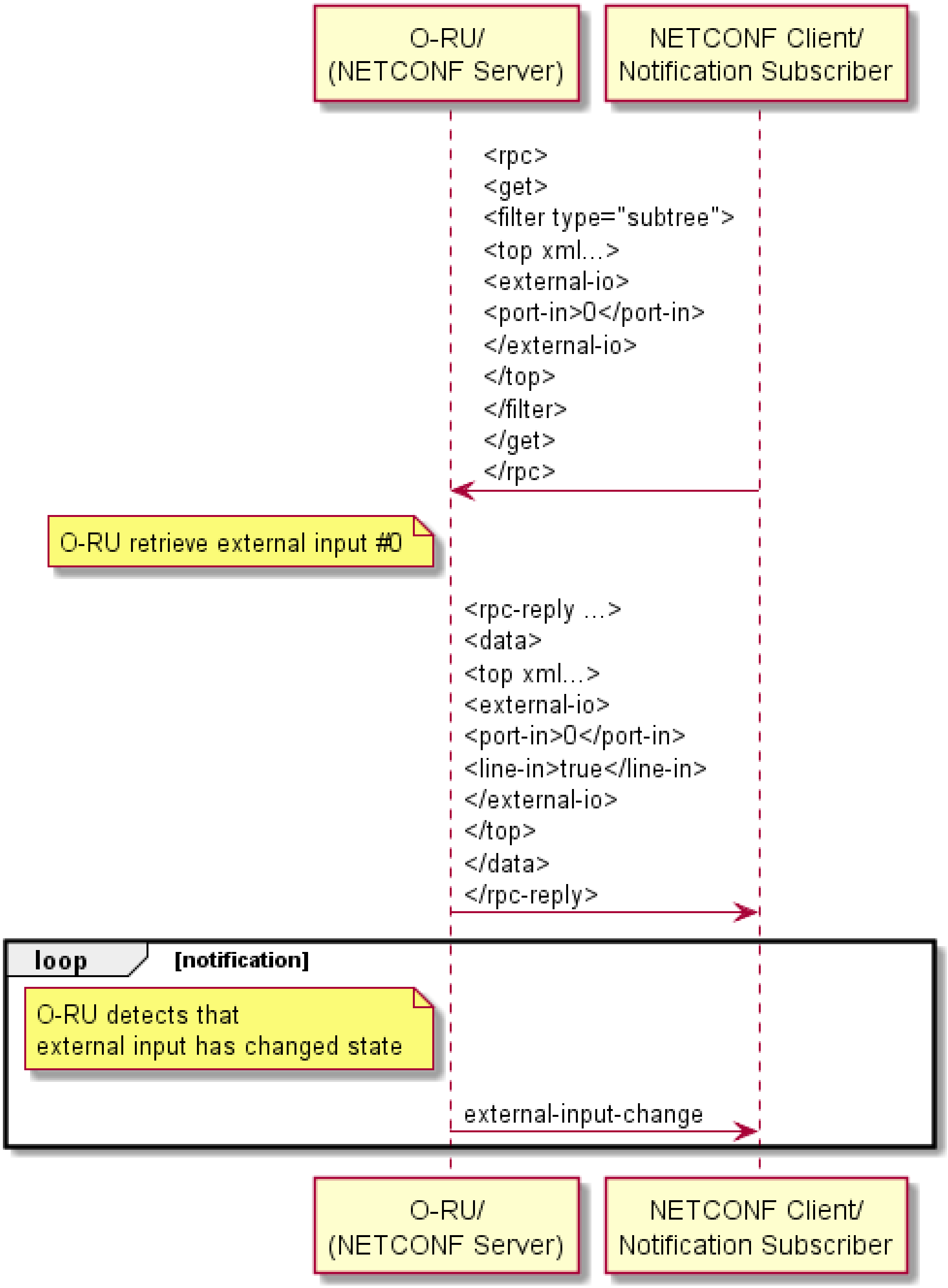


Figure 14.5.2-1: Retrieve external line-in

14.5.3 External output

This clause 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, as illustrated in Figure 14.5.3-1 and Figure 14.5.3-2.

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.

- FALSE: Circuit is closed

The default values shall be TRUE.

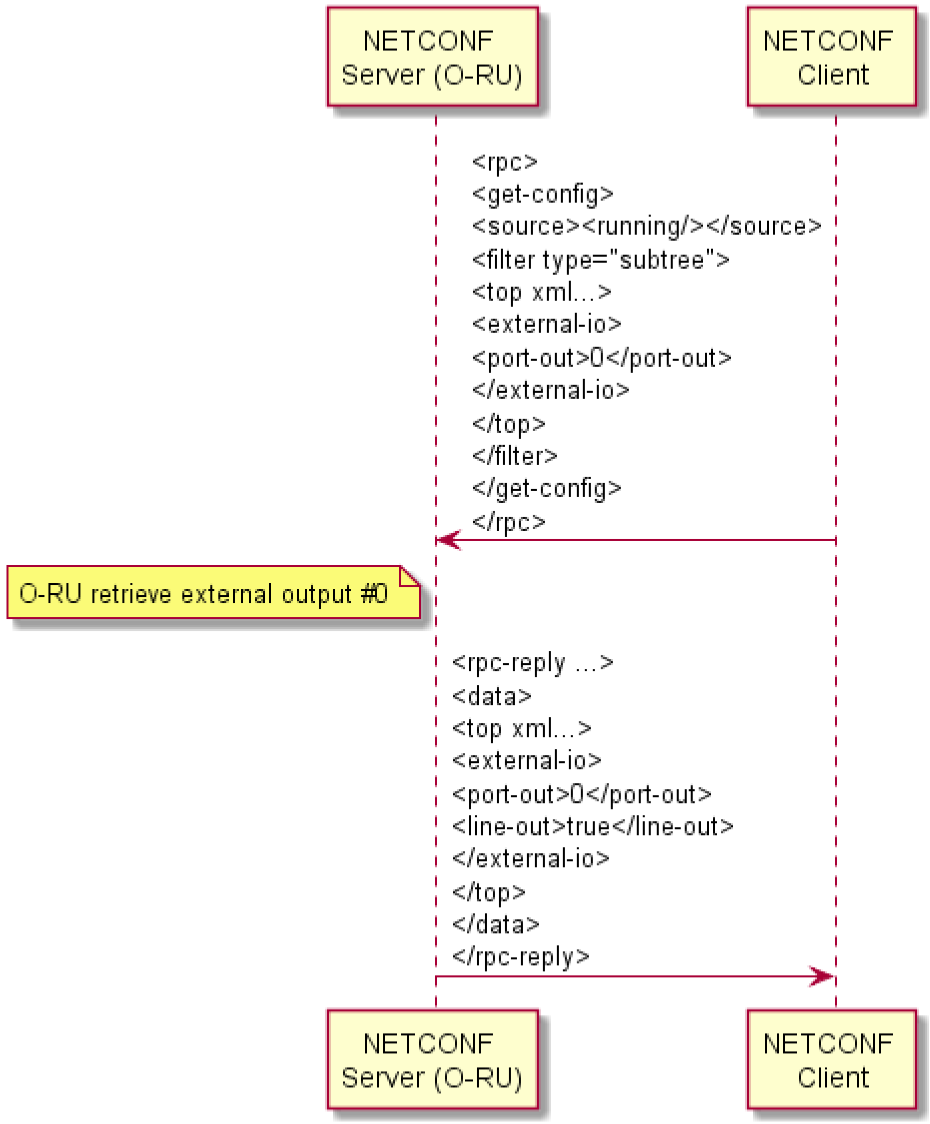


Figure 14.5.3-1: Retrieve external line-out

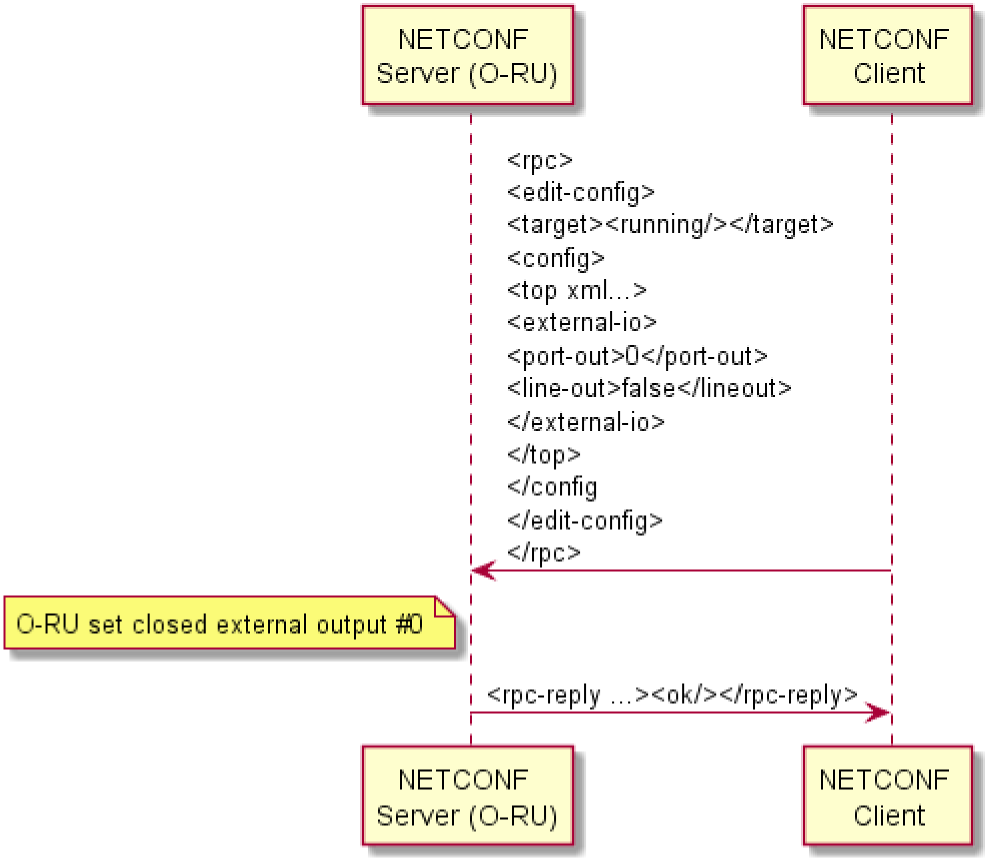


Figure 14.5.3-2: Control external line-out

14.6 O-RU connectors

14.6.1 Introduction to O-RU connectors

The O-RU may indicate presence and label of its physical connectors of specific type. The O-RU connectors can be exposed through o-ran-hardware.yang module as objects of **class** "O-RU-CONNECTOR". In the present document, coaxial antenna connectors serving for antenna feeders and beamforming-calibration lines are introduced.

14.6.2 Connectors related to antennas and antenna arrays

14.6.2.1 Antenna connectors

This clause describes usage of antenna connectors related to feeders (used to carry air interface signals between O-RU and external antennas or external antenna arrays) and beamforming calibration lines (used by O-RU i.e., to handle supplementary signals needed for external antenna arrays calibration).

Exposed O-RU Connectors of **class** "O-RU-ANTENNA-CONNECTOR" are based on **class** "O-RU-CONNECTOR" and are used in o-ran-uplane-conf.yang module to show how feeders and beamforming calibration lines are related to antenna arrays and their array elements. This allows to align configuration with physical cabling that technicians need to connect between O-RU and external physical antenna array.

14.6.2.2 Distinguishable types of antenna connectors

14.6.2.2.1 Feeder

Exposed O-RU connectors of **class** "O-RU-FEEDER" are based on **class** "O-RU-ANTENNA-CONNECTOR". Objects of this **class** are intended to represent physical connectors for coaxial feeders between O-RU and external antennas or antenna arrays. The feeder connector can be referenced by single array element of specific array, however in case the array element is shared between tx-array and rx-array or by multiband arrays, such O-RU connector is expected to be referenced by each of arrays containing array element fed with RF signals through such connector.

For this class of O-RU connectors, name shown in the model shall be identical as the label of physical connector visible on the O-RU's enclosure.

14.6.2.2.2 Beamforming calibration

Exposed O-RU connectors of **class** "O-RU-BF-CAL" are based on **class** "O-RU-ANTENNA-CONNECTOR". Objects of this **class** are intended to represent physical connectors for supplementary coaxial cables used in antenna calibration process for external antenna arrays. The beamforming calibration connector is to be referenced by group of array elements belonging to single array. This is also possible, that there will be references to few groups of array elements belonging to different antenna arrays (e.g., related tx-array and rx-array or arrays for different bands that share array elements).

For this class of O-RU connectors, name shown in the model shall be identical as the label of physical connector visible on the O-RU's enclosure.

15 Details of O-RU operations

15.1 Retrieval of O-RU information

This clause 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 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
- 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.

15.2 User plane message routing

15.2.1 Introduction

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 clause 5.4 of [2].

Precondition:

- M-Plane connectivity is established between NETCONF Client and NETCONF Server

15.2.2 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 clause 5.1.3.2.7 of [2], the eAxC_ID consists of four parameters: DU_Port_ID, RU_Port_ID, CC_ID and BandSector_ID. Order of parameters in eAxC_ID shall 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.

NOTE 1: [2] refers to the eAxC_ID parameters as DU_Port_ID, RU_Port_ID, CC_ID and BandSector_ID and this document and associated YANG models refer to the eaxc-id parameters as du-ports, ru-ports, cc-ids and band-sectors.

To handle flexible bit assignment, configurable bitmasks are defined for each parameter.

NOTE 2: 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**>

15.2.3 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 clause 15.2.2. Please refer to the eaxc-id parameter description in CUS plane specification clause 5.1.3.2.7 of [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

The O-RU shall reject any configuration that corresponds to a prohibited eaxc-id assignment.

15.2.4 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 1: 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 shall be linked to an existing rx-array-carrier element and existing processing-element element or existing transport-qualified-processing-element element when the processing element is configured by the list 'additional-transport-session-type-elements'. In this latter case, the O-RU shall use the /user-plane-configuration/low-level-rx-links/transport-qualified-processing-element schema nodes instead of the /user-plane-configuration/low-level-rx-links/processing-element schema node, an O-RU Controller shall still configure the processing-element schema node in the low-level-rx-links list with a leafref to valid ru-element, but this shall not be used by the O-RU.

- at the moment of creation, every low-level-tx-link shall be linked to an existing tx-array-carrier element and existing processing-element element (or existing transport-qualified-processing-element element when the processing element is configured by the list 'additional-transport-session-type-elements'. In this latter case, the O-RU shall use the /user-plane-configuration/low-level-tx-links/transport-qualified-processing-element schema nodes instead of the /user-plane-configuration/low-level-tx-links/processing-element schema node, an O-RU Controller shall still configure the processing-element schema node in the low-level-tx-links list with a leafref to valid ru-element, but this shall not be used by the O-RU.

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
   - static_low-level-tx-endpoint elements - by fetching the list **static-low-level-tx-endpoints** in o-ran-uplane-conf.yang
   - static_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
   - O-RU's connectors - by fetching list of **related-o-ru-connectors** (if exist) in o-ran-uplane-conf.yang

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 shall 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
   - o-ru-connectors and [tr]x-array elements in o-ran-uplane-conf.yang (if available)
   - interfaces and port elements in ietf-interfaces.yang (with o-ran port-reference augmentation)

NOTE 2: 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 static_low-level-rx-endpoint elements and static_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) and ports (suitable in terms of capabilities and able to transfer signals from/to a desired interface). Details of configuring **interfaces** and **processing-elements** are described in clause 7.

7) Once transport layer is configured, O-DU may perform initial verification of C/U Plane Transport Connectivity as described in clause 7.6 – with respect to content of list "restricted-interfaces" every desired endpoint is reachable through.

8) NETCONF Client creates low-level-tx-endpoints and low-level-rx-endpoints related to static-low-level-tx-endpoints and static-low-level-rx-endpoints determined in step 5) as suitable for desired configuration. NETCONF Client assigns unique eaxc-id(s) values to every created low-level-[tr]x-endpoint.

NOTE 3: Uniqueness of eaxc-id is mandatory within the O-RU in the same direction (Tx or Rx) even across interface elements having relationship to low-level-rx-endpoint elements or low-level-tx-endpoint elements.

NOTE 4: In case NETCONF Client wants particular value of eAxC_ID to be used for non-time managed traffic, NETCONF Client shall assign this eAxC_ID to parameter "eaxc-id" belonging to low-level-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 shall 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 shall configure DSS by using different eAxC ids (i.e., different endpoints) as described in CUS plane specification clause 7.3.6 of [2].

Table 15.2.4-1: Centre Bandwidth Calculation

Type	N_{RB}	Centre of channel bandwidth (same as F_{REF} as defined in clause 5.4.2.1 of 3GPP TS38.104 Section 5.4.2.1 [66])
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	Centre of k^{th} RE of n_{PRB} RB
	N_{RB}mod2=0	

The parameters k, n_{PRB} and N_{RB} referenced here are specified in Table 5.4.2.2-1 of 3GPP TS 38.104.

10) 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 shall be followed. This is illustrated in Figure 15.2.4-1.

NOTE 5: 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], clause 5.3.

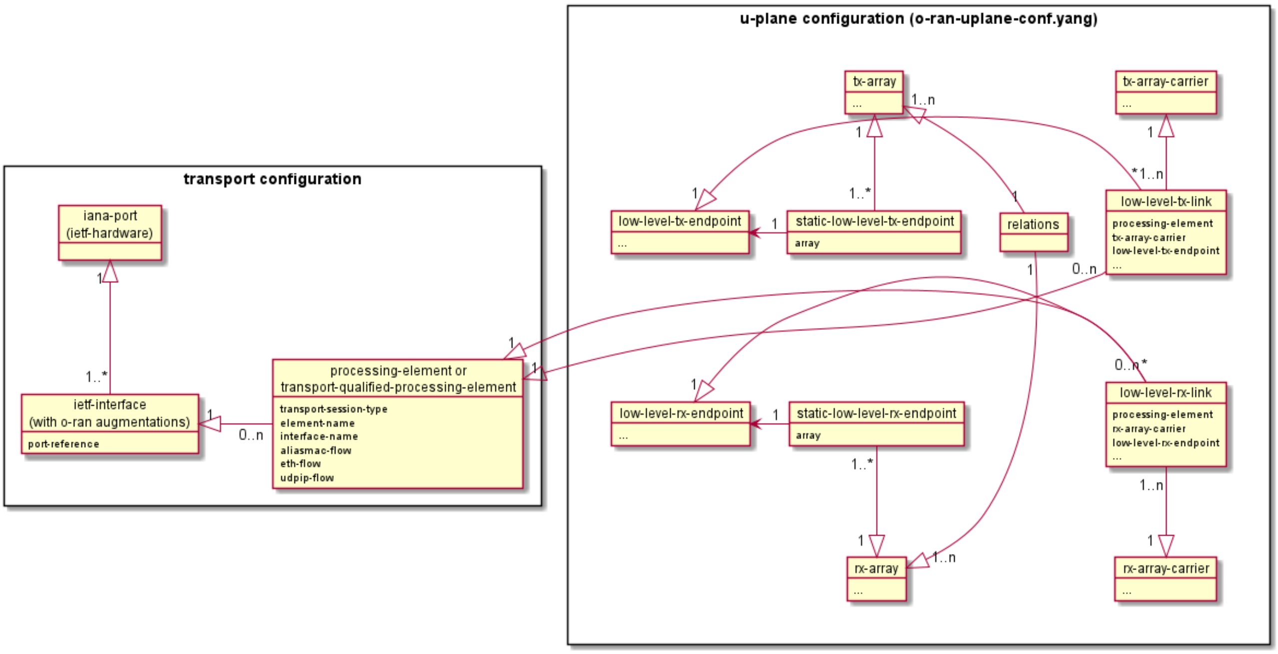


Figure 15.2.4-1: Diagram showing relations between CU-Plane and Carrier configuration elements

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 clause 15.3.

15.3 Carrier configuration

15.3.1 Carrier creation

This clause provides basic scenario for carrier creation procedure. Precondition for below steps is to fulfil steps from clause 15.2.4

1) NETCONF Client creates the **tx-array-carriers** in relation to the desired **tx-arrays**.

NOTE 1: Generally the number of **tx-array-carriers** is 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 2: Generally the number of **rx-array-carriers** is 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 low-level-tx-endpoints and low-level-rx-endpoints related to desired static-low-level-tx endpoints and static-low-level-rx-endpoints respectively.

5) NETCONF Client creates the **low-level-tx-links** containing relationship to the existing **tx-array-carriers**, **low-level-tx-endpoints** and existing **processing-elements**.

6) 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.

15.3.2 Activation, deactivation and sleep

The NETCONF Client performs activation of a tx/rx-array-carrier by setting the value of the parameter "active" at tx-array-carrier element / rx-array-carrier element to "ACTIVE".

The NETCONF Client performs deactivation of a tx/rx-array-carrier 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".

O-RU shall not autonomously change [tr]x-array-carrier::**active** to “ACTIVE”.

For detailed description of tx-array-carriers and rx-array-carriers please refer to *description* substatement in YANG models.

Figure 15.3.2-1 shows possible transitions and values combination to be followed by "active" and "state". Combination or transitions outside of below diagram is not allowed.

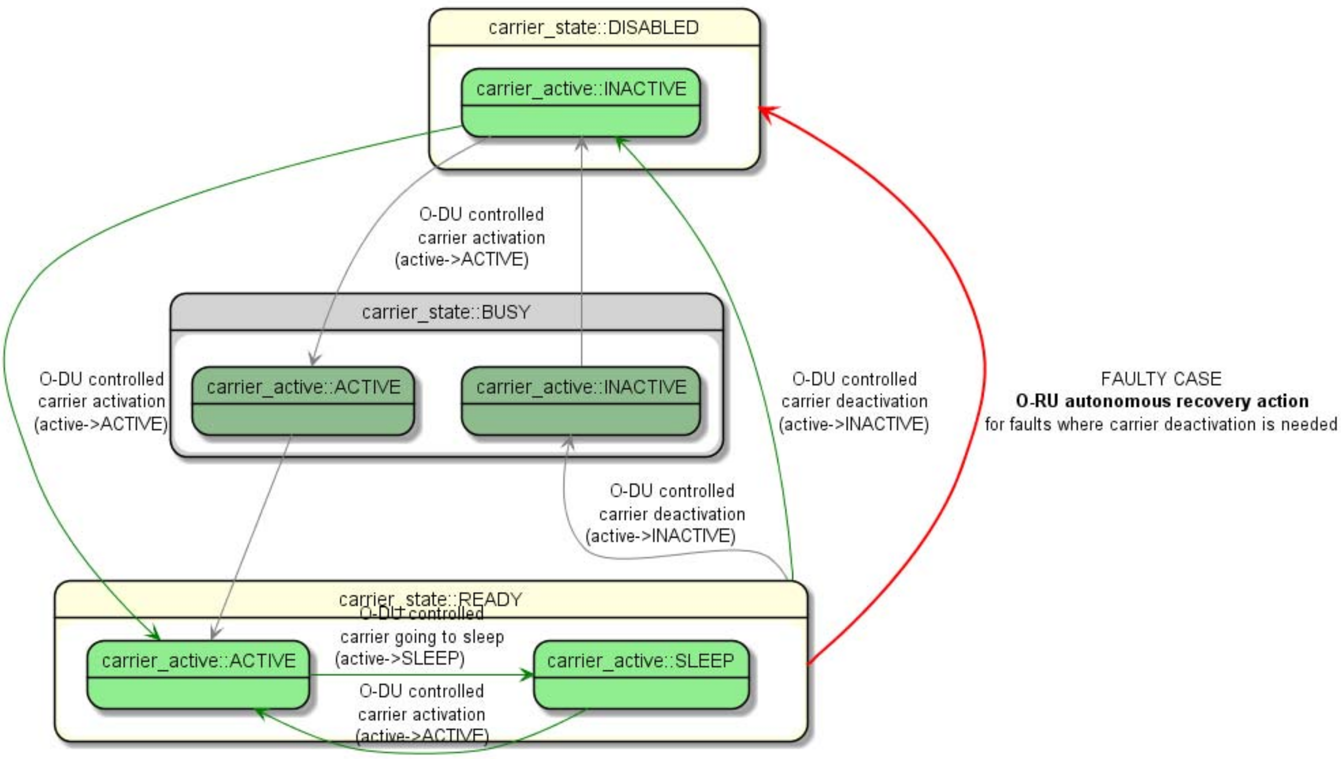


Figure 15.3.2-1: Diagram showing all possible transitions and combination of "active" and "state" parameters

NOTE: BUSY state is only available during transition and existence of this state depends on internal O-RU design.

15.3.3 Carrier state's relationship to synchronisation state

15.3.3.1 Synchronisation state and carrier state transitions

The O-RU's **tx-array-carrier** and **rx-array-carrier** states depend upon the O-RU's **sync-state**. The flow chart shown in Figure 15.3.3.1-1 illustrate the possible transitions and associated parameters values for the two array-carrier in the possible synchronisation state. When the O-RU implements HOLDOVER, **tx-array-carrier** and **rx-array-carrier** possible states and transitions are the same for sync LOCKED and HOLDOVER mode. When O-RU transitions to the FREERUN state the only possible tx-array-carrier and rx-array-carrier state is DISABLE/INACTIVE. Figure 15.3.3.1-1 shows possible transitions and according to parameters value combination.

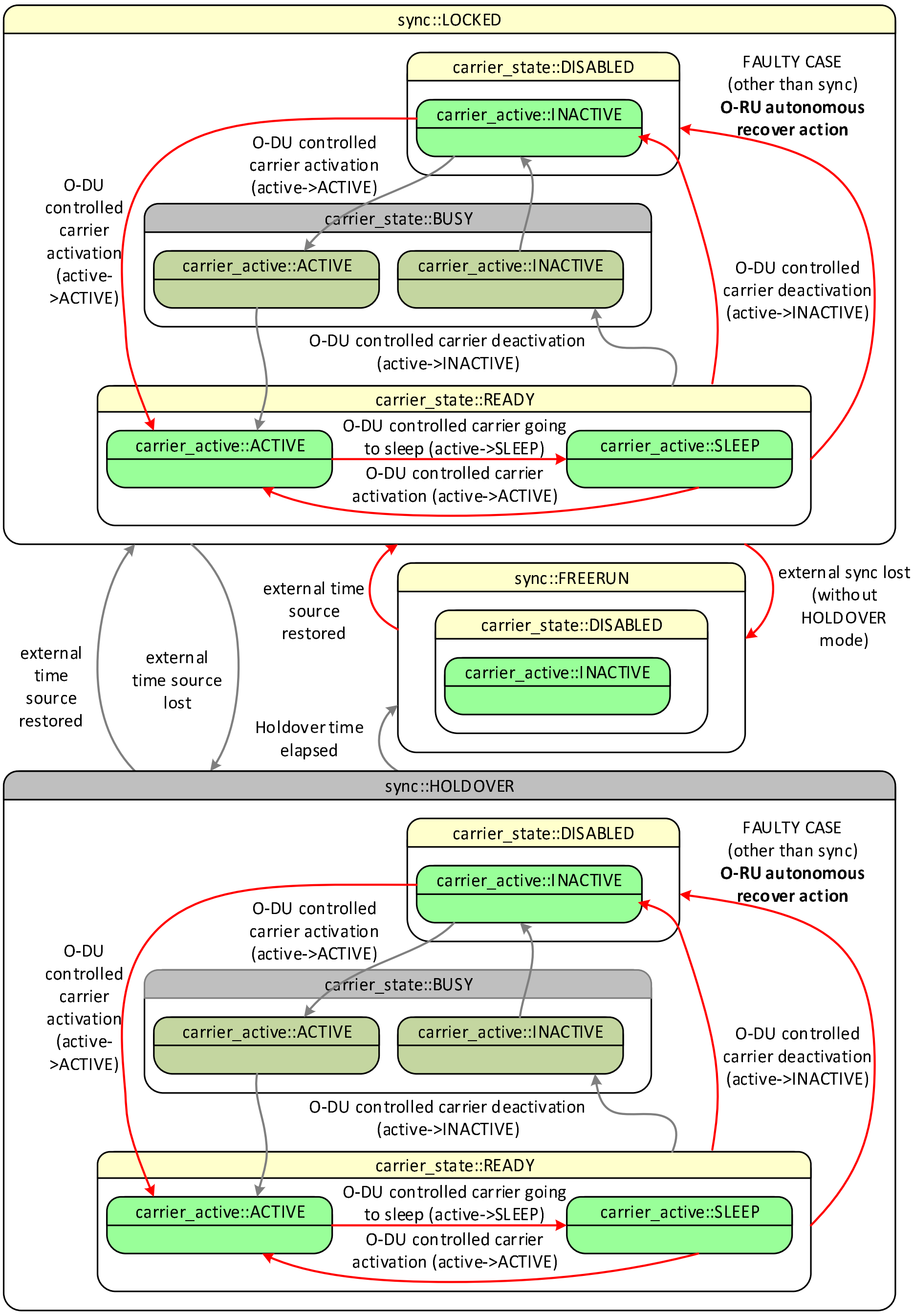


Figure 15.3.3.1-1: Combination of tx-array-carrier/rx-array-carrier transitions and allowed states compared to sync state

When an O-RU changes its synchronisation state, in addition to changing tx-array-carrier and rx-array-carrier, the O-RU can and normally would report these changes via notifications. The O-RU communicates this sync state information to all NETCONF clients subscribed to this O-RU's synchronization-state-change notification. This enables O-DU(s) which are subscribed to an O-RU to maintain carrier state for that O-RU and react to changes in the O-RU's carrier state.

The following two sections describe the tx-array-carrier/rx-array-carrier behaviour and O-RU/O-DU communication when synchronisation is lost, and when synchronisation is restored.

15.3.3.2 Synchronization lost and HOLDOVER mode expired

Figure 15.3.3.2-1 shows the **tx-array-carrier** and **rx-array-carrier** behaviour and illustrates the communication between the O-RU and O-DU when O-RU loose synchronisation and enters to FREERUN mode. The process can be divided into 5 steps as shown in the figure and briefly described below.

1) If the O-RU implements the optional HOLDOVER state, once the O-RU detects that it has lost connection to its timing source and its **sync-state** moves to HOLDOVER, the O-RU will send a **synchronization-state-change** notification to one (or more) subscribed O-DU(s) indicating that the O-RU is in HOLDOVER. If the O-RU does not implement HOLDOVER, then this step is skipped.

2) After the O-RU HOLDOVER timer expires, or if the O-RU does not implement HOLDOVER, the O-RU moves the **sync-state** to FREERUN and sets the **rx-array-carrier** and **tx-array-carrier** to the **active**=INACTIVE/**state**=DISABLED states. Moving to this state shall cause the O-RU to stop sending traffic to O-DU.

3) The O-RU generates a **synchronization-state-change** notification to all subscribed O-DUs indicating that the O-RU is in FREERUN.

4) The O-RU also sends **rx-array-carriers-state-change**, and **tx-array-carriers-state-change** notifications to all subscribed O-DUs indicating that both array carriers are INACTIVE/DISABLED.

5) After having received the notifications from the O-RU, the O-DU sets its copy of the **rx-array-carrier** and **tx-array-carrier** parameters to INACTIVE/DISABLED. The O-DU then shall stop sending traffic to the O-RU and ignore any residual traffic from the O-RU.

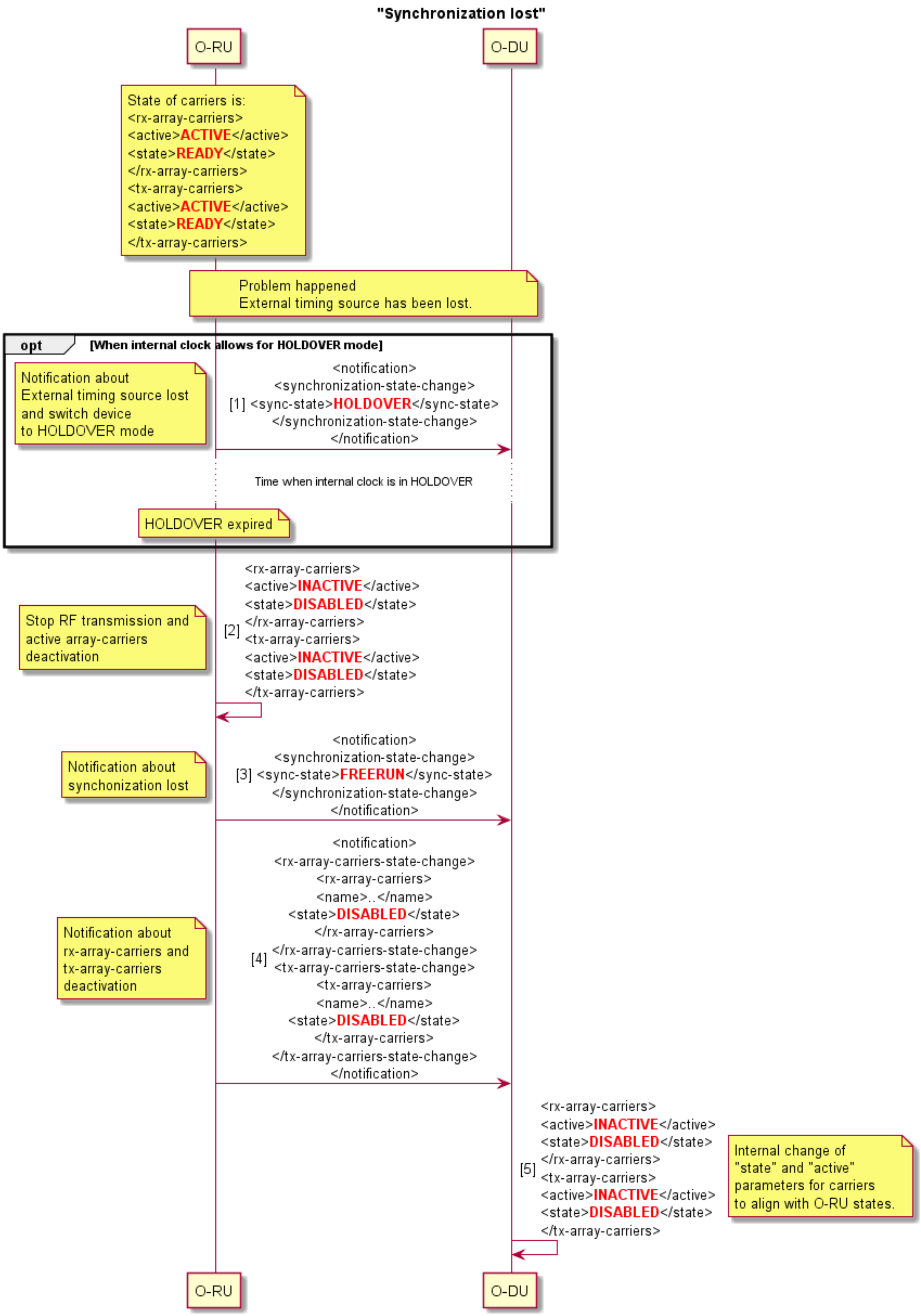


Figure 15.3.3.2-1: Synchronization lost scenario

15.3.3.3 External timing source restored

Figure 15.3.3.3-1 shows how **tx-array-carrier** and **rx-array-carrier** can be reactivated when external timing source is restored, and it illustrates the communication between O-RU and O-DU to restore operation between O-RU and O-DU.

The restoration process can be divided into 4 steps as shown in Figure 15.3.3.2-1 and described below.

1) Once the O-RU detects that synchronisation has been restored, it changes the sync-state to LOCKED, and then sends a **synchronization-state-change** notification to all subscribed O-DUs.

2) The O-RU then waits for the O-DU to send back a notification indicating that **tx-array-carrier** and **rx-array-carrier**'s **active** parameter is in the ACTIVE state.

3) Once the O-RU has received the ACTIVE indication, it then brings **tx-array-carrier** and **rx-array-carrier** state to READY.

4) After the array carriers are in the **state**=READY state, the O-RU sends **rx-array-carrier-state-change** and **tx-array-carrier-state-change** notifications to the O-DU(s) to inform the O-DU that it is ready to resume operation.

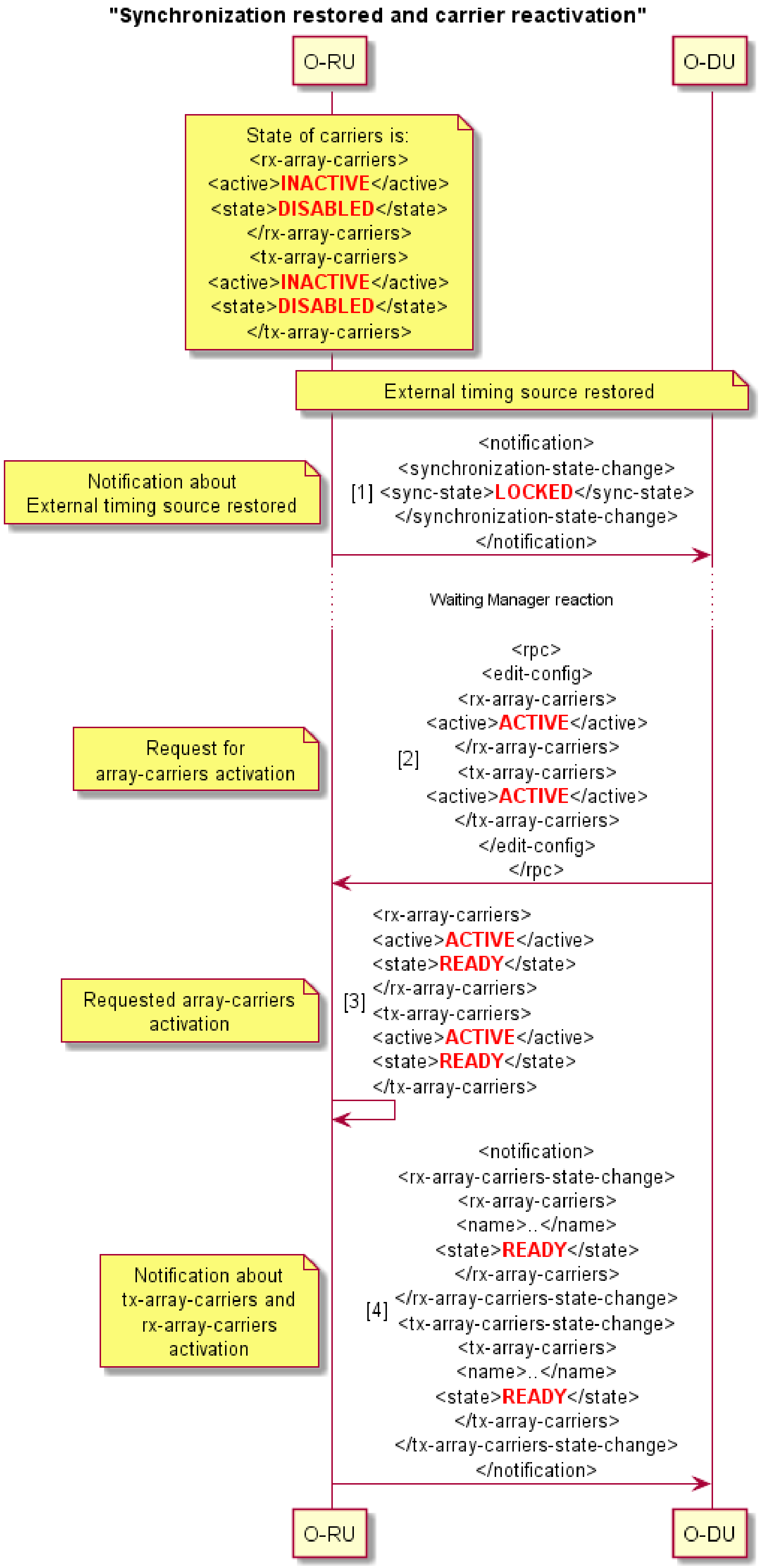


Figure 15.3.3.3-1: Synchronization restored and carrier reactivation scenario

15.4 Beamforming

15.4.1 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.

An O-RU indicates the support of a beamforming mode of operation per endpoint. Endpoints capable of either BF or non-BF operation are distinguished by the capabilities exposed by endpoint-types in endpoint-beam-capacity grouping.

Arrays capable of BF operation are distinguished by the relation to the endpoints and the reference from o-ran-beamforming yang module.

15.4.2 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.

15.4.3 Beamforming configuration update

This clause 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 clause 8. The O-RU shall locate the beamforming configuration file in the generic folder, i.e., O-RAN/beamforming/ or 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/ or 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 clause. 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 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.

Figure 15.4.3-1 and Figure 15.4.3-2 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.

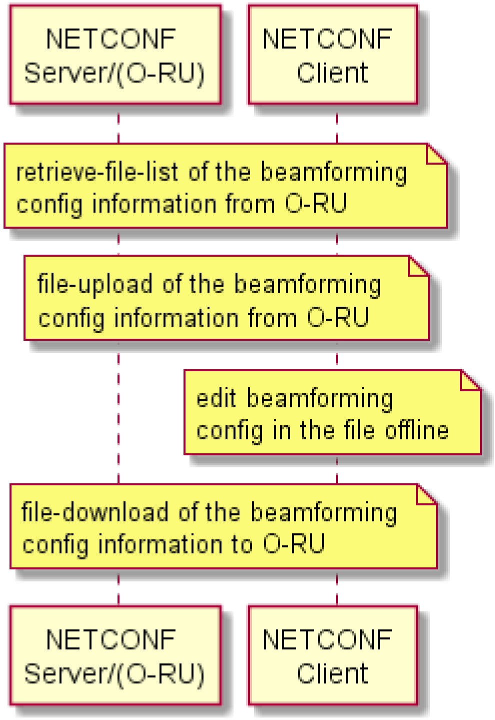


Figure 15.4.3-1: Method to Modify the File of Beamforming Configuration Information

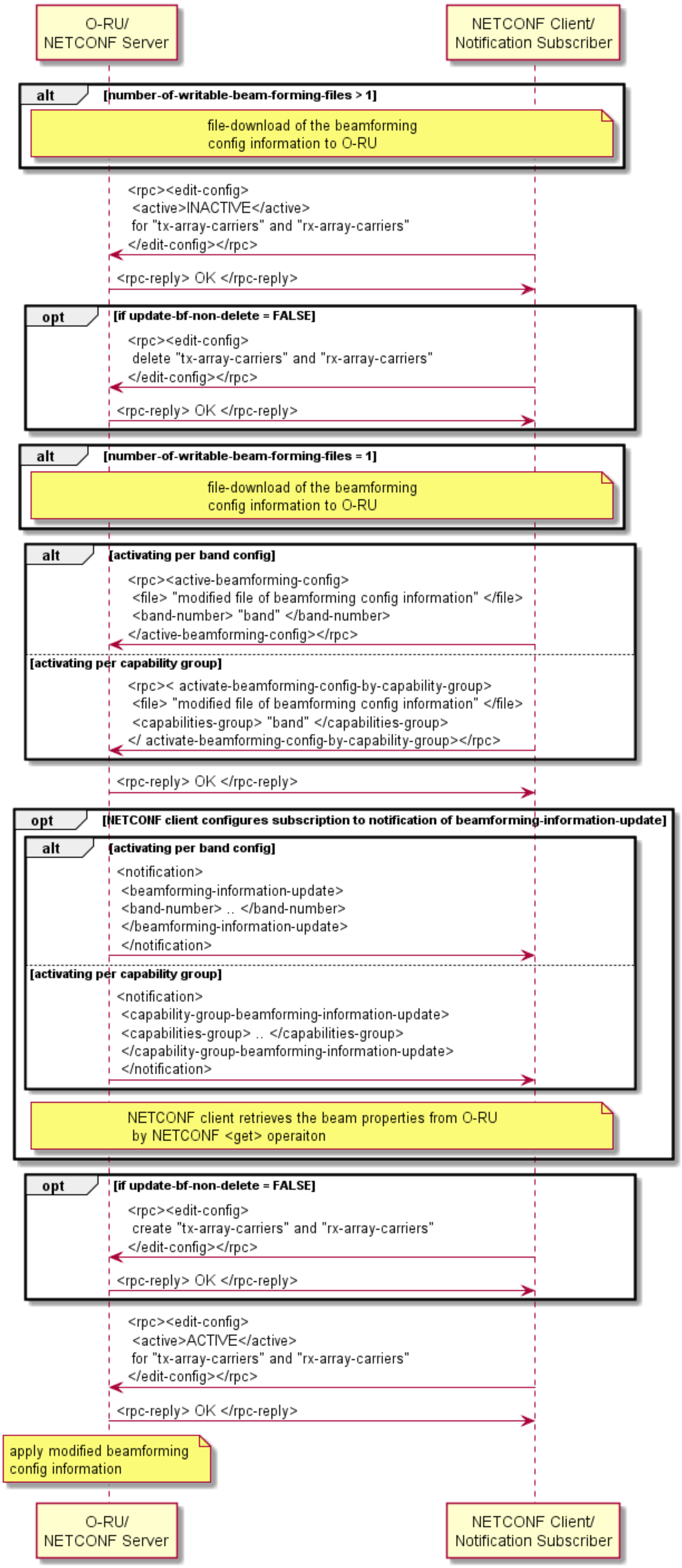


Figure 15.4.3-2: Method to Apply the modified file for Beamforming Configuration Information

15.4.4 Tilting pre-defined beams

This clause 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 clause 14, or modifying the beamforming configuration using the "**MODIFY-BF-CONFIG**" feature described in clause 15.4.3.

NOTE 1: 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 can 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 2: **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).

Figure 15.4.4-1 shows the sequence diagram for predefined-beam-tilt-offset-information. 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, the O-RU shall ensure that at least one of the **elevation-tilt-offset-granularity** and **azimuth-tilt-offset-granularity** is 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** and the configuration values should meet the ranges and granularity information retrieved from **predefined-beam-tilt-offset-information**.

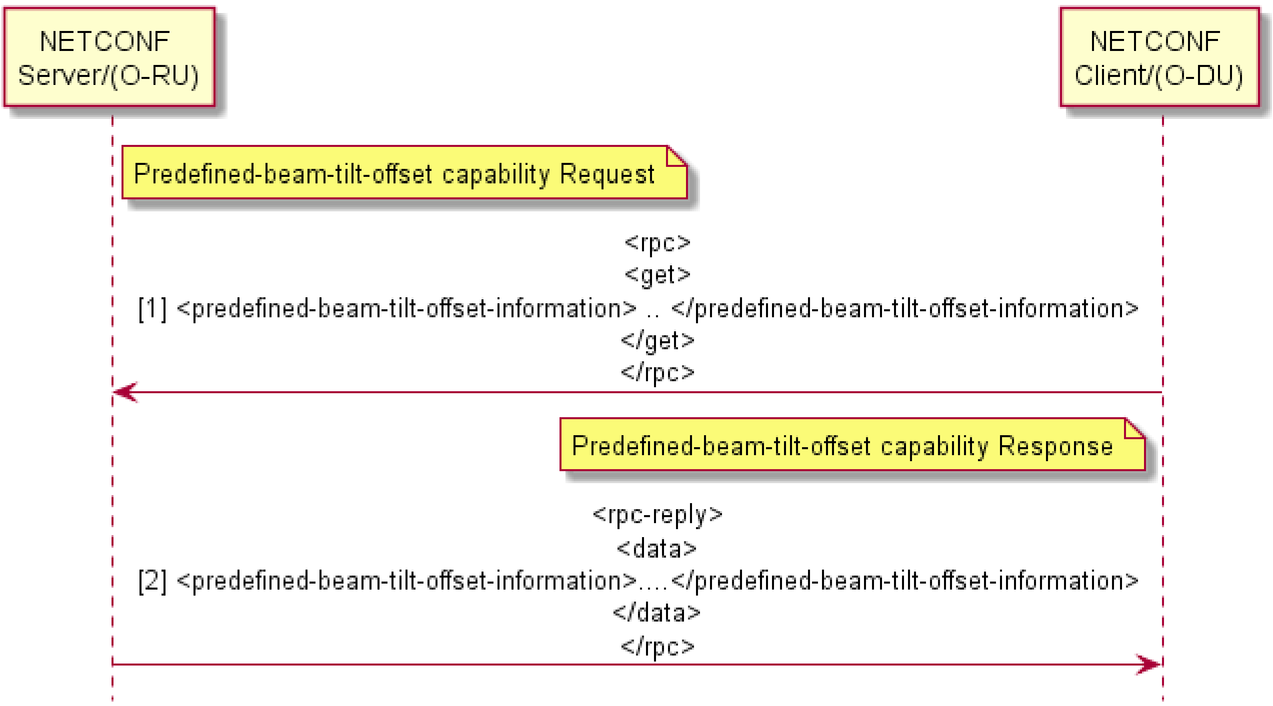


Figure 15.4.4-1: Sequence diagram for predefined-beam-tilt-offset-information

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 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 required.

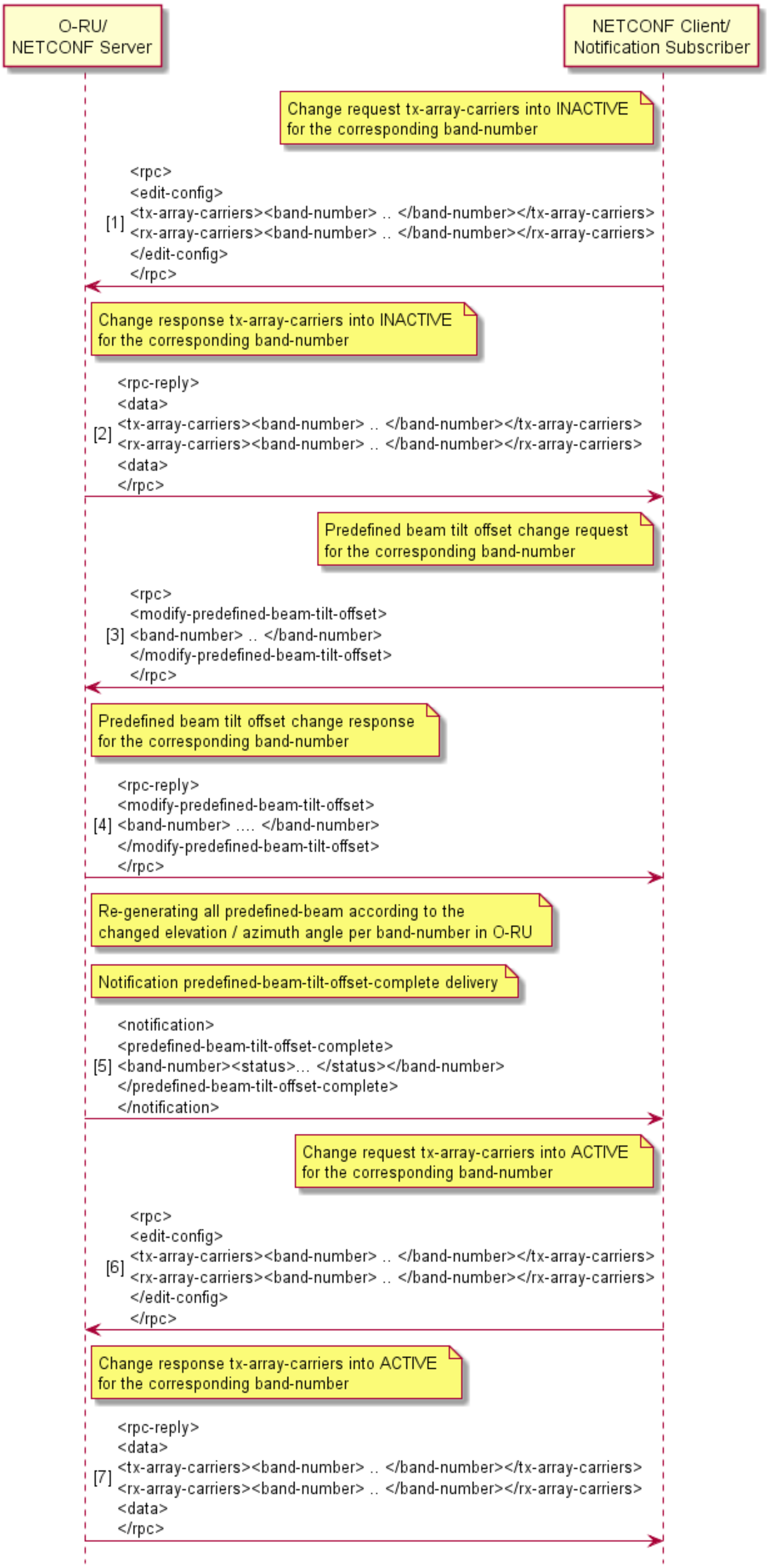


Figure 15.4.4-2: Procedure for the predefined-beam-tilt-offset

15.4.5 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 beamforming-trough-attributes-supported (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 ORAN-WG4.CUS specification, clause "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: Neighbourhood 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 the present document.

15.5 Antenna calibration

15.5.1 Background

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 clause, 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-required** 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.

15.5.2 Overall operation

15.5.2.1 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.

15.5.2.2 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 the present document. 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, or, when the O-RU supports the optional O-RU-COORDINATED-ANT-CAL feature, by using the **antenna-calibration-coordinated** notification.

Upon reception of the **antenna-calibration-required** 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.

When **coordinated-calibration-support** is set to true, this indicates that the O-RU is able to determine a priori the time-frequency resources required for antenna self-calibration and the O-RU uses **antenna-calibration-coordinated** notification to indicate these to the O-DU instead **of antenna-calibration-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. An O-RU may also report the optional capability of configured-preparation-timer-supported which indicates that it supports configuration of the preparedness timer that controls how far in advance of the coordinated self-calibration procedure the O-RU is required to send the notification of impacted resources. If configured by the NETCONF Client, the O-RU shall send the **antenna-calibration-coordinated** notification at least **coordinated-ant-calib-prep-timer** seconds before the operation of the coordinated antenna calibration procedure. If such an optional capability is not supported, the O-RU shall indicate that time-frequency resources are sent to a subscribed O-DU at least 60 seconds before the operation of the coordinated antenna calibration procedure. SFN wrap around will occur multiple times during these 60 seconds, this is handled according to statements in clause 15.5.2.4.

The O-DU shall not send a **start-antenna-calibration** RPC request when a coordinated antenna calibration period is in progress. The O-RU is allowed to reject such a request if it is received during a coordinated antenna calibration period. An O-DU receiving an **antenna-calibration-coordinated** 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.

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.

15.5.2.3 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 clause 11.4.

During the self-calibration, i.e., when O-RU exposes **self-calibration-support** with value TRUE and when O-DU sets **self-calibration-allowed** with value TRUE to O-RU, there could be no coordination of time-frequency resources between the O-RU and O-DU. The O-DU can continue to schedule user data during calibration process using the resources identified in **antenna-calibration-required** notification without impacting the operation of the calibration procedure. Scheduled user data will be affected by ongoing calibration procedure.

15.5.2.4 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.

15.5.2.5 Antenna calibration procedure

Figure 15.5.2.5-1 shows the overall operation for antenna calibration.

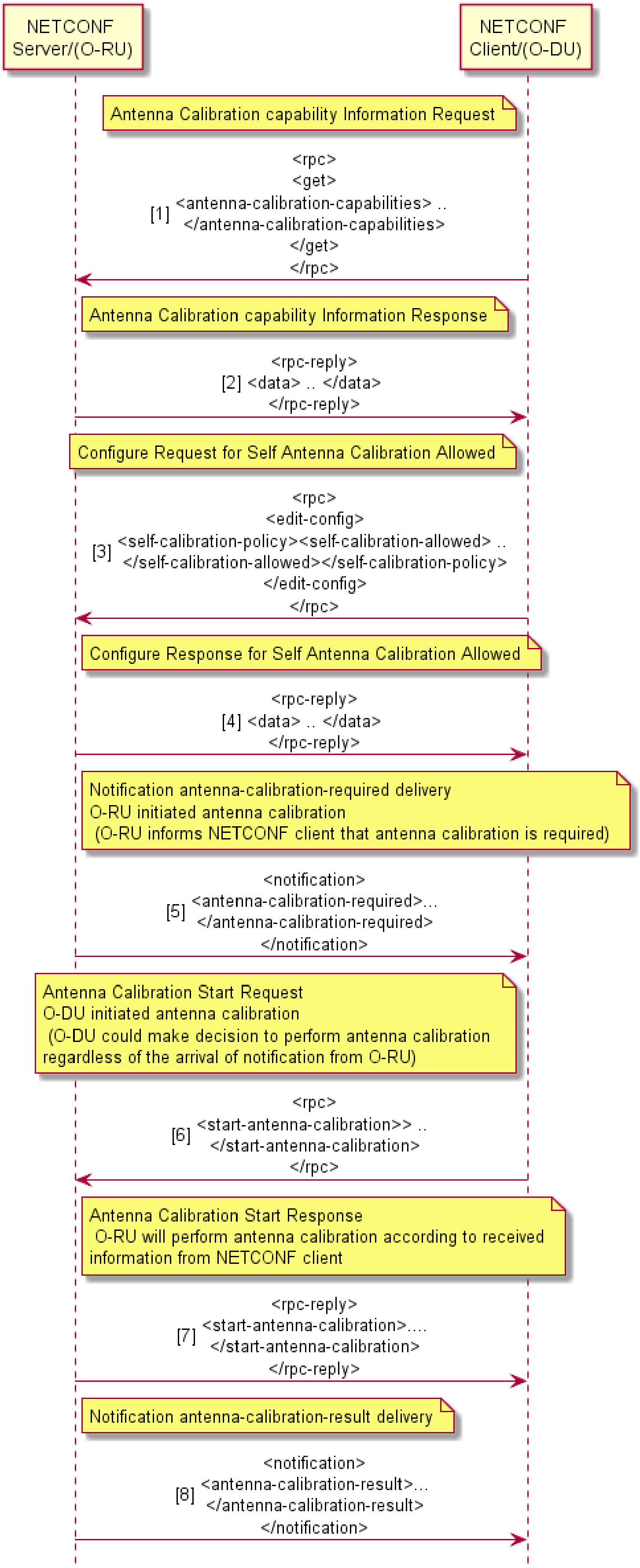


Figure 15.5.2.5-1: Overall of antenna calibration operation

15.5.3 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.

Figure 15.5.3-1 shows the relationship between the various antenna calibration capabilities parameters described above.

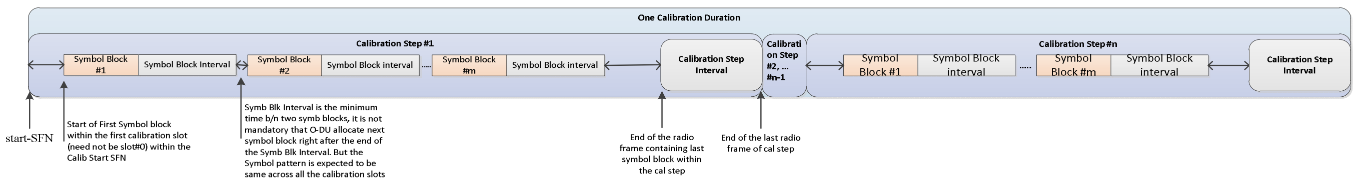


Figure 15.5.3-1: Relationship among Antenna Calibration Capability parameters

15.5.4 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.

- **end-calibration-frequency-ul**: indicates the highest frequency value in Hz of the frequency range is required for UL antenna calibration operation.

15.5.5 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 1: 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.

Table 15.5.5-1 lists the parameters configured in the O-RU using the "**start-antenna-calibration** RPC request"

Table 15.5.5-1: Antenna Calibration Parameters

Parameters	Type / Range	Descriptions
symbol-bitmask-dl	string	Bitmask indicating DL calibration symbol within a calibration slot. First character in the string indicates first symbol, next character in the string indicates 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 indicates first symbol, next character in the string indicates 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 indicates first slot, next character in the string indicates 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 indicates first slot, next character in the string indicates 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 indicates first radio frame equal to the start-SFN, next character in the string indicates 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 indicates first radio frame equal to the start-SFN, next character in the string indicates 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

15.5.6 Example antenna calibration operation

This clause 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 15.5.6-1.

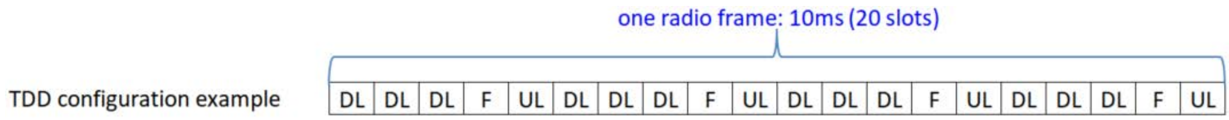


Figure 15.5.6-1: Example of TDD configuration

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 allocates 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.

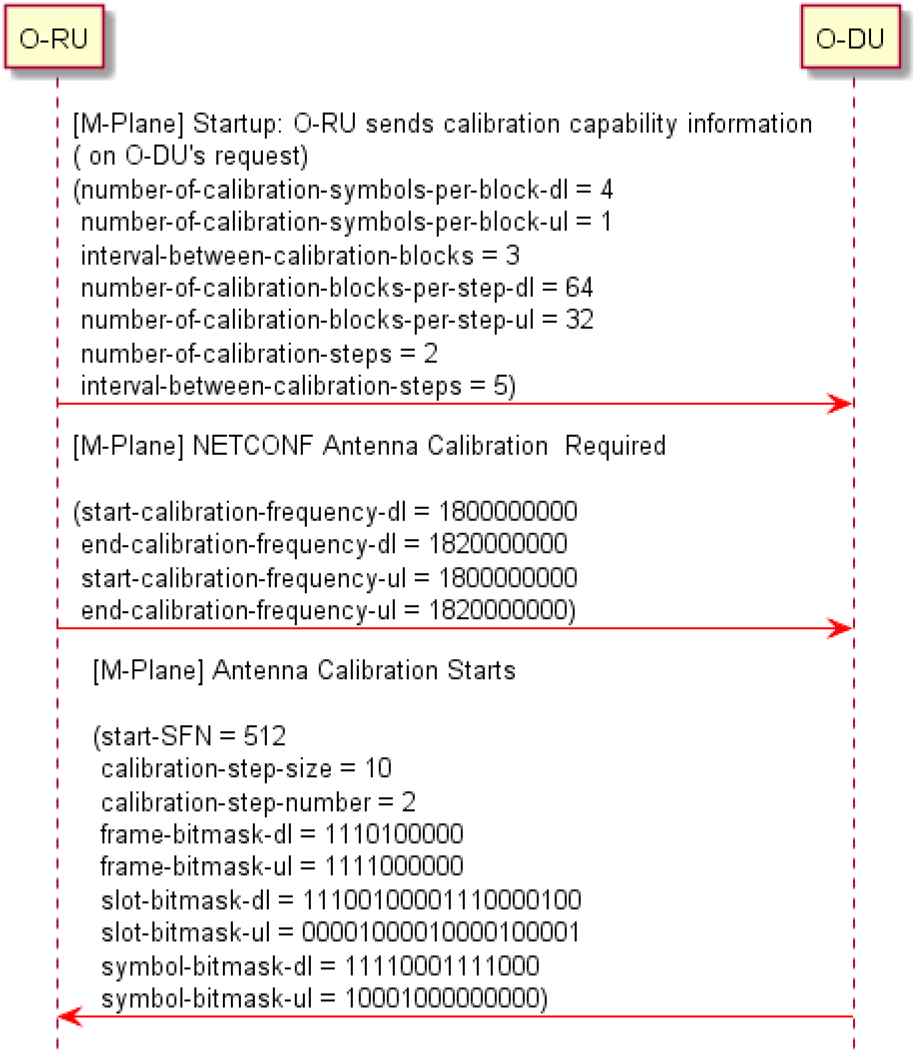


Figure 15.5.6-2: Example of message exchange

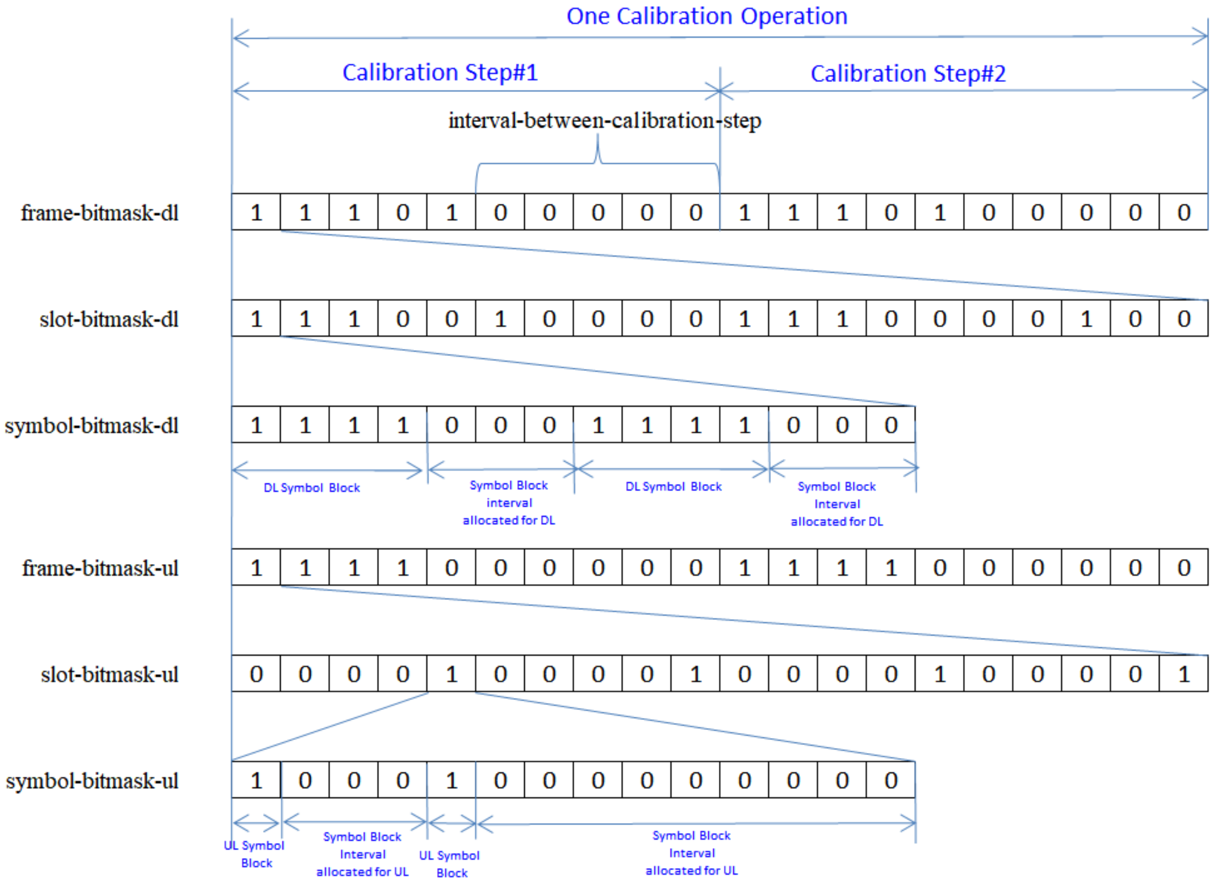


Figure 15.5.6-3: Time domain bitmask information from O-DU

15.5.7 Calibration with multiple timing resource sets

The O-RU may indicate its ability to support multiple time resource configuration sets for antenna calibration by support of the feature O-RU-COORDINATED-ANT-CAL-MULTIPLE-TIME-RESOURCE in o-ran-antenna-calibration.yang module. The feature is intended to extend the antenna calibration framework to support multiple time resources instead of single time resource supported. This capability can be used by the O-RU to specify unique calibration time resources for different calibration types. O-RU which supports this capability, exposes a list of time resources (antenna-calibration-multiple-time-resource-list). O-RU indicates desired set of time resources in 'antenna-calibration-multiple-time-resource' notification using specific value of 'antenna-calibration-time-resource-index' parameter.

This feature applies to O-RU initiated calibration where O-RU supporting this feature can use new notification 'antenna-calibration-multiple-time-resource' containing the parameter 'antenna-calibration-time-resource-index', defined in Sec 12.5.7. Based on the index value in the notification respective antenna calibration time resource values should apply while initiating "**start-antenna-calibration** RPC request". O-RU can use this calibration feature only in case O-DU configured parameter 'coordinated-calibration-multiple-time-resources-allowed' is set to TRUE.

NOTE: At any point in time only one calibration timing resources should be indicated the O-RU.

15.5.8 antenna-calibration-multiple-time-resource-params notification parameters

If the O-RU supports O-RU-COORDINATED-ANT-CAL-MUL-TIMING-RES-CONFIG, to initiate the calibration operation, the O-RU notifies the notification subscriber (O-DU) using the **antenna-calibration-multiple-time-resource-params** notification message. Remaining parameter list of this notification includes the frequency resources required for calibration same as 'antenna-calibration-required' described in clause 15.5.3:

- **antenna-calibration-time-resource-index:** key value to index the list 'antenna-calibration-variable-time-resource-list' based on the calibration duration required by the O-RU.

15.6 Static configuration for PRACH and SRS

15.6.1 Background

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 1: 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 2: 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.

15.6.2 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-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.

15.6.3 Frequency domain configuration

The meaning of frequency-related parameters is illustrated using Figure 15.6.3-1.

NOTE: Parameter offset-to-absolute-frequency-center belongs to low-level-rx-endpoint.

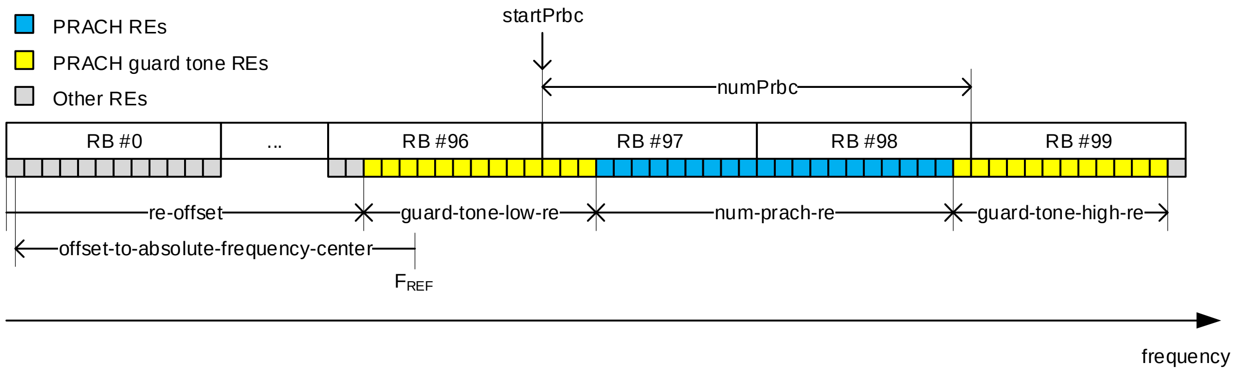


Figure 15.6.3-1: Relation between frequency-related parameters of the PRACH occasion

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], clause 7.5.3.2.

15.6.4 Time domain configuration

Meaning of parameters is illustrated using Figure 15.6.4-1.

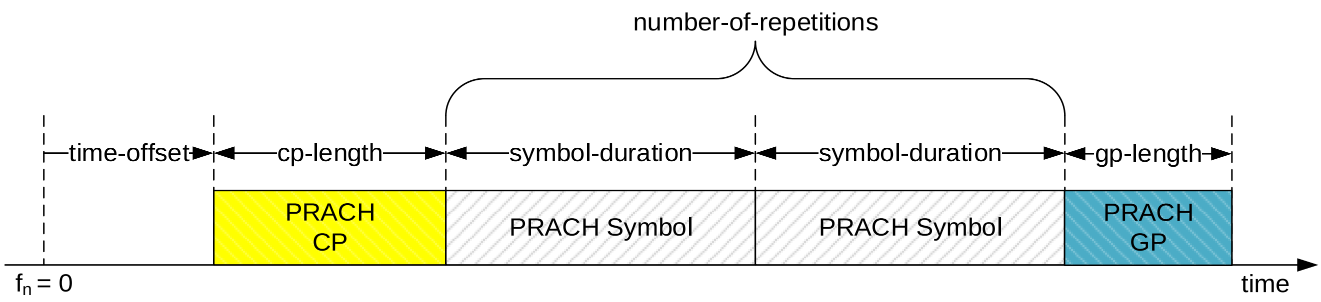


Figure 15.6.4-1: Timing-related parameters of single PRACH occasion

Figure 15.6.4-1 shows a single PRACH occasion containing 2 PRACH Symbols.

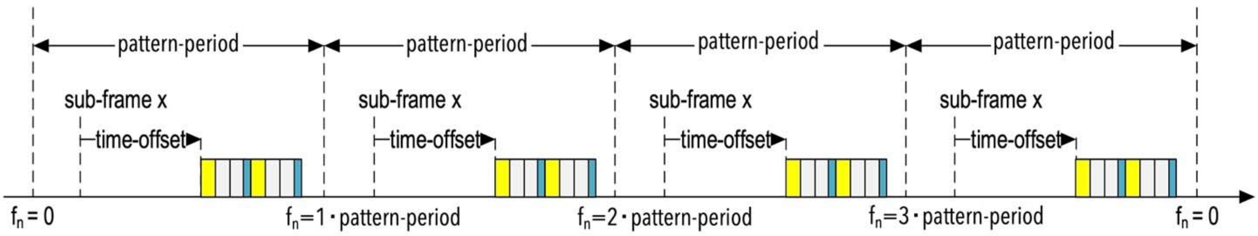


Figure 15.6.4-2: Timing-related parameters of one PRACH pattern

Figure 15.6.4-2 shows a single PRACH pattern containing two occasions of 2 PRACH Symbols (reuse of occasion shown on figure for single PRACH occasion for simplified view).

The corresponding parameters for above diagram are: ("number-of-prach-occasions" = 2, "number-of-repetitions" = 2)

NOTE 1: **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.

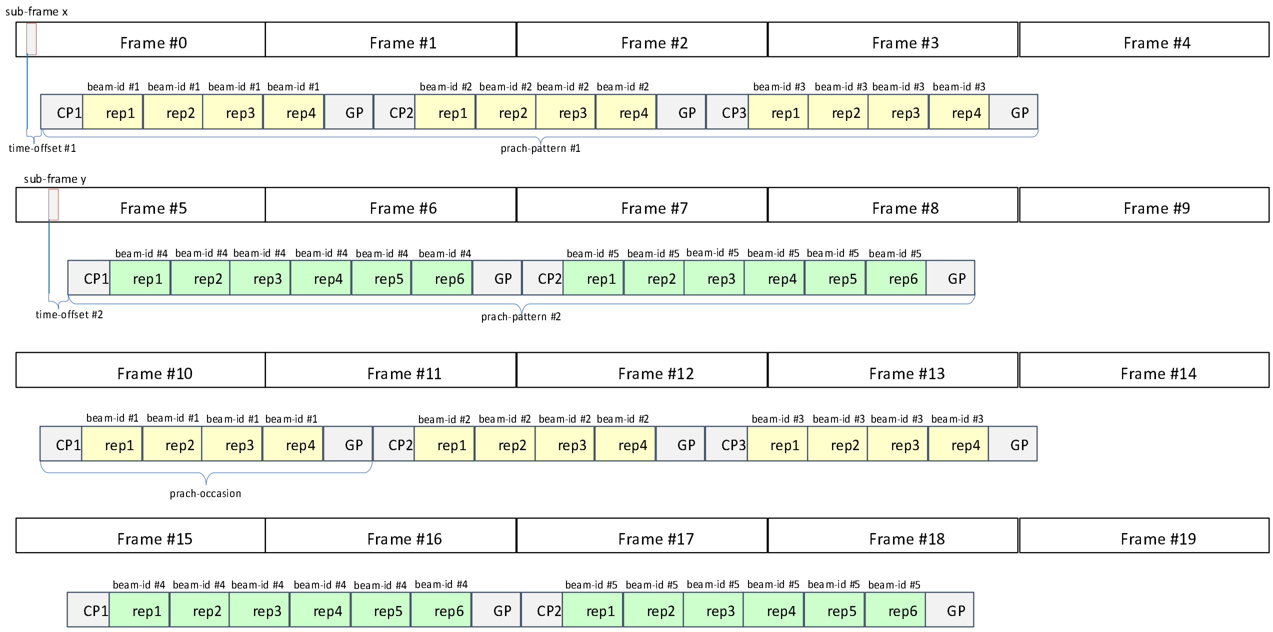


Figure 15.6.4-3: Example PRACH configuration formed of two PRACH patterns having different number of PRACH Symbols

NOTE 2: PRACH occasions are expanded for visual clarity.

NOTE 3: The Figure 15.6.4-3 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

For the PRACH configuration shown on in the figure, the parameter **pattern-period** = 10 as this is the number of frames after which PRACH pattern repeats.

NOTE 4: 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.

15.6.5 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.

Offset to the start of pattern periods can be configured either in low-level-rx-endpoint or in static-prach-configuration with sfn-number parameter. If the value is provided in static-prach-configuration, then it takes the precedence over the value configured in low-level-rx-endpoint.

Once the RF signal corresponding to the PRACH Symbol 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 Symbol 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 Symbols 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.

15.6.6 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 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 can 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.

15.6.7 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,

15.7 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-[tr]x-endpoints can be used (through low-level-[tr]x-endpoints) by [tr]x-array-carriers having configurable TDD pattern assigned.

Configured TDD pattern shall not be violated by C-Plane and U-Plane messages. 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 shall 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** shall 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 3: 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 shall 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 4: 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

15.8 C-Plane message limits

The O-RU exposes its ability to support C-Plane message limits by including support of the feature **CPLANE-MESSAGE-PROCESSING-LIMITS** in o-ran-wg4-features.yang module. Refer to clause 7.8.2.2 of CUS-Plane specification to understand details of this feature. The presence of this feature means, that in addition to an O-RU's "per-endpoint processing limits" e.g., endpoint-section-capacity, endpoint-beam-capacity, endpoint-prb-capacity, an O-RU may also have "per-cplane message limits"**.** To support this feature on a per endpoint basis, a new flag **'cplane-message-processing-limits-required'** is added to **'endpoint-types'** to indicate an endpoint's requirement to support C-Plane message processing limits. The flag shall be set to **true** for the endpoints to which C-Plane message processing limits apply. An additional configuration flag **'cplane-message-processing-limits-enabled'** is added to low-level-[tr]x-endpoints (applicable only for endpoints to which C-Plane limit requirement is exposed by O-RU) to enable the O-DU to use this feature on a per endpoint basis.

1) If the O-DU supports C-Plane message processing limits, it can choose to indicate it adheres to the limits by configuring schema node 'cplane-message-processing-limits-enabled= true'. In such case, the O-DU shall follow limits specified by the parameters, e.g., 'max-beams-per-slot-with-cplane-limits' and 'max-highest-priority-sections-per-slot-with-cplane-limits' when forming C-Plane messages.
2) If O-DU does not support C-Plane message processing limits by configuring schema node 'cplane-message-processing-limits-enabled= false' OR O-RU does not indicate it supports the **CPLANE-MESSAGE-PROCESSING-LIMITS** YANG feature, no C-Plane message limits shall apply and O-RU continues to use endpoint capacity limits specified in existing endpoint by per-endpoint limits e.g., endpoint-section-capacity, endpoint-beam-capacity, endpoint-prb-capacity.

15.9 Advanced endpoint capability report

Both **endpoint-types** and **endpoint-capacity-sharing-groups** provide an optional method for advanced endpoint capability and capacity reporting by using **supported-configuration-combinations**. The report is provided through

supported-configuration-combinations which allows the O-RU to convey its supported configurations of endpoint UL/DL signal transmission/reception and processing as a function of different combinations. In short, the O-RU may report a different processing capacity dependent on a specified set of configurations. This clause describes the data structure of **supported-configuration-combinations** and provides examples of its usage.

Figure 15.9-1 depicts the data structure in a condensed format. As shown **supported-configuration-conbinations** is a three-level list consisting of the levels: **supported-configuration-combinations**, **set**, **config**. The parameter **supported-configuration-combinations** will henceforth be referred to as **combination** for the sake of brevity.

An entry of **config** provides a specific configuration with several parameters to represent the processing capacity of one or more internal components of the O-RU when these are configured to a certain state via endpoint mapping. The parameters **carrier-types** and **center-from-freqoffset** may be omitted to indicate that their value does not affect the O-RU's resource usage, i.e., a wildcard entry.

An entry of **set** provides a list of one or more **config** entries. The value of **max-overlapping-instances** indicates the number of **config** entries which may be activated simultaneously in time, including multiple activations of the same **config** entry within a **set**. When **max-overlapping-instances** is 1 the O-DU can only utilize one **config** entry from the **set**. Simultaneously overlapping is defined by the DL or UL signals overlapping in time when transmitted or received by the O-RU at the antenna connector. The **config** entries are indirectly activated when the O-RU receives C-plane messages which can be processed by the capabilities reported in those entries.

NOTE: When Section Type 3 C-plane messages are received the O-RU may rely on fields such as filterIndex, timeOffset, frameStructure and freqOffset to map to internal processing resources.

An entry of **combination** provides a list of one or more **sets** from which one or more **config** entries (dependent on the value of **max-overlapping-instances** of the specific **set**) may be activated at an overlapping moment in time. At any overlapping moment in time only a single combination entry shall be applied to the O-RU. I.e., **config** entries from different **combination** entries shall not be activated simultaneously. The **combination** entry which the O-DU intents to utilize shall be conveyed to the O-RU via **combination-configuration** as part of the general carrier configuration procedure. Optionally the O-DU may convey its intent to utilize a subset of **sets** and **configs** of the selected **combination** via the same parameter. If the list of configurations is not populated by the O-DU, the O-RU shall assume that the **combination** entry will be utilized to the greatest extent possible.

In all cases the activation of a particular configuration is performed indirectly by a combination of M-plane configuration and received C-plane messages as they are mapped to the internal processing resources of the O-RU. In cases where conflicts arise between the capabilities of the endpoint conveyed via **supported-configuration-combinations** and other endpoint level capabilities it is the most restrictive capacity restriction option that shall apply. In other words, all applicable restrictions shall be met.

Once the supported combinations are advertised by the O-RU, the O-DU is expected to comply with a given combination capability advertised by the O-RU. However, if the O-DU for any reason violates the O-RU advertised capabilities, the O-RU shall send an alarm notification, e.g., 'fault-id = 31' to subscribers following the procedures defined in clause 11 and Table A.1-1.

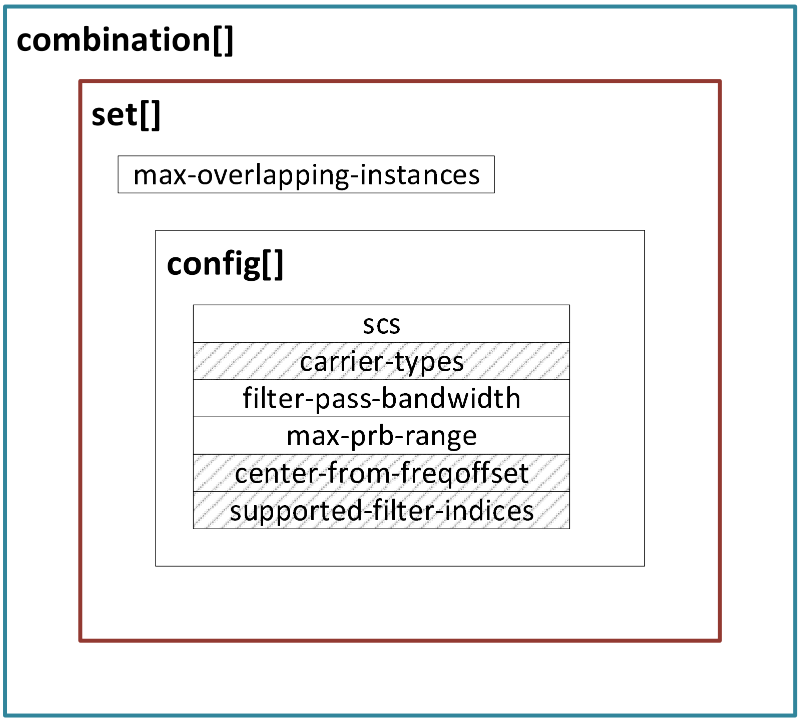


Figure 15.9-1: Supported configuration combinations group list overview

The **config** listed parameter **center-from-freqoffset** indicates whether deriving a new carrier center from a C-plane section type 3 message (as defined in [2] clause 15.4.1) is expected to be enabled for the activate configuration. It shall be enabled via the endpoint level parameter **center-from-freqoffset-enabled**. If an endpoint supports this capability, it shall report so by setting **center-from-freqoffset-supported** to TRUE.

The **config** listed parameter **filter-pass-bandwidth** conveys the maximum size of the passband filter relative to the host carrier center, as defined by **center-of-channel-bandwidth**, and therefore defines the frequency range from which PRBs can be extracted with the endpoint. In turn, the O-DU configures the expected occupied bandwidth to the O-RU via **occupied-bandwidth**, e.g., refer to [2] clause 15.4.1, 15.4.2 and 15.4.3. A list of supported filter pass bandwidths is provided in **supported-filter-pass-bandwidths.**

The **config** leaf-list parameter **supported-filter-indices** conveys the filter indices supported or intended by the **config**. As an example, the O-RU may use the parameter to restrict an endpoint to only support NB-IoT or a specific NB-IoT channel such as NPUSCH.

A staticRX endpoint intended to support a NB-IoT carrier with an NPUSCH and two NPRACH channels may be conveyed in a single **combination** entry, as illustrated in Figure 15.9-2.

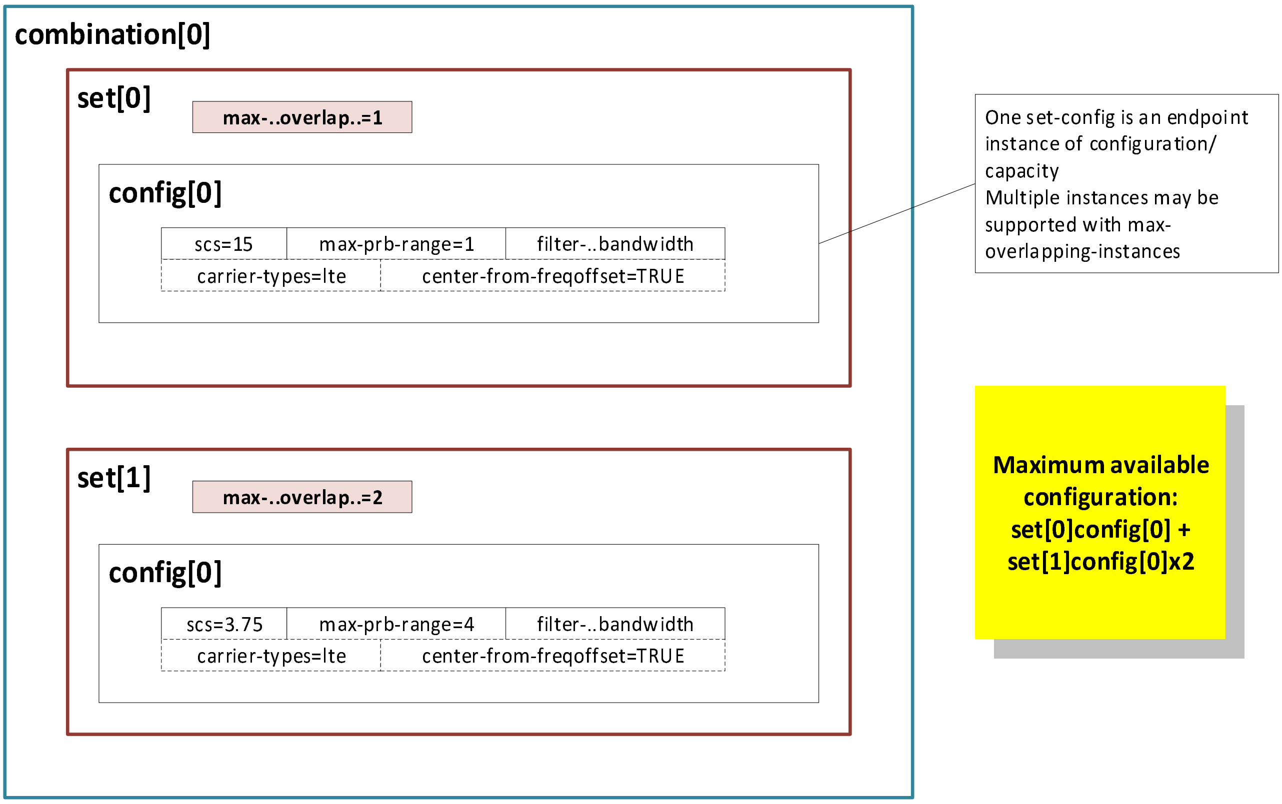


Figure 15.9-2: Example a static RX endpoint indirectly indicating support for a single NB-IoT carrier support with one NPUSCH and two NPRACH channels

A static RX endpoint may additionally support an LTE carrier and PRACH, however, not at the same time as an NB-IoT carrier. In which case, an additional **combination** entry may be listed to convey support for either NB-IoT or an LTE carrier as shown in Figure 15.9-3.

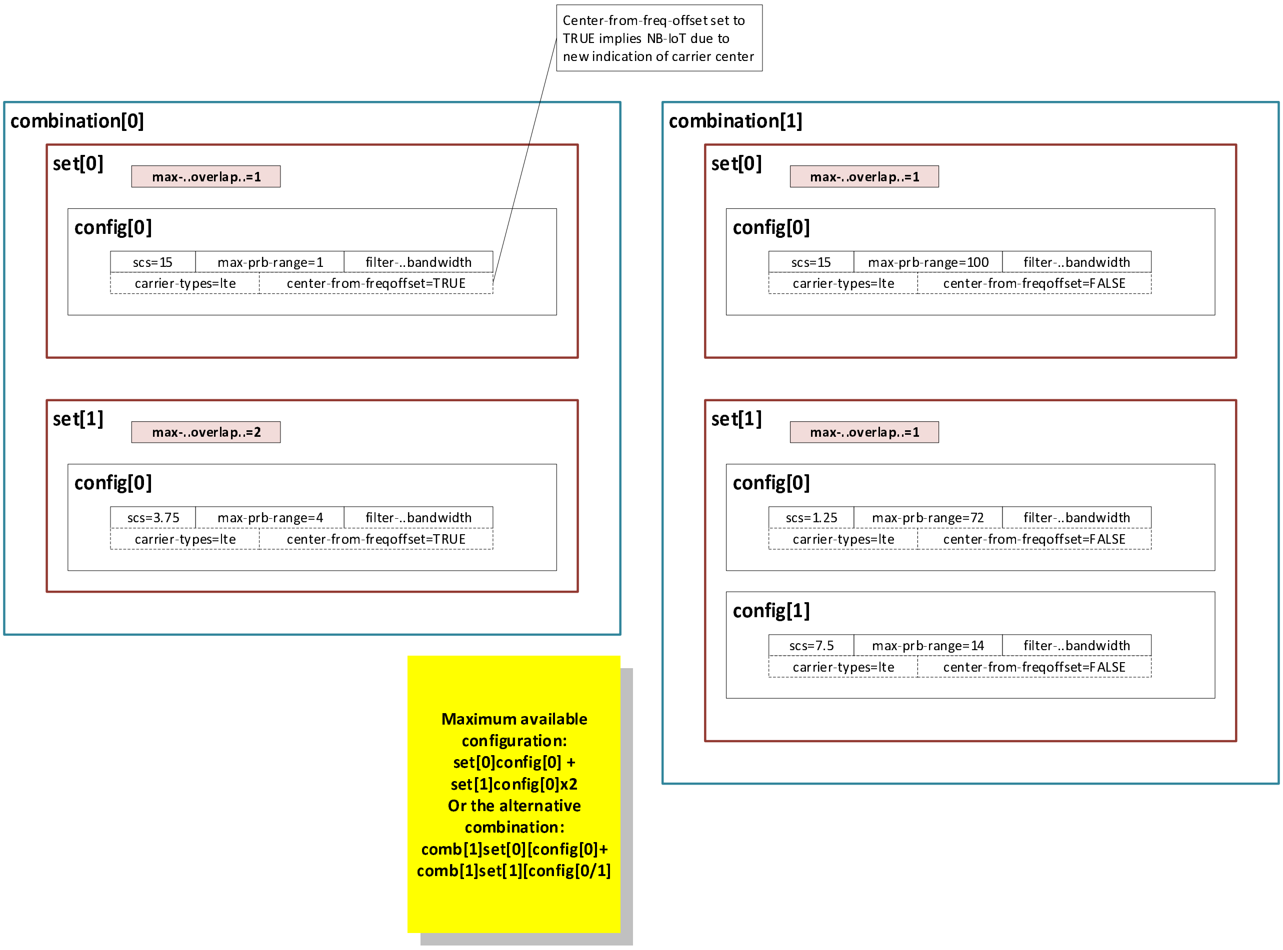


Figure 15.9-3: Example A staticRX endpoint indirectly indicating support for either a NB-IoT and LTE carrier, however, not both at the same time

Finally, an endpoint capability report may be structured to indicate a large degree of flexibility in terms of active **configs** as illustrated in **combination**-*1* of Figure 15.9-4 which shows any three combinations of the listed **configs** may be utilized. Whereas **combination**-*0* can only support a single instance of a 15 kHz channel.

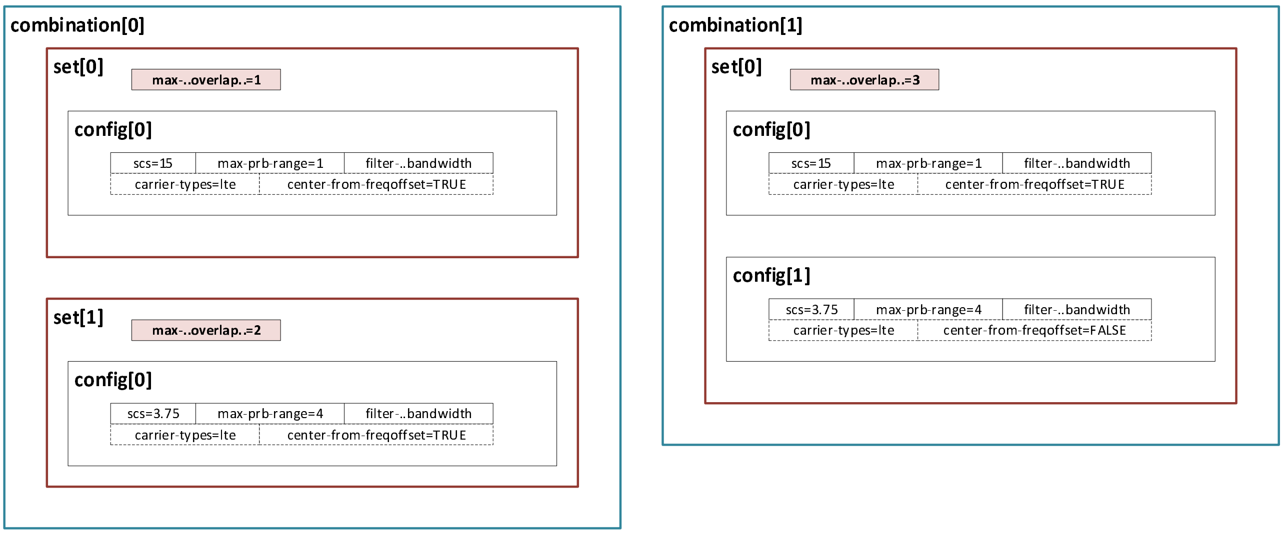


Figure 15.9-4: Example A static RX endpoint indirectly indicating support for a NB-IoT carrier with more flexibility in the right-hand side combination

15.10 U-Plane message limits

O-RU may indicate that its endpoints have U-Plane message processing limits as specified in clause 8.5.1 of CUS-Plane specification [2], by including feature **UPLANE-MESSAGE-PROCESSING-LIMITS** in o-ran-wg4-features.yang. The presence of this feature means, an O-RU may expose data node **uplane-message-processing-limits-required** per **static-low-level-tx-endpoint** via **endpoint-types**. **uplane-message-processing-limits-required** indicates whether an endpoint has U-Plane message processing limits or not.

1) If O-DU supports feature **UPLANE-MESSAGE-PROCESSING-LIMITS** and supports the leaf **max-section-headers-per-uplane-message** , it should set **uplane-message-section-header-limit-enabled** to 'true' and should comply with O-RU limit specified by the parameters **max-section-headers-per-uplane-message** when forming U-Plane messages.

2) When the O-DU does not configure the leaf **uplane-message-section-header-limit-enabled** to 'true', O-RU shall process the U-Plane messages that exceed the limits even though the O-RU may operate with degraded performance or capacity, (refer to [2] clause 8.5.1 for more details).

3) When U-Plane message exceeds the limit reported by O-RU, O-RU may raise alarm to indicate the degraded performance or capacity. For details about alarm notification 'fault-id = 31', refer to Table A.1-1.

16 Licensed assisted access

16.1 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 clause is focused on the LAA-related messages and procedures needed in the M-plane. The C-plane related messages are defined in clause 7.2.5.2 of 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:

1) 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 as specified in RFC 8525 [59].

   - 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 6.1.1 in clause 6, 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 6.1.1.

2) Carrier-selection: Selecting the best channel in the unlicensed band, both initially and dynamically over time, as illustrated in Figure 16.1-1.

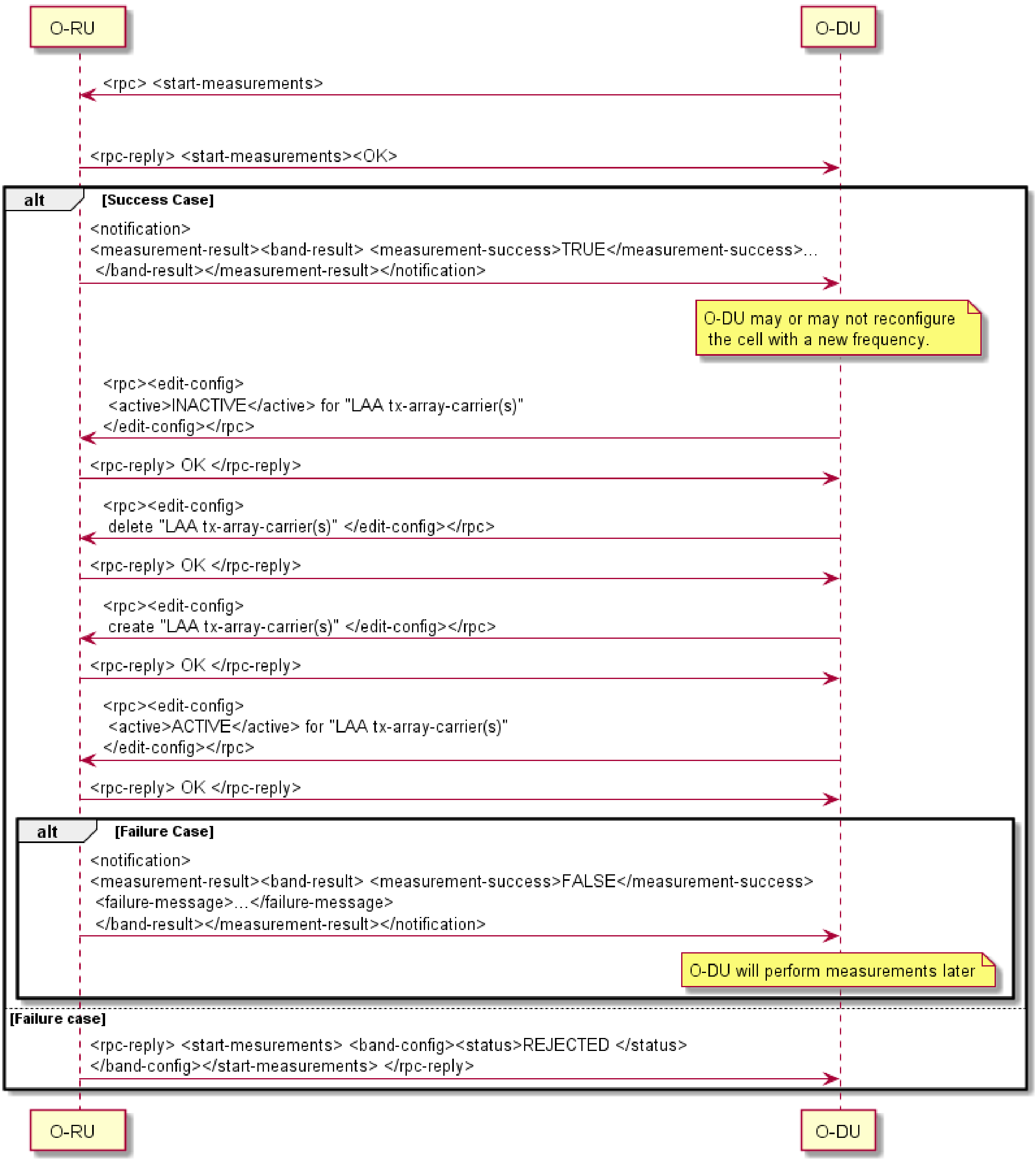


Figure 16.1-1: Carrier-Selection Call-flow

16.2 LAA-initiation process

16.2.1 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.

16.2.2 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) as specified in clause 15.1.5 of 3GPP TS 36.213 [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.

3) **tx-antenna-ports** (uint8): This value indicates the Tx antenna ports for DRS.

4) **transmission-power-for-drs** (int8): This value indicates the offset of CRS power to reference signal power (dB).

5) **dmtc-period** (enumeration): This value indicates DMTC period in milliseconds.

6) **dmtc-offset** (uint8): This value indicates DMTC offset in Subframes.

7) **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, as specified in clause 15.1.3 of 3GPP TS 36.213 [32], is represented as K. Based on the 3GPP specification, this value is selected by O-RU from the set of values {1, 2, …, 8}

16.3 Carrier selection

16.3.1 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-uplane-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.

16.3.2 LAA carrier frequency configuration

After receiving the measurements, the NETCONF client configures the O-RU with the new channel(s), if needed. To start radio transmission and reception with a new centre frequency, for every component carrier (CC) that needs to be re-configured with a new centre frequency, the O-DU shall first deactivate the TX carrier, delete it, then create a new TX carrier (using the new centre 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 clause 15.3: Carrier Configuration, in the M-plane specification and elaborated in Figure 15.2.4.1.

NOTE 1: 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 2: The 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.

17 Shared cell

17.1 Introduction

This cluse specifies the support of the "Shared Cell" O-RU use case. The features of C/U-Plane aspects are described in clause 13 of [2]. The M-Plane aspects necessary to support Shared Cell are described in this clause 17.

17.2 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, as illustrated in Figure 17.2-1.

In Figure 17.2-1, 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 clause 5.1.2. New functionality which is required to be added, together with existing functionality which is required to be enhanced are specified in clauses 17.3 to 17.6. There are no changes to the functionality described in the following clauses and their associated YANG models:

- Clause 8 : Software Management
- Clause 11 : Fault Management
- Clause 12 : File Management

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 does 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.

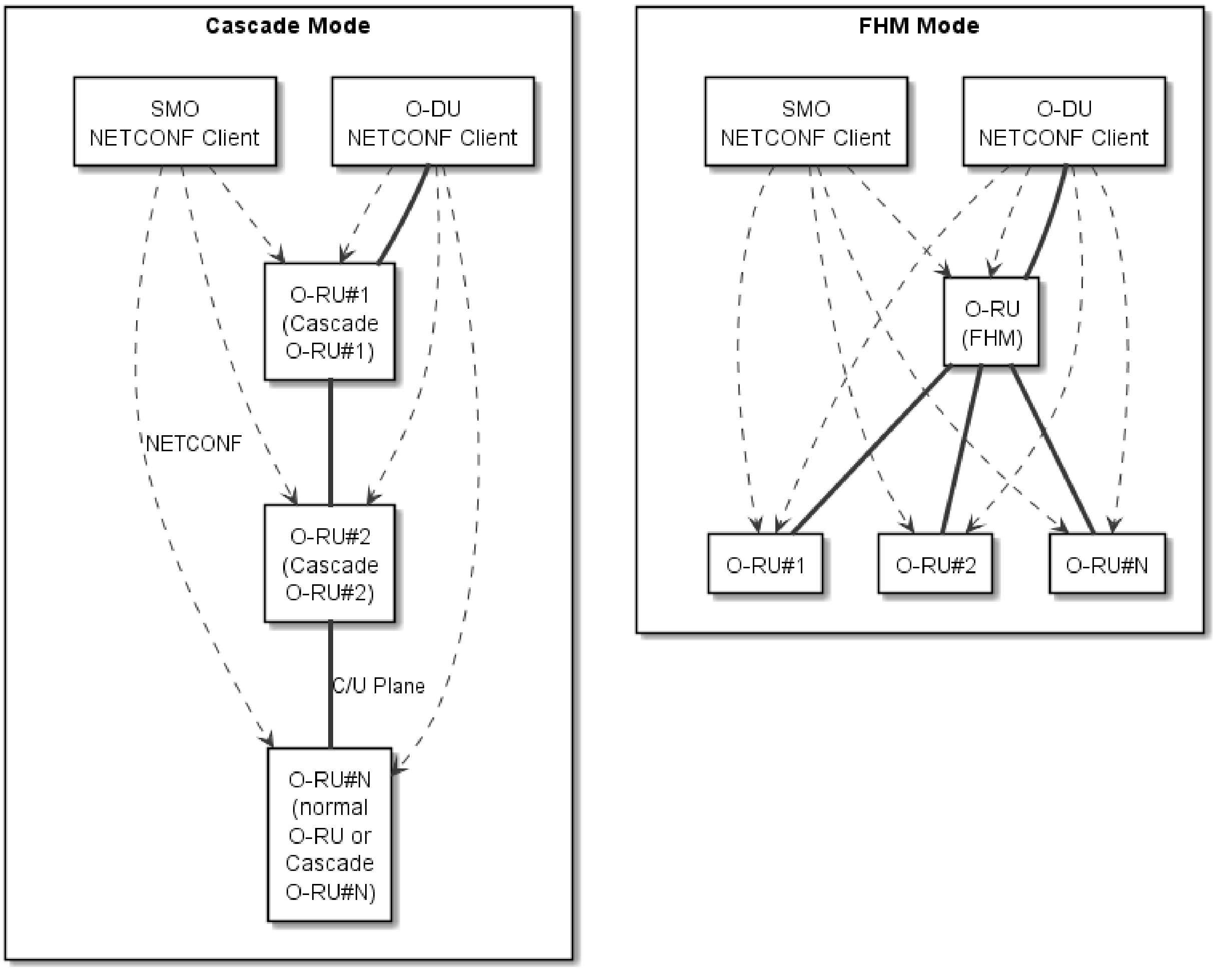


Figure 17.2-1: M-Plane Connection

17.3 Start-up and Installation

This clause 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".

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 Figure 6.1.1 on clause 6 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.

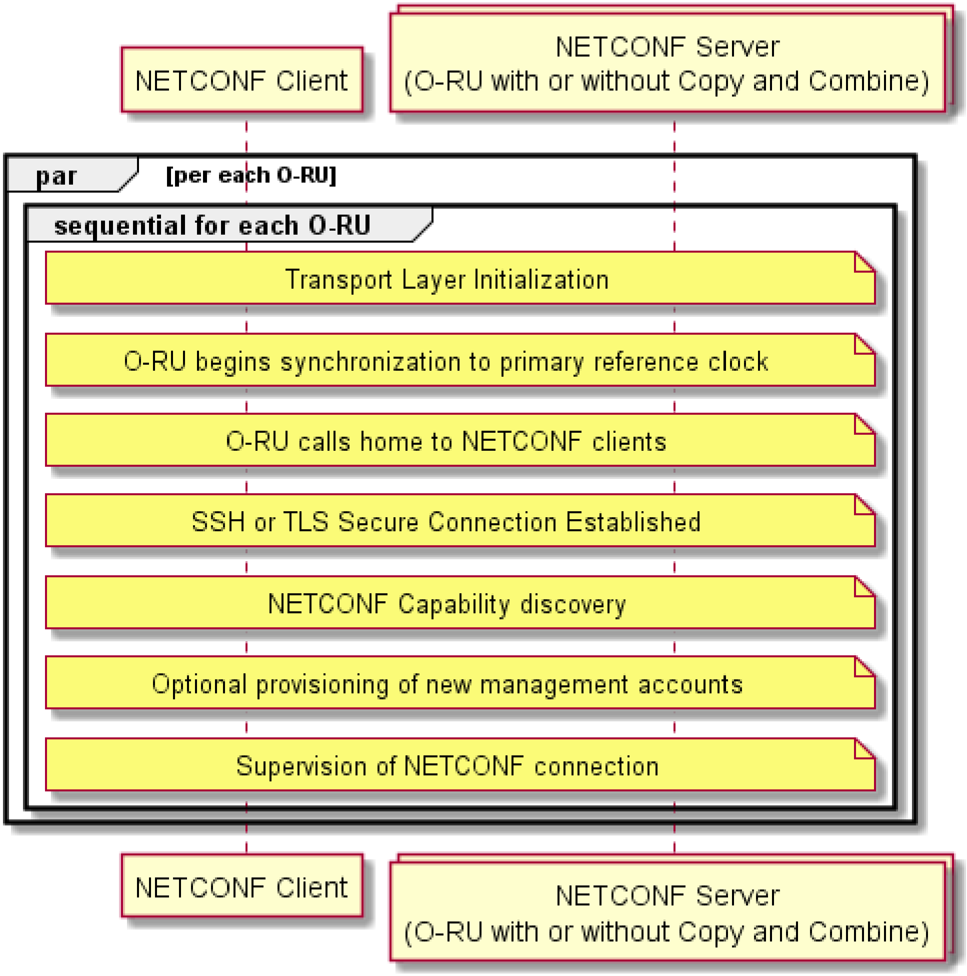


Figure 17.3-1: NETCONF establishment of the start-up

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 clause 17.6.1.

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 clause 17.6.3)

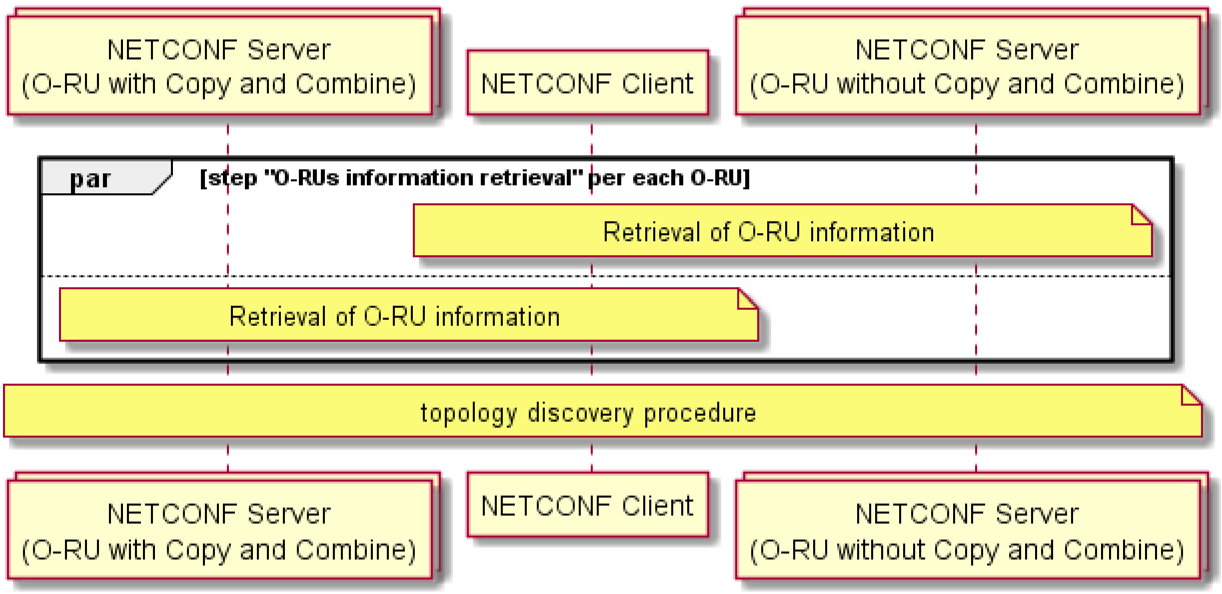


Figure 17.3-2: RU information retrieval and topology detection at the start-up

The NETCONF client performs software management per O-RU as described in clause 8.

As described in clause 17.2, 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.

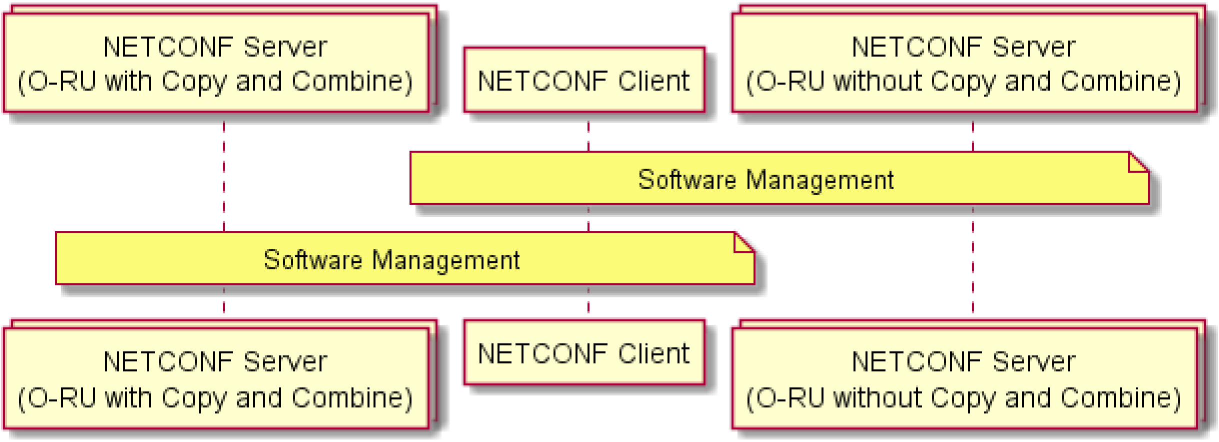


Figure 17.3-3: Software management per O-RU

After the software management steps are completed for all NETCONF servers, the NETCONF client performs shared cell configuration. (See clause 17.6.3)

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 clause 17.6.2.

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, **shared-cell-copy-uplane-config** and **shared-cell-combine-uplane-config** in o-ran-shared-cell.yang module are used. (See clause 17.6.4)

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)

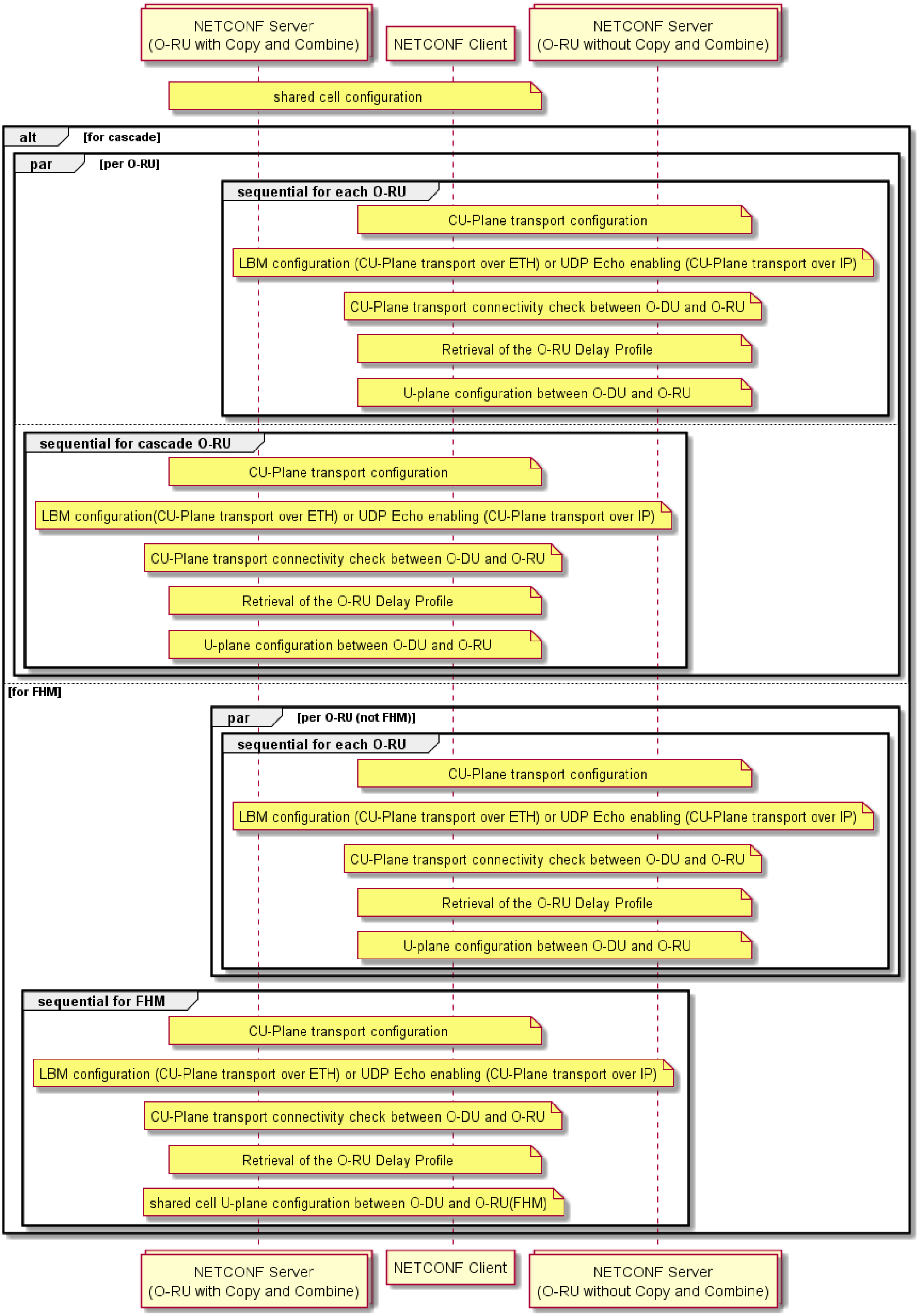


Figure 17.3-4: shared cell configuration and u-plane configuration

The NETCONF client performs further steps for the regular start-up procedure as described in clause 6.

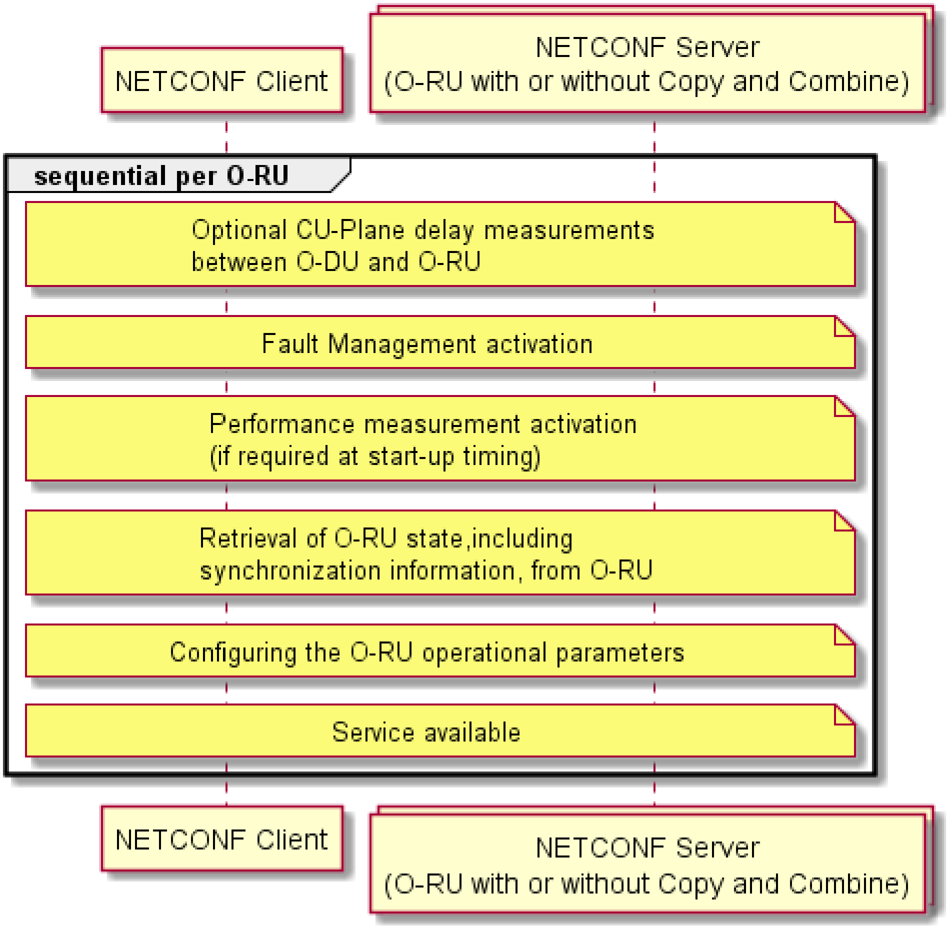


Figure 17.3-5: further steps for start-up procedure

17.4 Performance management

This clause provides description of the specific part of Performance Management for shared cell.

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 clause 17.6.1 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 1: 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 2: 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.

Epe-stats:

O-RU (Cascade / FHM) shall monitor **epe-stats** per hardware component.

Symbol-rssi-stats:

O-RU may monitor **symbol-rssi-stats** per rx-array-carrier.

NOTE 3: O-RU without radio transmission/reception capability (FHM) does not support monitoring of **symbol-rssi-stats**

Table 17.4-1: Measurement-group of O-RU (Cascade / FHM)

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
symbol-rssi-stats-object	rx-array-carrier

For more detail, please refer to Table B.2-1 Counters definition in Annex B.

17.5 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 **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 clause 13.4.3 of [2].

The **ta3-prime-max** schema node is relevant to the timing on each symbol. The **scs** schema node indicates each symbol timing. The capability of supported SCSes is reported by leaf-list **scs-supported**. Among them the intended SCSes should be informed to FHM/Cascade O-RU as **scs** schema node, associating with eaxc-id. SCS in an eaxc-id can be different from that of another eaxc-id. SCS in an eaxc-id can be also plural. FHM/Cascade O-RU reports the capability of supporting multiple SCS in a single eaxc-id as the feature MULTIPLE-SCS-IN-EAXC, and O-DU should check before configuring multiple SCS in a single eaxc-id.

- **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.

Clause 13.5.3 of the CUS specification [2] introduces new parameters for controlling combine operations:

- **t-combine-net:** is the read-only processing delay as the reported by the FHM/Cascade O-RU
- **tx-duration:** is a configurable parameter by O-DU or a calculated parameter by FHM/Cascade O-RU, corresponding to the message transmission duration

where

the sum of **t-combine-net** processing delay and the maximum **tx-duration** transmission duration corresponds to the **t-combine** duration.

NOTE: CUS Plane [2] refers to these parameters as T_Combine_net, Tx_Duration and T_Comb.

An FHM/Cascade O-RU can signal to an O-RU Controller that it supports these parameters by indicating it supports the ENHANCED-T-COMBINE YANG feature. The O-DU configures such an FHM/Cascade O-RU to use the new parameters by setting the **enhanced-t-combine-enabled** to 'true'. In this case, the FHM/Cascade O-RU shall use **t-combine-net** and **tx-duration** to calculate the T_waiting (Refer to 13.5.3 of [2]) instead of **t-combine**. If **enhanced-t-combine-enabled** is set to 'true' and if no value of **tx-duration** is configured by the O-DU, the FHM/Cascade O-RU shall calculate **tx-duration** from **shared-cell-combine-entities**, ensuring that the maximum amount of calculation doesn't exceed the symbol duration without cyclic prefix designated by **ta3-prime-max**.

17.6 Details of O-RU operations for shared cell

17.6.1 O-RU information for shared cell

This clause provides the function detail of O-RU operations for a 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 in Figure 17.6.1-1.

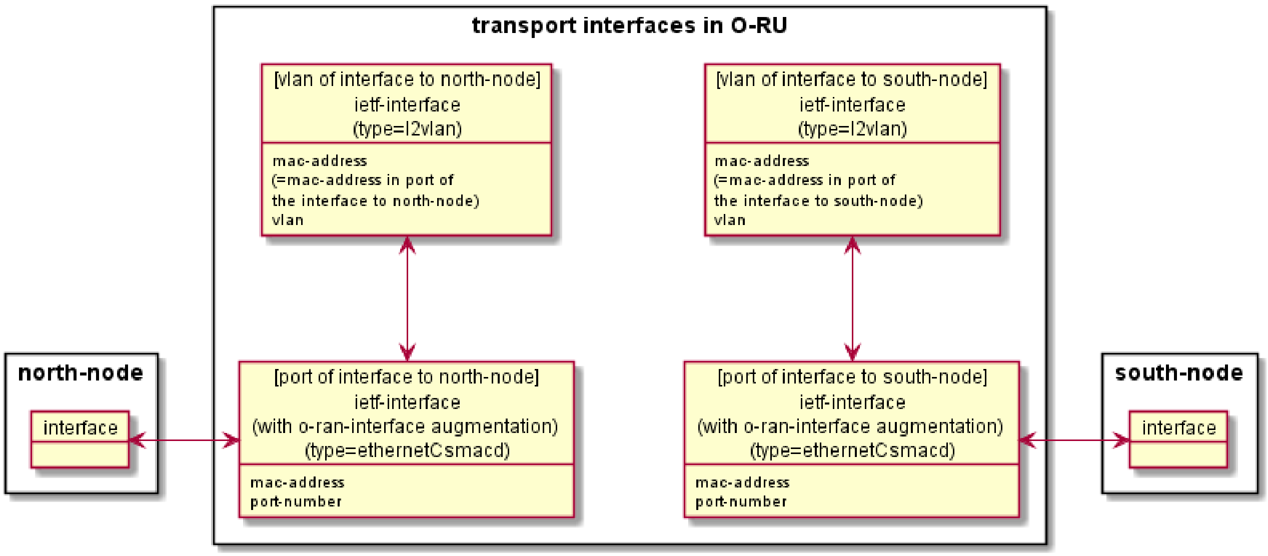


Figure 17.6.1-1: Transport Interfaces of both side of O-RU

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.

NOTE: The present document defines the use of IEEE 802.1X port based access control, as described in clause 7.12. In order to use IEEE 802.1X, a supplicant PAE in an O-RU needs to signal with an authenticator PAE. The present document does not define operation of an authenticator PAE in an O-RU, instead only defining operation of the supplicant PAE functionality. Operators wanting to benefit from IEEE 802.1X when operating a shared cell configuration need to consider which network element(s) is/are responsible for supporting the authenticator PAE function.

The role of interfaces shall be detected by topology discovery procedure described in clause 17.6.2.

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 clause 7.3.

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.

<u>Capability of O-RU with Copy and Combine function</u>

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. It contains **scs-supported** capability of FHM in the present document.

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-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.

<u>Yang modules for FHM mode</u>

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.

17.6.2 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 clause 7.6 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:

1) In a first stage, the O-DU performs Ethernet connectivity monitoring on all discovered O-RU MAC addresses, using the procedures defined in clause 7.6.1. 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 automatically learns the MAC addresses switched through these devices.

NOTE: In addition to Ethernet connectivity monitoring, DHCP discovery, call home and M-plane connection establishment can 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.

2) 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.

If the topology of the cascade architecture does not ensure that Ethernet frames sent between a south-node and a north-node are bridged by an O-RU, the O-DU needs information in addition to the O-RU FIB to perform topology discovery. How an O-DU becomes aware that additional information is required, together with the definition of such information, is outside the scope of the present document.

Figure 17.6.2-1 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.

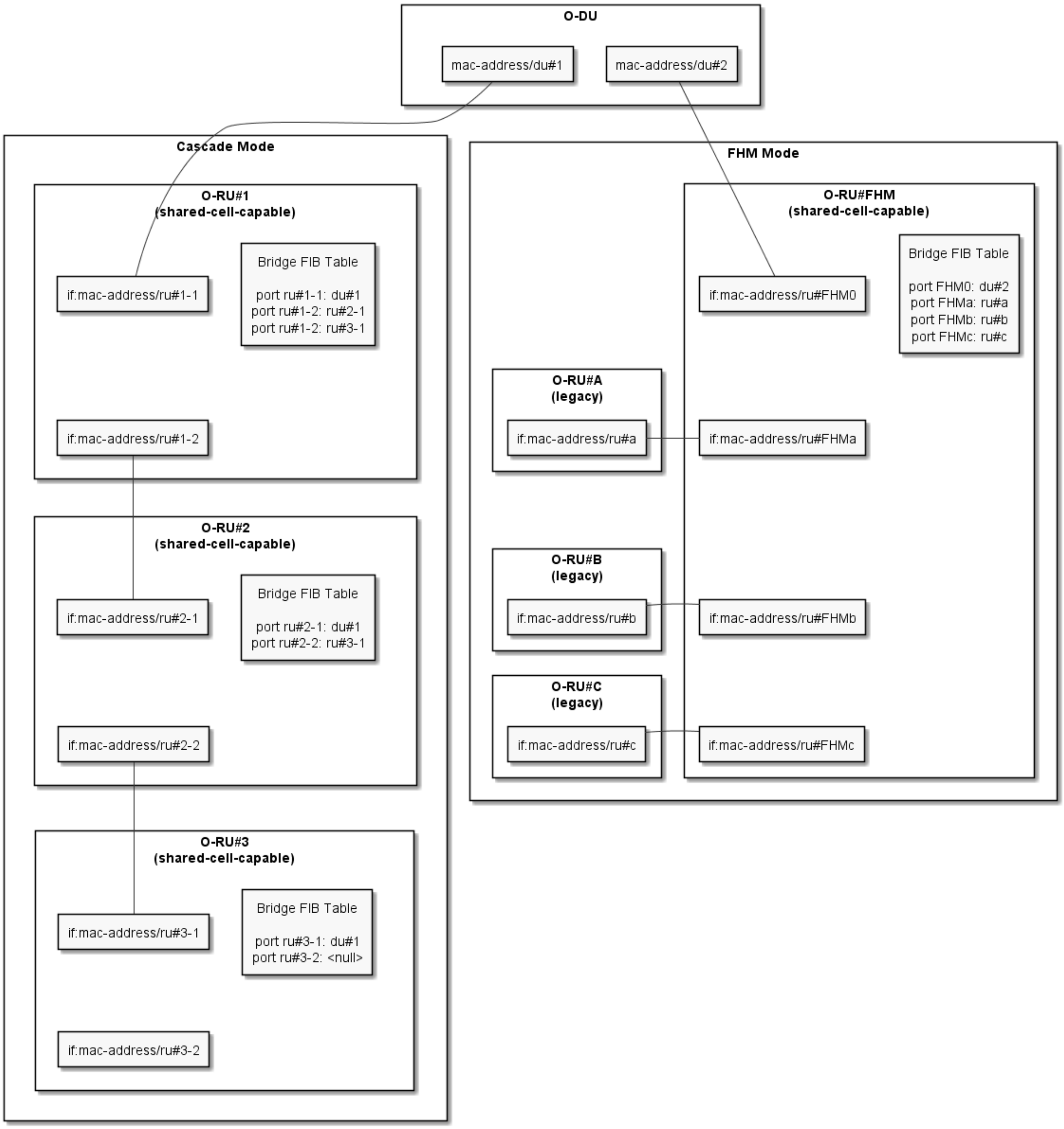


Figure 17.6.2-1: Bridge FIB table entries learned from LBM/LMR connectivity check procedures

17.6.3 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 clause 17.5.

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**.

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 1: 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 2: 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.

Figure 17.6.3-1 illustrates the shared cell configuration and transport configuration.

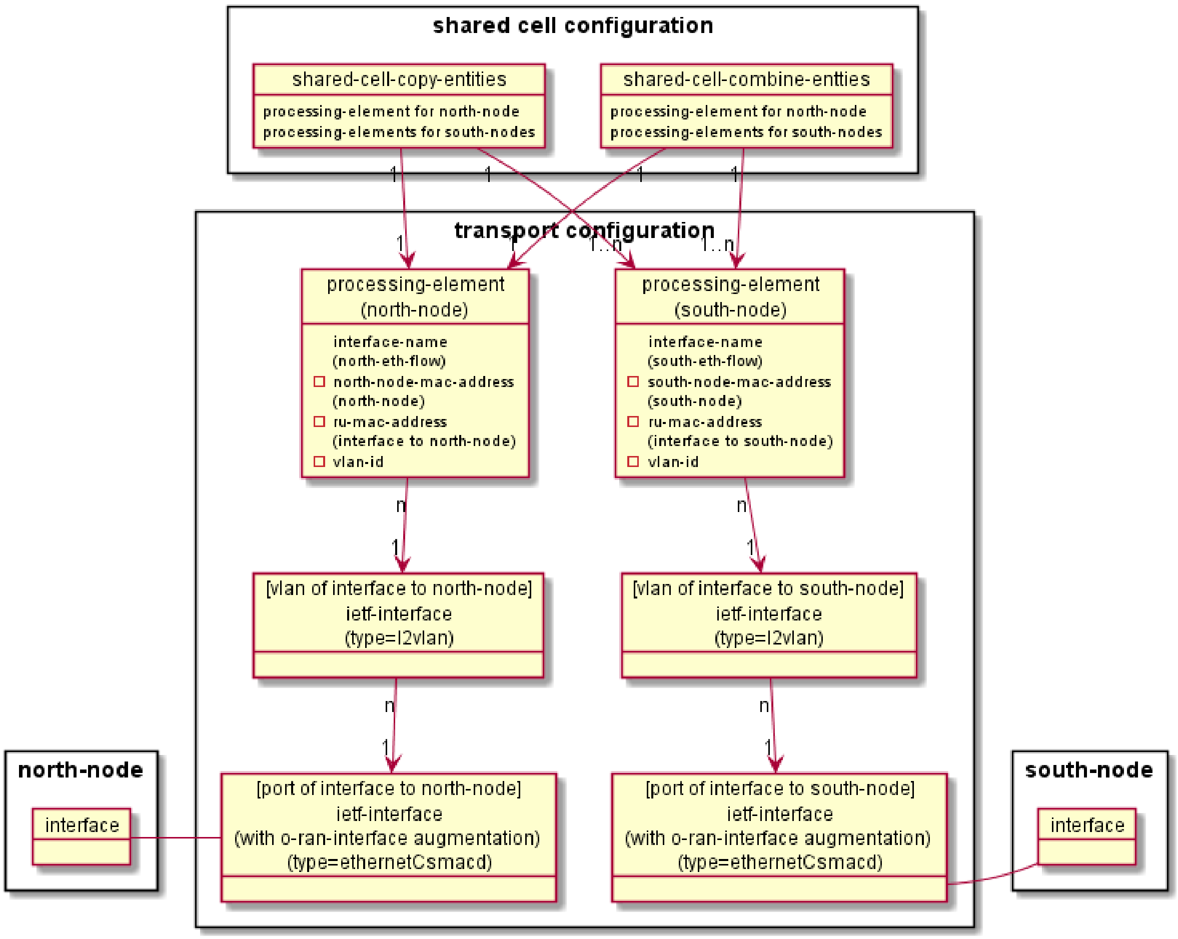


Figure 17.6.3-1: Relation of Shared Cell Copy and Combine Entities and Transport Configuration

For the cascade mode, the processing element for north-node shall be connected to the **low-level-tx(rx)-links** in u-plane configuration as in the Figure 15.2.4-1 in clause 15.2.4. In the O-RU (FHM), there is no radio transmission. Figure 17.6.3-2 illustrates the example of topology diagram and shared cell copy/combine entities. Each thick blue line indicates the **transport-flow** in the processing element.

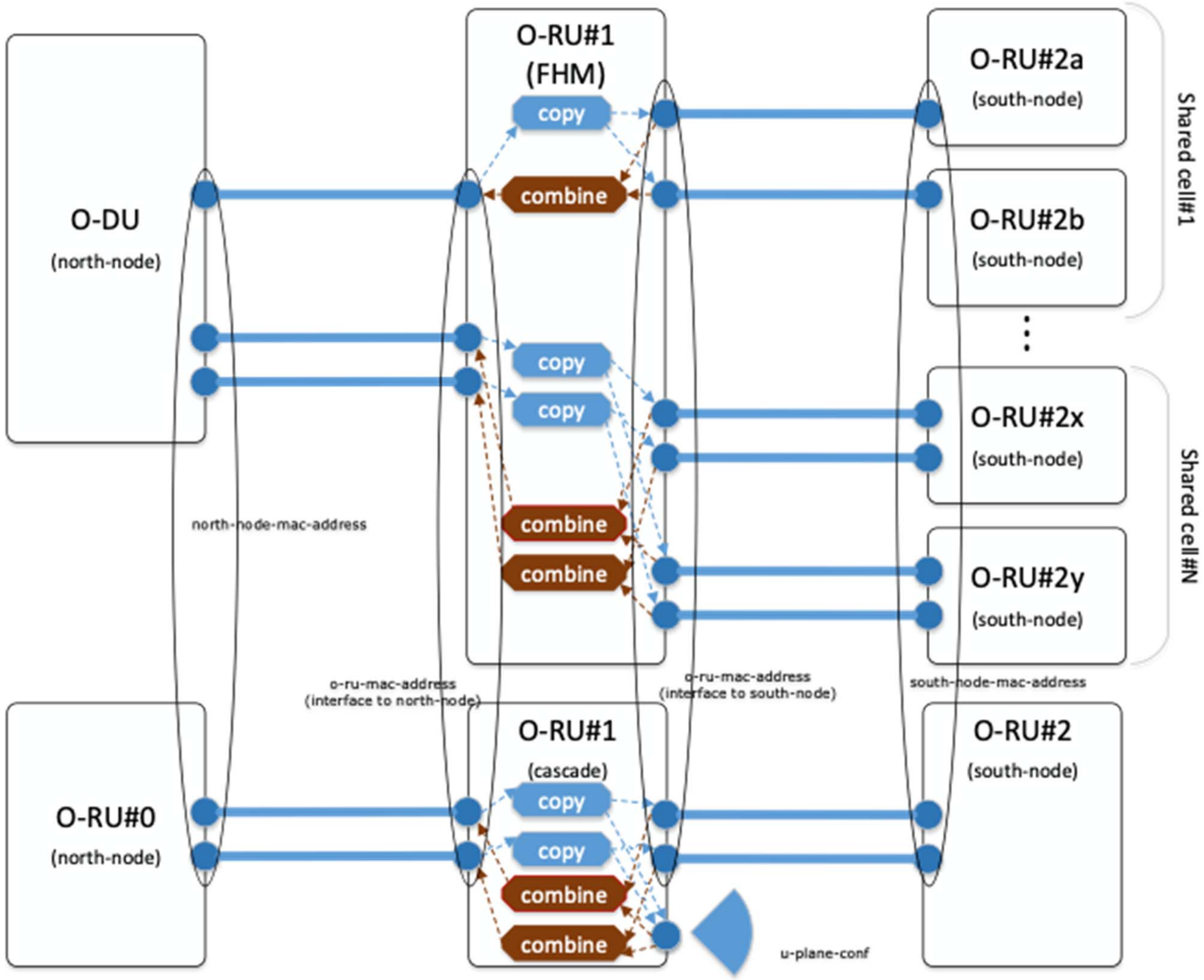


Figure 17.6.3-2: Example of Topology and Shared cell Copy/Combine entities

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 3: "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 the present document.

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 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.

17.6.4 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. The **shared-cell-combine-uplane-config** includes 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 cases that all PRBs in numPrbc are controlled by C-plane message or **tx-duration** is calculated by FHM.

FHM may have multiple sets of shared cell networks as described in Figure 17.6.3-2 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 1: **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 2: If the FHM supports the C/U-plane monitoring timer described in clause 7.10, 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 avoids 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.

17.6.5 Support of selective transmission and reception function

This clause describes M-Plane support for selective transmission and reception function which is specified in clause 13.3 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.

17.7 Cascade-FHM mode

17.7.1 Background

The introduction of Cascade-FHM mode is described in clause 13.7 of [2]. This clause describes mainly shared cell configuration on cascaded FHMs. For other common parts, like start-up procedures, topology discovery procedure etc., please refer to relevant parts in clause 17. 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.

17.7.2 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 the Figure 17.7.2-1 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. In this scenario a single SCS per FHM is assumed. When multiple SCS are used in FHM, shared-cell-combine-entities are increased accordingly.

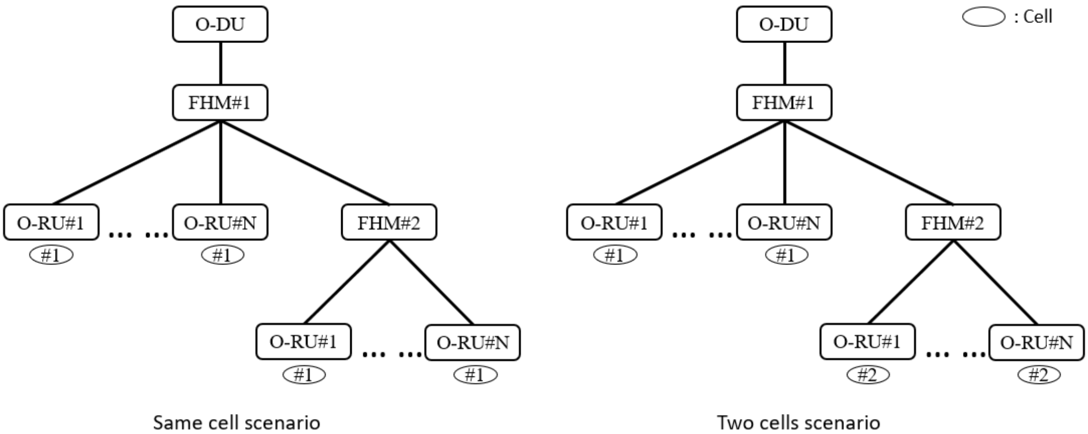


Figure 17.7.2-1: Typical cell scenarios in Cascade-FHM mode

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.

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.

18 Configured subscriptions

18.1 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 as specified in RFC 8639 [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. An O-RU may store configured subscription information in its reset-persistent memory.

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.

18.2 Description

An O-RU controller can discover the event-streams supported by an O-RU. The O-RU Controller may then establish a configured subscription to a particular event-stream. The same NACM privileges defined in clause 6.5 shall 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.

18.3 Procedure

The overall procedure is illustrated in Figure 18.3-1.

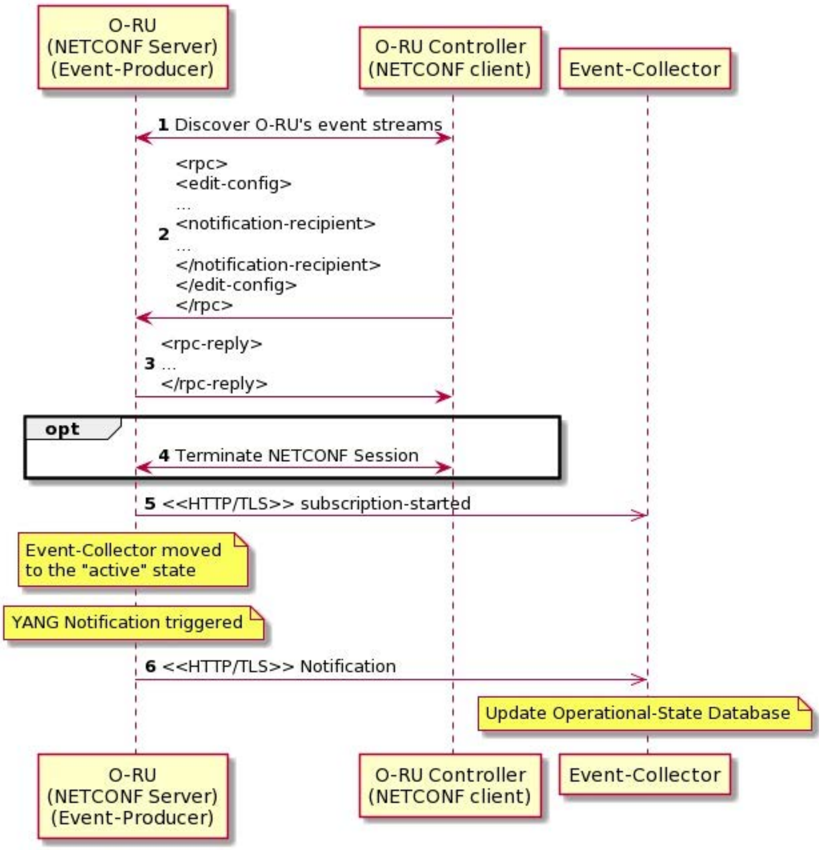


Figure 18.3-1:Message sequence exchange for provisioning a configured notification

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 "edit-config" 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 may 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, as specified in clause 2.5 of RFC8639 [37].

6) Upon an event that triggers a YANG notification, the O-RU sends a notification over HTTPS , according to clause 2.5 of RFC8639.

Post-condition: The Event-Collector is able to update its operational-state datastore with the information received in the notification.

18.4 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 18.4-1.

```
{ "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": "sftp://nms-user@10.10.10.10/home/pm/
C201805181300+0900_201805181330+0900_ABC0123456.csv",
      "status": "SUCCESS"
    }
  }
```


Figure 18.4-1:Example of a JSON encoded YANG Notification

18.5 Notification transport

As described in RFC8639 clause 2.5.7, an O-RU supporting configured subscriptions shall 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 clause 18.4 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 18.4-1.

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.

Figure 18.5-1 illustrates an example of a JSON encoded VES Event Carrying a YANG Notification.

```
{ "event": {
  "commonEventHeader": {
    "version": "4.1",
    "vesEventListenerVersions": "7.2",
    "domain": "stndDefined",
    "eventName": "ORU-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": "sftp://nms-user@10.10.10.10/home/pm/
C201805181300+0900_201805181330+0900_ABC0123456.csv",
          "status": "SUCCESS"
        }
      }
    },
    "stndDefinedFieldsVersion": "1.0"
  }
}}
```

Figure 18.5-1: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 [i.2]. The complete protocol stack is illustrated in Figure 18.5-2.

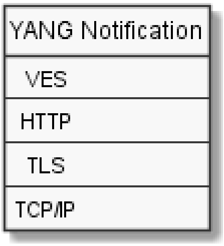


Figure 18.5-2: Protocol stack for O-RAN VES transport of YANG notifications

18.6 Monitoring the communications channel between O-RU and Event-Collector

18.6.1 Background

An O-RU controller can use the NETCONF monitoring capability described in clause 6.7 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 described in clause 6.2.7, i.e., in this version of the specification, this capability adopts the ONAP defined guidelines for Heartbeat, as defined in [i.2].

18.6.2 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 18.6.2-1.


```
{ "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 18.6.2-1:Example of a JSON encoded VES Event Carrying a Heartbeat Notification

18.6.3 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 Controller shall delete the configuration in the **event-collector-monitoring** container.

18.6.4 Heartbeat procedure

Figure 18.6.4-1 illustrates the message sequence exchange for heartbeat operation.

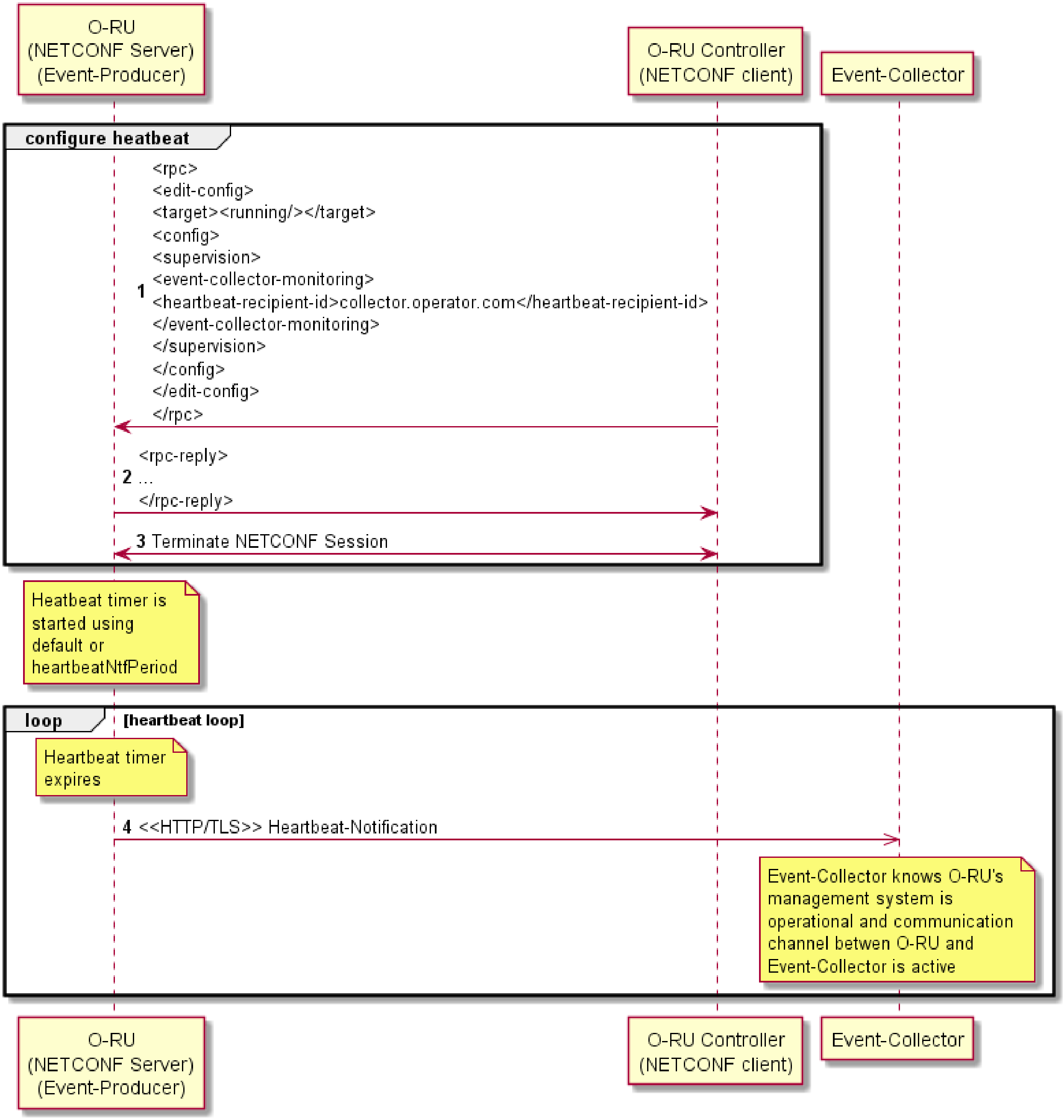


Figure 18.6.4-1:Message sequence exchange for heartbeat operation

In contrast to the monitoring of NETCONF connectivity described in clause 6.7 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 the present document.

19 Multi-Operator O-RU Operation

19.1 Introduction

The support by an O-RU of Multi-Operator O-RU operation is an optional capability, advertised by the O-RU indicating it supports the **SHARED-ORU-MULTI-OPERATOR** feature in its **o-ran-wg4-features** YANG model. All O-RUs that support the **SHARED-ORU-MULTI-OPERATOR** feature shall also support the **SHARED-ORU-MULTI-ODU** feature. These capabilities enable an O-RU to connect to multiple O-DUs that belong to different Shared Resource Operators.

The Multi-Operator O-RU architecture enables a Shared Resource Operator to configure an agreed subset of shared O-RU resources independently from configuration and operating strategies of the other Shared Resource Operators. More specifically, a Shared O-RU Host makes available its shared O-RUs to enable connectivity to the O-DUs of one or more Shared Resource Operators allowing these Shared Resource Operators to configure and control such Shared O-RU.

NOTE: How a Shared O-RU Host defines the partitioning of shared O-RU resources and communicates that information to a Shared Resource Operator is outside the scope of the present document.

19.2 High level shared O-RU architecture

When an O-RU is being configured by an independent Shared Resource Operator, a separate O-RU Controller associated with the Shared O-RU Host is required to configure the common aspects of the shared O-RU. Such a deployment is illustrated in Figure 19.2-1. When the Shared O-RU Host also operates an O-DU, the shared O-RU host may operate the M-Plane in either hybrid or hierarchical approach, as illustrated in figures 19.2-2 and 19.2-3 respectively. In either case, the Shared O-RU Host uses a NETCONF client with "sudo" privileges, as defined in clause 6.5, to configure the shared O-RU. Each Shared Resource Operator uses NETCONF clients with " carrier" privileges that have parallel M-Plane connections with the shared O-RU.

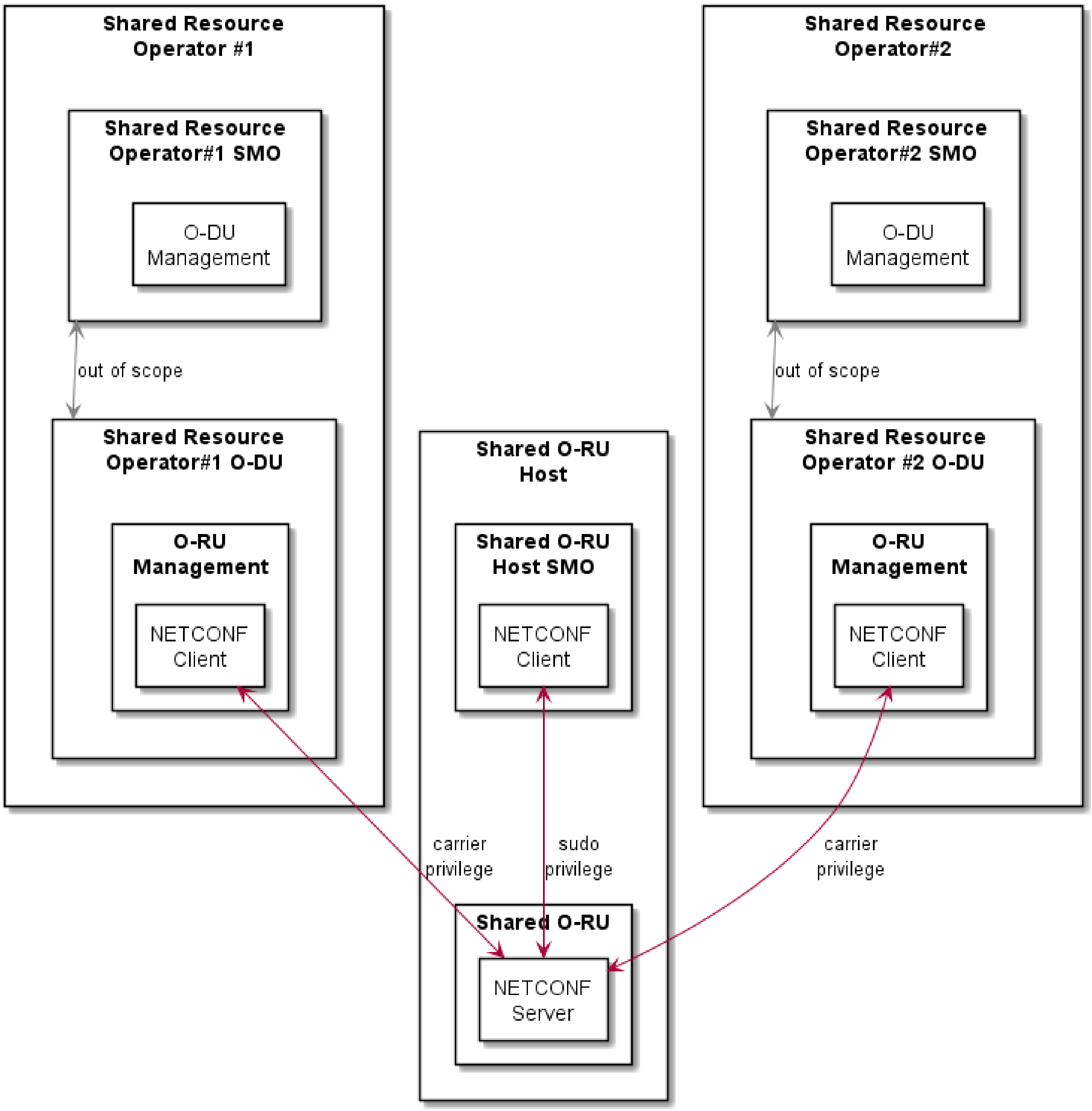


Figure 19.2-1 – Shared O-RU M-Plane Architecture where Shared O-RU Host is independent of Shared Resource Operator(s).

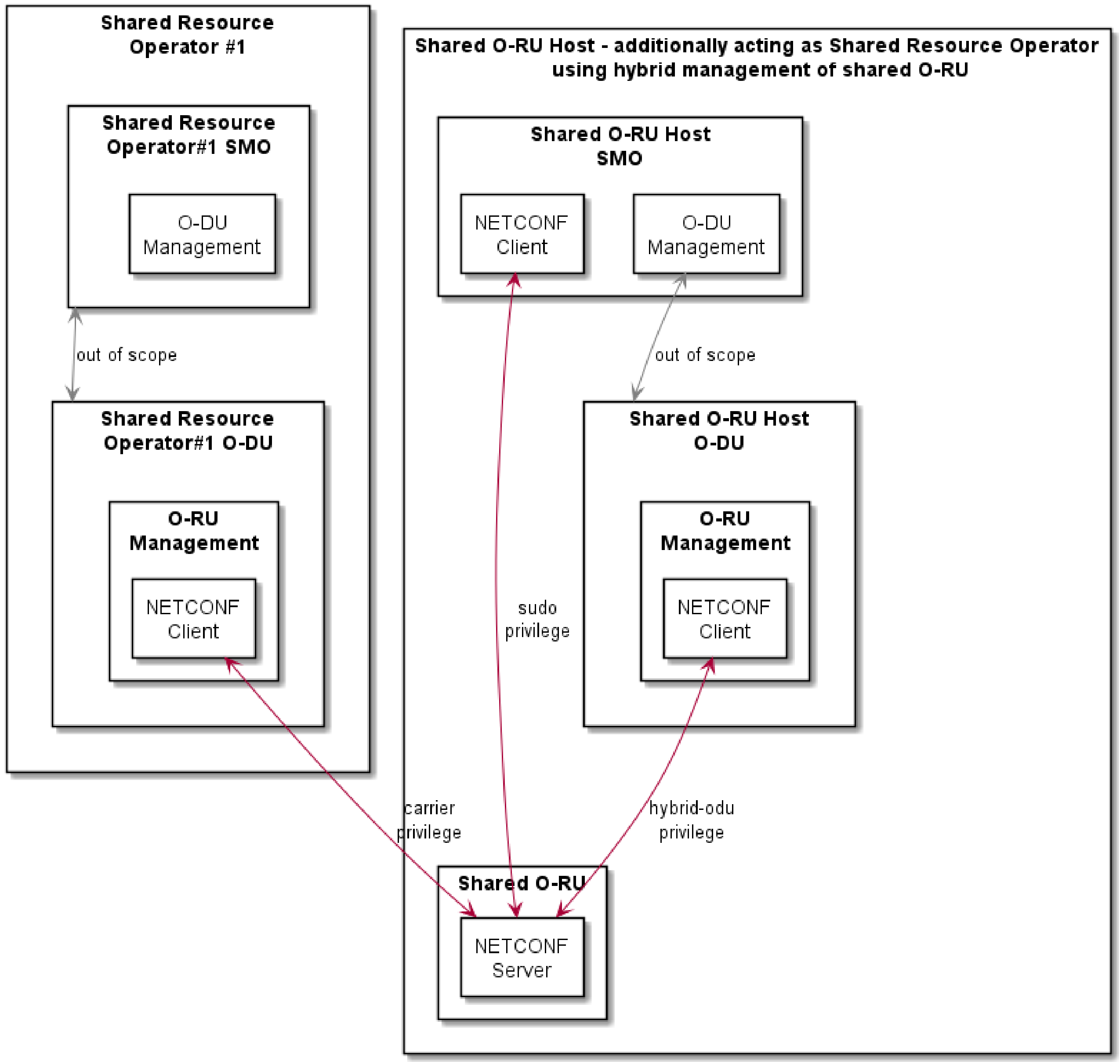


Figure 19.2-2 – Shared O-RU M-Plane Architecture where the Shared O-RU Host is additionally a Shared Resource Operator managing Shared O-RU using hybrid approach

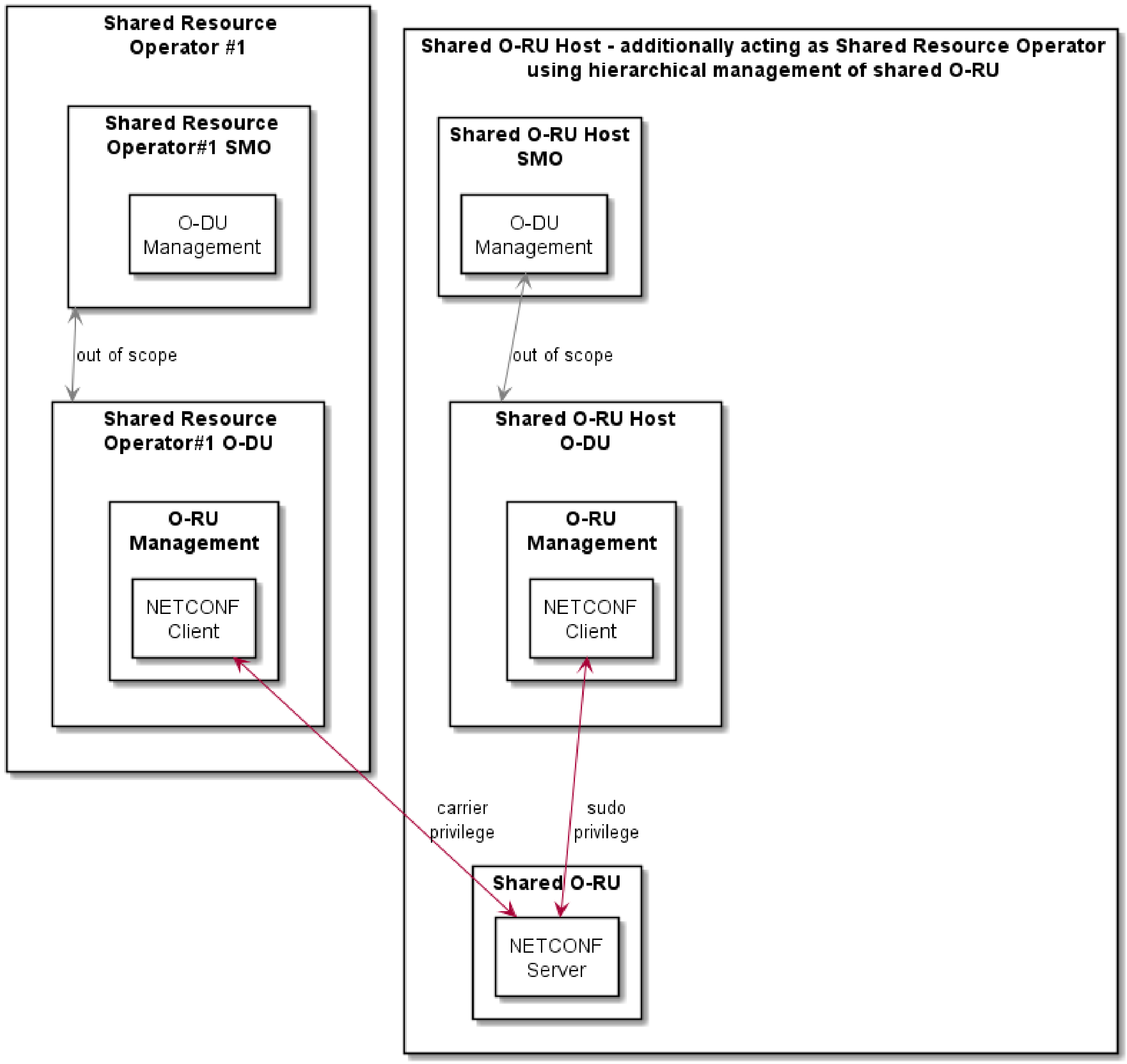


Figure 19.2-3 – Shared O-RU M-Plane Architecture where the Shared O-RU Host is additionally a Shared Resource Operator managing Shared O-RU using hierarchical approach

As illustrated in Figure 19.2-1, 19.2-2 and 19.2-3, from an Open Fronthaul perspective, the Shared O-RU is able to be deployed in variety of architectures.

Clause 5.1.2 requires all O-RUs to support multiple NETCONF sessions, with the number of simultaneous sessions exposed using the **maximum-simultaneous-netconf-sessions** schema node in o-ran-operations YANG model. A Multi-Operator O-RU that is able to support (n) simultaneous NETCONF sessions, shall be able to support (m) Shared Resource Operators, where the cumulative NETCONF sessions operated by the (m) Shared Resource Operators is less than (n).

19.3 Shared O-RU "start up" procedure

19.3.1 NETCONF server user account provisioning for shared resource operators

The Shared O-RU Host shall use the procedures defined in clause 6.4 and the o-ran-usermgmt YANG model to configure separate user accounts on the shared O-RU's NETCONF server for each Shared Resource Operator. The user account for a Shared Resource Operator that is not also a Shared O-RU Host should only be configured with "carrier" access control group, as defined in clause 6.5. A user account is identified as being associated with a Shared Resource Operator by having one or more configured **sro-id**s. The **sro-id** is allocated by the Shared O-RU Host to the Shared Resource Operator(s) and is used in partitioning shared O-RU resources.

NOTE 1: How a Shared O-RU Host communicates NETCONF user account information, including sro-id information, to a Shared Resource Operator is outside the scope of the present document.

NOTE 2: The format of the **sro-id** string is not defined and not interpreted by the shared O-RU. In one example, the operators can agree to use a PLMN-Id corresponding to the Shared Resource Operator as the **sro-id**.

19.3.2 NETCONF call home to shared resource operator O-DUs

The procedures defined in clauses 6.2.5 and 6.3 are used by the Shared O-RU Host to trigger the establishment of the NETCONF session with a NETCONF client of the Shared O-RU Host. The Shared O-RU Host can subsequently use the **o-ran-mplane-int** YANG model and the **configured-client-info** container to configure the Multi-Operator O-RU with NETCONF client information corresponding to the individual Shared Resource Operator(s) of the shared O-RU.

NOTE: How the Shared O-RU Host becomes aware of information to identify the Shared Resource Operator's NETCONF clients to be used with a specific shared O-RU is out of scope of the present document.

Following procedures defined in clause 6.3, this will trigger the Multi-Operator O-RU to perform additional call home procedures to any configured clients, which in this scenario will be the NETCONF clients corresponding to individual Shared Resource Operators.

19.3.3 Enhanced sro-id based NETCONF access control

The resource configuration framework defines particular named list entries in the shared O-RU's configuration that can be allocated to an **sro-id**. The enhanced NACM privileges defined in this clause enable the Shared Resource Operator to configure the shared O-RU's list entries that have been previously configured by the Shared O-RU Host with the shared resource operator's **sro-id**.

The O-RU NETCONF access control techniques defined in clause 6.5 are enhanced for operation with NETCONF clients corresponding to Shared Resource Operators of a shared O-RU, i.e., those where the user account of the NETCONF client has been configured with an **sro-id**. Specifically, the NETCONF access control read permissions for group name "carrier" are further refined for the following YANG models:

- urn:o-ran:processing-elements:x.y
- urn:o-ran:uplane-conf:x.y
- urn:o-ran:performance-management:x.y
- urn:o-ran:message5:x.y
- urn:o-ran:shared-cell:x.y

For the above models, the read privileges for specific node-instance-identifiers defined in Table 19.3.3-1 through Table 19.3.3-5 are refined based on the **sro-id**(s) associated with the user account of the NETCONF client. Normal NACM rules shall apply to any node-instance identifier not listed in the table.

Table 19.3.3-1: Refined NETCONF Access Control Read Privileges for the "carrier" group with YANG module o-ran:processing-elements:x.y

Restricted node-instance-identifier	Refined privileges for "carrier" group for NETCONF clients with user name user-list entry containing a configured sro-id
o-ran-elements:processing-elements/o-ran-elements:ru-elements	A NETCONF client with user name **user-list** entry containing a configured **sro-id** shall only have read privileges for those list entries where the **o-ran-elements:processing-elements/o-ran-elements:ru-elements/o-ran-elements:sro-id** leaf matches the **sro-id** of the NETCONF client.

Table 19.3.3-2: Refined NETCONF Access Control Read Privileges for the "carrier" group with YANG module o-ran:uplane-conf:x.y

Restricted node-instance-identifier	Refined privileges for "carrier" group for NETCONF clients with user name user-list entry containing a configured sro-id
o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-tx-links	A NETCONF client with user name **user-list** entry containing a configured **sro-id** shall only have read privileges for those list entries where the **o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-tx-links/o-ran-uplane-conf:sro-id** leaf-list includes the **sro-id** of the NETCONF client.
o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-rx-links	A NETCONF client with user name **user-list** entry containing a configured **sro-id** shall only have read privileges for those list entries where the **o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-rx-links/o-ran-uplane-conf:sro-id** leaf-list includes the **sro-id** of the NETCONF client.
o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-tx-endpoints	A NETCONF client with user name **user-list** entry containing a configured **sro-id** shall only have read privileges for those list entries where the **o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-tx-endpoints/o-ran-uplane-conf:sro-id** leaf-list includes the **sro-id** of the NETCONF client.
o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-rx-endpoints	A NETCONF client with user name **user-list** entry containing a configured **sro-id** shall only have read privileges for those list entries where the **o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-rx-endpoints/o-ran-uplane-conf:sro-id** leaf-list includes the **sro-id** of the NETCONF client.
o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:tx-array-carriers	A NETCONF client with user name **user-list** entry containing a configured **sro-id** shall only have read privileges for those list entries where the **o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:tx-array-carriers/o-ran-uplane-conf:sro-id** leaf-list includes the **sro-id** of the NETCONF client.
o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:rx-array-carriers	A NETCONF client with user name **user-list** entry containing a configured **sro-id** shall only have read privileges for those list entries where the **o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:rx-array-carriers/o-ran-uplane-conf:sro-id** leaf-list includes the **sro-id** of the NETCONF client.

Table 19.3.3-3: Refined NETCONF Access Control Read Privileges for the "carrier" group with YANG module o-ran:performance-management:x.y

Restricted node-instance-identifier	Refined privileges for "carrier" group for NETCONF clients with user name user-list entry containing a configured sro-id
o-ran-pm:performance-measurement-objects/o-ran-pm:rx-window-measurement-objects	A NETCONF client with user name **user-list** entry containing a configured **sro-id** shall only have read privileges for those list entries where the **o-ran-pm:performance-measurement-objects/o-ran-pm:rx-window-measurement-objects/o-ran-pm:tr-measured-result/o-ran-pm:name** leaf refers to an **o-ran-elements:processing-elements/o-ran-elements:ru-elements** list entry where the **o-ran-elements:processing-elements/o-ran-elements:ru-elements/o-ran-elements:sro-id** leaf represents the **sro-id** of the NETCONF client or where the **o-ran-pm:performance-measurement-objects/o-ran-pm:rx-window-measurement-objects/o-ran-pm:eaxc-measured-result/o-ran-pm:eaxc-id** leaf matches a value of **eaxcid** in the container **o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-rx-endpoints/o-ran-uplane-conf:e-axcid** where the **sro-id** in **o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-rx-endpoints/o-ran-uplane-conf:sro-id** matches the **sro-id** of the NETCONF client or where the **o-ran-pm:performance-measurement-objects/o-ran-pm:rx-window-measurement-objects/o-ran-pm:eaxc-measured-result/o-ran-pm:eaxc-id** leaf matches a value of **eaxcid** in the container **o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-tx-endpoints/o-ran-uplane-conf:e-axcid** where the **sro-id** in **o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-tx-endpoints/o-ran-uplane-conf:sro-id** matches the **sro-id** of the NETCONF client.
o-ran-pm:performance-measurement-objects/o-ran-pm:tx-measurement-objects	A NETCONF client with user name **user-list** entry containing a configured **sro-id** shall only have read privileges for those list entries where the **o-ran-pm:performance-measurement-objects/o-ran-pm:tx-measurement-objects/o-ran-pm:tr-measured-result/o-ran-pm:name** leaf refers to an **o-ran-elements:processing-elements/o-ran-elements:ru-elements** list entry where the **o-ran-elements:processing-elements/o-ran-elements:ru-elements/o-ran-elements:sro-id** leaf represents the **sro-id** of the NETCONF client or where the **o-ran-pm:performance-measurement-objects/o-ran-pm:tx-measurement-objects/o-ran-pm:eaxc-measured-result/o-ran-pm:eaxc-id** leaf matches a value of **eaxcid** in the container **o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-rx-endpoints/o-ran-uplane-conf:e-axcid** where the **sro-id** in **o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-rx-endpoints/o-ran-uplane-conf:sro-id** matches the **sro-id** of the NETCONF client or where the **o-ran-pm:performance-measurement-objects/o-ran-pm:tx-measurement-objects/o-ran-pm:eaxc-measured-result/o-ran-pm:eaxc-id** leaf matches a value of **eaxcid** in the container **o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-tx-endpoints/o-ran-uplane-conf:e-axcid** where the **sro-id** in **o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-tx-endpoints/o-ran-uplane-conf:sro-id** matches the **sro-id** of the NETCONF client.

NOTE: The enhanced **sro-id** based NACM privileges in Table 19.3.3-3 do not limit the privileges of a NETCONF client associated with an **sro-id** to read Rx and Tx performance management results when the **object-unit-id** is configured as RU. A Shared O-RU Host can exclude such value of **object-unit-id** from the configuration of the o-ran-performance-management YANG model to avoid enabling a first Shared Resource Operator from recovering aggregate O-RU level performance data that may include performance data related to the resources of a second Shared Resource Operator.

Table 19.3.3-4: Refined NETCONF Access Control Read Privileges for the "carrier" group with YANG module o-ran:message5:x.y

Restricted node-instance-identifier	Refined privileges for "carrier" group for NETCONF clients with user name user-list entry containing a configured sro-id
o-ran-msg5:ecpri-delay-message/o-ran-msg5:message5-sessions/o-ran-msg5:session-parameters	A NETCONF client with user name **user-list** entry containing a configured **sro-id** shall only have read privileges those list entries where the **o-ran-msg5:ecpri-delay-message/o-ran-msg5:message5-sessions/o-ran-msg5:session-parameter/o-ran-msg5:processing-element-name** leaf refers to an **o-ran-elements:processing-elements/o-ran-elements:ru-elements** list entry where the **o-ran-elements:processing-elements/o-ran-elements:ru-elements/o-ran-elements:sro-id** leaf represents the **sro-id** of the NETCONF client.

Table 19.3.3-5: Refined NETCONF Access Control Read Privileges for the "carrier" group with YANG module o-ran:shared-cell:x.y

Restricted node-instance-identifier	Refined privileges for "carrier" group for NETCONF clients with user name user-list entry containing a configured sro-id
o-ran-sc:shared-cell/o-ran-sc:shared-cell-config/o-ran-sc:shared-cell-copy-entities	A NETCONF client with user name **user-list** entry containing a configured **sro-id** shall only have read privileges for those list entries where the **o-ran-sc:shared-cell/o-ran-sc:shared-cell-config/o-ran-sc:shared-cell-copy-entities/o-ran-sc:sro-id** leaf-list includes the **sro-id** of the NETCONF client.
o-ran-sc:shared-cell/o-ran-sc:shared-cell-config/o-ran-sc:shared-cell-combine-entities	A NETCONF client with user name **user-list** entry containing a configured **sro-id** shall only have read privileges for those list entries where the **o-ran-sc:shared-cell/o-ran-sc:shared-cell-config/o-ran-sc:shared-cell-combine-entities/o-ran-sc:sro-id** leaf-list includes the **sro-id** of the NETCONF client.
o-ran-sc:shared-cell/o-ran-sc:shared-cell-config/o-ran-sc:shared-cell-copy-entities-selective-beam-id	A NETCONF client with user name **user-list** entry containing a configured **sro-id** shall only have read privileges for those list entries where the **o-ran-sc:shared-cell/o-ran-sc:shared-cell-config/o-ran-sc:shared-cell-copy-entities-selective-beam-id /o-ran-sc:sro-id** leaf-list includes the **sro-id** of the NETCONF client.
o-ran-sc:shared-cell/o-ran-sc:shared-cell-config/o-ran-sc:shared-cell-combine-entities-for-selective-beam-id	A NETCONF client with user name **user-list** entry containing a configured **sro-id** shall only have read privileges for those list entries where the **o-ran-sc:shared-cell/o-ran-sc:shared-cell-config/o-ran-sc:shared-cell-combine-entities-for-selective-beam-id /o-ran-sc:sro-id** leaf-list includes the **sro-id** of the NETCONF client.

For example, if the Shared O-RU has been configured to operate with two Shared Resource Operators, **sro-id** "23415" and **sro-id** "23410", then when a NETCONF client of the first Shared Resource Operator attempts to read the processing element configuration of the shared O-RU it can receive a reply as shown Figure 19.3.3-1, where the O-DU of the first Shared Resource Operator has an Ethernet MAC-address of 11:95:a0:af:5f:b9 and VLAN 100 is being used for the control and user-plane traffic between the shared O-RU and the O-DU of the first Shared Resource Operator.

```
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <processing-elements xmlns="urn:o-ran:processing-element:1.0">
      <ru-elements>
        <name>element1</name>
        <sro-id>23415</sro-id>
        <transport-flow>
          <interface-name>10Geth0.100</interface-name>
          <eth-flow>
            <ru-mac-address>00:e0:fe:00:23:30</ru-mac-address>
            <vlan-id>100</vlan-id>
            <o-du-mac-address>11:95:a0:af:5f:b9</o-du-mac-address>
          </eth-flow>
        </transport-flow>
      <ru-elements>
    <processing-elements>
  </data>
</rpc-reply>
```

Figure 19.3.3-1: example rpc-reply to a first sro-id "23415"

Conversely, if the NETCONF client of the second Shared Resource Operator attempts to read the processing element configuration it can receive a reply as shown in Figure 19.3.3-2, where the O-DU of the second Shared Resource

Operator has an Ethernet MAC-address of 5a:2a:a7:61:98:f0 and VLAN 200 is being used for the control and user-plane traffic between the Shared O-RU and the O-DU of the second Shared Resource Operator.

```
<rpc-reply message-id=”101” xmlns=”urn:ietf:params:xml:ns:netconf:base:1.0”>
  <data>
    <processing-elements xmlns=”urn:o-ran:processing-element:1.0”>
      <ru-elements>
        <name>element2</name>
        <sro-id>23410</sro-id>
        <transport-flow>
          <interface-name>10Geth0.200</interface-name>
          <eth-flow>
            <ru-mac-address>00:e0:fe:00:23:30</ru-mac-address>
            <vlan-id>200</vlan-id>
            <o-du-mac-address>5a:2a:a7:61:98:f0</o-du-mac-address>
          </eth-flow>
        </transport-flow>
      <ru-elements>
    <processing-elements>
  </data>
</rpc-reply>
```

Figure 19.3.3-2: example rpc-reply to a second sro-id "23410"

In addition to the enhanced read access privileges, the NETCONF server of the shared O-RU shall implement additional write access privileges for the following YANG model:

- urn:o-ran:uplane-conf:x.y

For the above model, the write privileges for specific node-instance-identifiers are refined based on the **sro-id** associated with the user account of the NETCONF client, as described in the table 19.3.3-6.

Table 19.3.3-6: Refined NETCONF Access Control Write Privileges for the "carrier" group with YANG module o-ran:uplane-conf:x.y

Restricted node-instance-identifier	Refined privileges for "carrier" group for NETCONF clients with user name user-list entry containing a configured sro-id
o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-tx-links/o-ran-uplane-conf:name	The NETCONF client shall be prohibited from writing to this schema node for all list entries.
o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-tx-links/o-ran-uplane-conf:sro-id	The NETCONF client shall be prohibited from writing to this schema node for all list entries.
o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-tx-links	A NETCONF client with user name **user-list** entry containing a configured **sro-id** shall be prohibited from creating list entries. A NETCONF client with user name **user-list** entry containing a configured **sro-id** shall only have write privileges to enable updating of those list entries where the **o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-tx-links/o-ran-uplane-conf:sro-id** leaf-list includes the **sro-id** of the NETCONF client.
o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-rx-links/o-ran-uplane-conf:name	The NETCONF client shall be prohibited from writing to this schema node for all list entries.
o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-rx-links/o-ran-uplane-conf:sro-id	The NETCONF client shall be prohibited from writing to this schema node for all list entries.
o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-rx-links	A NETCONF client with user name **user-list** entry containing a configured **sro-id** shall be prohibited from creating list entries. A NETCONF client with user name **user-list** entry containing a configured **sro-id id** shall only have write privileges to enable updating of those list entries where the **o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-rx-links/o-ran-uplane-conf:sro-id** leaf-list includes the **sro-id** of the NETCONF client.
o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-tx-endpoints/o-ran-uplane-conf:sro-id	The NETCONF client shall be prohibited from writing to this schema node for all list entries.
o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-tx-endpoints	A NETCONF client with user name **user-list** entry containing a configured **sro-id** shall be prohibited from creating list entries. A NETCONF client with user name **user-list** entry containing a configured **sro-id id** shall only have write privileges to enable updating of those list entries where the **o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-tx-endpoints/o-ran-uplane-conf:sro-id** leaf-list includes the **sro-id** of the NETCONF client.
o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-rx-endpoints/o-ran-uplane-conf:sro-id	The NETCONF client shall be prohibited from writing to this schema node for all list entries.
o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-rx-endpoints	A NETCONF client with user name **user-list** entry containing a configured **sro-id** shall be prohibited from creating list entries. A NETCONF client with user name **user-list** entry containing a configured **sro-id id** shall only have write privileges to enable updating of those list entries where the **o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-rx-endpoints/o-ran-uplane-conf:sro-id** leaf-list includes the **sro-id** of the NETCONF client.

Table 19.3.3-7: Refined NETCONF Access Control Write Privileges for the "carrier" group with YANG module o-ran:uplane-conf:x.y (continued)

Restricted node-instance-identifier	Refined privileges for "carrier" group for NETCONF clients with user name user-list entry containing a configured sro-id
o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:tx-array-carriers/o-ran-uplane-conf:sro-id	The NETCONF client shall be prohibited from writing to this schema node for all list entries.
o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:tx-array-carriers	A NETCONF client with user name **user-list** entry containing a configured **sro-id** shall be prohibited from creating list entries. A NETCONF client with user name **user-list** entry containing a configured **sro-id id** shall only have write privileges to enable updating of those list entries where the **o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:tx-array-carriers/o-ran-uplane-conf:sro-id** leaf-list includes the **sro-id** of the NETCONF client.
o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:rx-array-carriers/o-ran-uplane-conf:sro-id	The NETCONF client shall be prohibited from writing to this schema node for all list entries.
o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:rx-array-carriers	A NETCONF client with user name **user-list** entry containing a configured **sro-id** shall be prohibited from creating list entries. A NETCONF client with user name **user-list** entry containing a configured **sro-id id** shall only have write privileges to enable updating of those list entries where the **o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:rx-array-carriers/o-ran-uplane-conf:sro-id** leaf-list includes the **sro-id** of the NETCONF client.

19.3.4 Supervision monitoring between shared O-RU and shared resource operator O-DUs

An O-RU Controller associated with a Shared Resource Operator shall use the per O-DU monitoring capability specified in clause 14.1.1. The Shared O-RU Host shall configure the **per-odu-monitoring** container in o-ran-supervision YANG model with the **sro-id** and **odu-id** values agreed to be used by the Shared Resource Operator.

19.4 Shared O-RU interface management

19.4.1 VLAN and IP address management

The procedures defined in clauses 7.3 and 7.4 are used by the Shared O-RU Host to manage the VLAN and IP address configuration of the shared O-RU.

19.4.2 Processing element configuration

The procedures defined in clause 7.5 are used by the Shared O-RU Host to configure the processing elements in the shared O-RU. The optional **sro-id** leaf shall be used by the Shared O-RU Host to configure a particular **ru-element** list entry as being uniquely associated with a particular Shared Resource Operator.

NOTE 1: How the Shared O-RU Host becomes aware of information to identify the remote CU-Plane endpoints corresponding to the address(es) used by individual Shared Resource Operator O-DUs for control and user-plane traffic and included in the configuration of an **ru-element** list entry is out of scope of the present document.

The Shared Resource Operator O-DU needs to become aware of the local endpoint(s) in the O-RU configured for use in processing element(s) used by a particular Shared Resource Operator. One approach for the Shared Resource Operator O-DU to become aware of such information is for the NETCONF client of the Shared Resource Operator O-DU to subscribe to receive notifications due to updates of the O-RU's configuration datastore, using the techniques described in clause 6.4.

NOTE 2: Any other approaches by with the operator of a Shared Resource Operator O-DU becomes aware of information related to the configured processing elements is out of scope of the present document.

19.4.3 Shared resource operator O-DU verification of C/U-Plane transport connectivity

The procedures defined in clause 7.6 are used by the Shared O-RU Host to configure the operation of C/U plane transport connectivity checks. These checks are able to be simultaneously performed between the individual Shared Resource Operator O-DUs and the shared O-RU. A Shared Resource Operator O-DU is able to recover details of the configured Maintenance End Point (MEP) and other necessary information on the Shared O-RU by reading the configuration associated with the **o-ran-lbm** YANG model.

19.5 Shared O-RU C/U-Plane delay management

19.5.1 Adaptive delay operation with shared O-RU

Clause 14.2 of [2] specifies that the Shared O-RU Host is responsible for configuring shared O-RU aspects related to delay management. When a shared O-RU indicates that it supports the optional **ADAPTIVE-RU-PROFILE** feature, the Shared O-RU Host can decide not to employ such a feature. If the Shared O-RU Host decides to use the **ADAPTIVE-RU-PROFILE** feature, the Shared O-RU Host shall be responsible for configuring the **adaptive-delay-configuration** container.

NOTE 1: How the Shared O-RU Host determines the parameters to use in the **o-du-delay-profile** container and/or the **transport-delay** container is out of scope of the present document.

NOTE 2: Operation of the optional O-RU adaptive delay capability defined in clause 7.8 requires carriers to be disabled prior to the O-RU adapting its delay profile. If used by the Shared O-RU Host, how the shared O-RU host co-ordinates disablement of carriers amongst the Shared Resource Operators is out of scope of the present document.

19.5.2 Measuring transport delay parameters with Shared O-RU

As defined in clause 7.9, an O-RU that supports the optional eCPRI based delay measurement capability shall be able to support simultaneous operation of delay measurements over any configured processing element. Where these processing elements correspond to remote endpoints from different Shared Resource Operator O-DUs, the operation of eCPRI delay measurements will allow each Shared Resource Operator O-DU to recover the necessary timing compensation information from the O-RU.

19.6 Shared O-RU configuration management

19.6.1 Carrier configuration of the shared O-RU

The "carrier" NACM privileges defined in clause 6.5 and clause 19.3.3 prevent a NETCONF client with "carrier" privileges, and whose user account is configured with an **sro-id**, from creating particular named list entries for the following lists:

- o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-tx-links
- o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-rx-links
- o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-tx-endpoints
- o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:low-level-rx-endpoints
- o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:tx-array-carriers
- o-ran-uplane-conf:user-plane-configuration/o-ran-uplane-conf:rx-array-carriers

A NETCONF client with "carrier" privileges and whose user account is configured with an **sro-id** shall only be permitted to update list entries after the list entry has been created and configured with a **sro-id** associated with the Shared Resource Operator.

19.6.2 Notification of configuration updates to shared resource operators

All Multi-Operator O-RUs shall support the Notification of Updates to its Configuration Datastore functionality, as described in clause 9.4. Hence, any Shared Resource Operator may configure subscriptions to receive notifications of modifications to a shared O-RU's datastore, according to the defined NETCONF access control privileges for NETCONF client of the Shared Resource Operator. In particular, such an approach can be used by a Shared Resource Operator to determine when the shared O-RU host has configured **o-ran-uplane-conf** list entries which include the Shared Resource Operator's **sro-id**.

19.7 Shared O-RU performance management

The NETCONF client of a Shared Resource Operator is identified by using a **user** list entry in **o-ran-usermgmt** YANG model that contains a configured **sro-id**. Such a NETCONF client shall have restricted access privileges to the **o-ran-performance-management** YANG model as described in sub-section 19.3.3.

An O-RU supporting the **SHARED-ORU-MULTI-OPERATOR** feature should support the **GRANULARITY-TRANSPORT-MEASUREMENT** and/or the **GRANULARITY-EAXC-ID-MEASUREMENT** features. These allow the O-RU to report **rx-window-measurement-objects** on a per ru-element and/or eaxcid basis, meaning window measurements pertain to the transport flows and/or eaxcids associated with a particular Shared Resource Operator.

A Shared O-RU Host can configure multiple **remote-file-uploads** list entries corresponding to the individual file servers of different Shared Resource Operators. However, there is currently no role-based access control applied to file management based performance management reporting, as specified in clause 10.3.2. If access to configured measurement results needs to be controlled on a per Shared Resource Operator basis, file management based performance management should not be used.

19.8 Shared O-RU fault management

There is no role-based access control applied to O-RU fault management. Any NETCONF client, including those corresponding to Shared Resource Operator NETCONF client, are able to recover the **active-alarm-list** from the shared O-RU and are able to subscribe to receive notifications of future alarms as described in clause 11.

19.9 Synchronization aspects of shared O-RU

Clause 14.3 of [2] specifies the synchronization aspects of shared O-RU. Each individual Shared Resource Operator should subscribe to receive the notifications defined in the o-ran-sync YANG model.

19.10 Co-ordinating service impacting procedures

19.10.1 Reset operation

The various procedures required to be performed by Shared O-RU Host may necessitate performing reset of the shared O-RU. The Shared O-RU Host is responsible for co-ordinating such procedures with the individual Shared Resource Operators.

NOTE: How the Shared O-RU Host performs such co-ordination is out of scope of the present document.

19.10.2 Locked administrative state

The Shared O-RU Host is responsible for configuring the **admin-state** of the shared O-RU which permits the Shared O-RU Host to set the **admin-state** to **locked**. The Shared O-RU Host is responsible for co-ordinating any changes to the **admin-state** of the shared O-RU with the individual Shared Resource Operators.

NOTE: How the Shared O-RU Host performs such co-ordination is out of scope of the present document.

19.10.3 Antenna calibration

When the shared O-RU supports antenna calibration, the Shared O-RU Host shall be responsible for co-ordinating the operation of the antenna calibration procedure with individual Shared Resource Operators.

NOTE: How the Shared O-RU Host performs such co-ordination is out of scope of the present document.

19.11 Partitioning of shared O-RU carrier resources

19.11.1 Partitioning of Shared O-RU advertised resources

O-RU capabilities as a whole shall be advertised by the O-RU (e.g. parameters defined in o-ran-module-cap.yang, o-ran-beamforming.yang) at start-up to each SRO and Shared O-RU Host. Hence, each SRO sharing the O-RU shall be aware of complete O-RU resources. These O-RU resources shall be partitioned between multiple SROs based on pre-defined agreement (outside the scope of WG4). Each SRO is expected to configure and use it's partitioned O-RU resources so that the maximum O-RU capacity is not exceeded. To clarify this with an example, value of parameters 'max-gain' and 'min-gain' in o-ran-uplane-conf YANG model, shall not be exceeded when multiple SROs use the same tx-array for the carriers configured by each of them.

19.11.2 Partitioning of eAxC identities

The eAxC-IDs are required to be unique within the shared O-RU in the same direction (Tx or Rx) even across different processing elements that may correspond to connection to different Shared Resource Operators. The Shared O-RU Host is responsible for partitioning the eAxC-IDs between different Shared Resource Operators.

NOTE: How the Shared O-RU Host decides on the eAxC-ID partitioning policy and how the partition policy information is signalled to the respective Shared Resource Operators is out of scope of this document.

The procedures described in clause 19.6.2 can be used by the Shared O-RU Host to confirm whether a Shared Resource Operator is adhering to the eAxC-ID partitioning policy.

19.11.3 Partitioning of links, endpoints and array carriers

The Shared O-RU Host is responsible for creating the list entries for **low-level-tx-links**, **low-level-rx-links**, **low-level-tx-endpoints**, **low-level-rx-endpoints**, **tx-array-carriers** and **rx-array-carriers**, including unique configured **name** and one or more configured **sro-id**. The **sro-id** is used to partition links and array carriers between separate Shared Resource Operators.

19.11.4 Partitioning of static endpoints

The Shared O-RU Host is responsible for allocating **static-low-level-rx-endpoints** and **static-low-level-tx-endpoints** to individual Shared Resource Operators. This allocation may take into account Shared Resource Operator requirements related to static configuration for PRACH and SRS as well as TDD pattern configuration. To allocate a **static-low-level-tx-endpoint** to a Shared Resource Operator, the Shared O-RU Host shall configure the **low-level-tx-endpoints** list entry with the name of the **static-low-level-tx-endpoint** and the **sro-id** of the Shared Resource Operator. To allocate a **static-low-level-rx-endpoint** to a Shared Resource Operator, the Shared O-RU Host shall configure a **low-level-rx-endpoints** list entry with the name of the **static-low-level-rx-endpoint** and the **sro-id** of the Shared Resource Operator.

The procedures described in clause 19.6.2 can be used by the Shared O-RU Host to confirm whether a Shared Resource Operator is adhering to the static endpoint partitioning policy.

19.11.5 Shared O-RU beamforming configuration

When the shared O-RU supports the o-ran-beamforming YANG model, the allocation of array carrier resources and band configuration to individual Shared Resource Operators will refer to any corresponding static beamforming configuration of the shared O-RU.

In this version of the specification, the operation of a shared O-RU with **rt-bf-weights-update-support** set to true is not defined.

19.11.6 Shared O-RU with antenna line devices

When the shared O-RU is connected to one or more antenna line devices as described in clause 14.4, the shared O-RU host shall be responsible for the operation of the antenna line devices.

NOTE 1: How the details of the configuration of the one or more antenna line devices are shared with the respective Shared Resource Operators is out of scope of the present document.

The Shared O-RU Host may be required to co-ordinate antenna line operations with the different Shared Resource Operators, e.g., remote electrical tilt.

NOTE 2: How the Shared O-RU Host coordinates such operations is out of scope of the present document.

19.12 Example shared O-RU carrier configuration and operation procedure

An example procedure for configuring a shared O-RU is as follows:

1) The Shared O-RU Host is assigned "sudo" privileges. The Shared O-RU Host allocates an **sro-id** to the Shared Resource Operator and creates account(s) for the Shared Resource Operator's NETCONF client(s) on the O-RU's NETCONF server which includes the **sro-id.** The Shared O-RU Host communicates the account information to the Shared Resource Operator. Optionally, the Shared O-RU Host can subscribe to be notified of updates to the O-RU's configuration datastore, using the techniques specified in clause 9.4

2) The Shared O-RU Host configures the list of **ru-elements** for the Shared Resource Operator based on agreed O-DU transport identifiers. Each such element has the corresponding **sro-id** parameter set to the value allocated to the Shared Resource Operator.

NOTE 1: How the Shared O-RU Host becomes aware of information to identify the remote CU-Plane endpoints corresponding to the address(es) used by individual Shared Resource Operator O-DUs for control and user-plane traffic and included in the configuration of an **ru-element** list entry is out of scope of the present document.

3) The Shared O-RU Host creates the **tx-array-carriers** and **rx-array-carriers** list entries in relation to the agreed **tx-arrays** and **rx-arrays** for use by the Shared Resource Operator. Each of the list entries will have the corresponding **sro-id** allocated to the Shared Resource Operator configured in the list of **sro-ids** associated with the array carriers.

NOTE 2: Configuring the **tx-array-carriers** and **rx-array carriers** list entries includes mandatory leaves which are specific to the Shared Resource Operator's configuration. The Shared O-RU Host can agree with the Shared Resource Operator how it will configure any mandatory parameters in its initial list configuration.

4) The Shared O-RU Host creates **low-level-tx-endpoints** and **low-level-rx-endpoints** related to agreed partitioned **static-low-level-tx endpoints** and **static-low-level-rx-endpoints** respectively. Each of the list entries will have the corresponding **sro-id** allocated to the Shared Resource Operator configured in the list of sro-ids associated with the endpoints.

NOTE 3: Configuring the **low-level-tx-endpoints** and **low-level-rx-endpoints** list entries includes mandatory leaves which are specific to the Shared Resource Operator's configuration. The Shared O-RU Host can agree with the Shared Resource Operator how it will configure any mandatory parameters in its initial configuration.

NOTE 4: The Shared O-RU Host and the Shared Resource Operator can agree how information describing the list entries created in steps 2, 3 and 4 are to be shared. In one example, information is shared out of band between the two operators. In a second example, the operators can agree that the Shared Resource Operator will use a GET RPC to read the list entries configured by the Shared O-RU Host. Using this second approach, the NACM privileges ensure that the Shared Resource Operator only has permissions to read list entries that have been configured with its **sro-id**.

5) The Shared Resource Operator enters the pre-configured list entry information into its management systems

6) The Shared O-RU Host configures the **configured-client-info** container with IP address(es) of the NETCONF client(s) used by the Shared Resource Operator. This triggers the shared O-RU to call home to the configured client. The Shared Resource Operator uses the shared account information to establish a NETCONF session to the shared O-RU.

7) The Shared Resource Operator uses the NETCONF session to complete the configuration of the relevant **o-ran-uplane-conf** defined list entries for **tx-array-carriers, rx-array-carriers, low-level-tx-endpoints** and **low-level-rx-endpoints**. As the list entries are all configured with the **sro-id** of the Shared Resource Operator, then the NACM privileges described in clause 19.3.3 permit the Shared Resource Operator to over-write the initial mandatory parameters configured by the Shared O-RU Host with its required operational values. If the Shared O-RU Host subscribed to be notified of modifications to the O-RU's configuration store in step 1, then the Shared O-RU Host will be automatically notified of the committed changes to the shared O-RU's configuration by the Shared Resource Operator.

8) The Shared O-RU Host can read the modified configuration of the shared O-RU and the information used to determine whether the configuration by the Shared Resource Operator adheres to pre-agreed sharing policies.

NOTE 5: How to determine whether the configuration adheres to a pre-agreed policy as well as any response triggered by determining that a configuration does not adhere to a pre-agreed policy are not defined in the present document.

With the above steps successfully performed, the relationship between C/U-Plane application endpoints at Shared Resource Operator's O-DU and shared O-RU is configured.

9) The Shared O-RU Host and Shared Resource Operator can subscribe to receive notifications of O-RU alarms.

10) The Shared O-RU Host is responsible for configuring the performance measurements performed by the shared O-RU.

NOTE 6: The Shared O-RU Host and the Shared Resource Operator can agree how information describing the configured performance measurements are to be shared. In one example, information is shared out of band between the two operators. In a second example, the operators can agree that the Shared Resource Operator will use a GET RPC to read the performance management configuration. Using this second approach, the NACM privileges ensure that the Shared Resource Operator only has permissions to read performance management configuration associated with the **ru-elements** list entries and/or **exacid** values associated with the Shared Resource Operator's **sro-id**.

11) The Shared Resource Operator can subscribe to receive notifications related to the shared O-RU's performance management counters. The NACM privileges defined in clause 19.3.3 restricts the Shared Resource Operator to only be able to recover measurements associated with the **ru-elements** list entries and/or **exacid** values associated with the Shared Resource Operator's **sro-id**.

12) The Shared Resource Operator performs carrier activation by setting the value of the parameter "active" at tx-array-carrier element / rx-array-carrier element to "ACTIVE" for those list entries configured with the **sro-id** of the Shared Resource Operator.

13) The Shared Resource Operator performs shared resource operator supervision as specified in clause 19.3.4. If an O-RU enters shared resource supervision failure, then as described in clause 19.3.4, the shared O-RU will deactivate any carriers uniquely associated with the **sro-id** of the Shared Resource Operator and raise an alarm notification indicating that it has lost SRO based supervision.

19.13 Shared O-RU and LAA operation

Operation of M-Plane procedures as specified in clause 16 by a Shared Resource Operator using a NETCONF account with "carrier" privileges is not defined in the present document.

NOTE: This does not prevent the Shared O-RU Host from configuring LAA operation to operate in combination with its own component carriers.

19.14 Shared O-RU operation in combination with shared cell

Operation of M-plane procedures as specified in clause 17 by a Shared Resource Operator using a NETCONF account with "carrier" privileges is restricted based on the **sro-id** configured in the **user-list** entry for the NETCONF account. A shared O-RU that indicates it supports the o-ran-shared-cell model in its YANG library shall support the partitioning of copy and combine entities on an **sro-id** basis and limit read access to those list-entries where the configured **sro-id** matches that of the NETCONF account, as specified in clause 19.3.3.

The present document further restricts all write access to the shared-cell YANG model from NETCONF accounts with user name **user-list** entry containing a configured **sro-id** and "carrier" privileges. As a consequence, the Shared O-RU Host shall be responsible for configuring shared cell copy and combine parameters on behalf of each Shared Resource Operator.

NOTE: How the Shared O-RU Host co-ordinates the configuration of copy and combine parameters between one or more Shared Resource Operators is outside the scope of the present document.

20 Network energy saving

20.1 Introduction

This section describes the requirements and scope of Energy Savings Techniques relevant to Fronthaul M-Plane interface.

20.2 Carrier deactivation for energy saving

20.2.1 High level principle of carrier deactivation for energy saving

This clause provides basic description of how an O-RU's carrier can be deactivated to achieve the energy savings within an O-RU.

In order to achieve the energy saving, NETCONF Client uses the existing parameter [tr]x-array-carrier::**active** to activate or de-activate [tr]x-array-carriers. NETCONF Client configures the parameter to INACTIVE to deactivate specific carrier. This causes the parameter [tr]x-array-carrier::**state** transition to DISABLED (can go through BUSY). In a result, power consumption of O-RU is reduced because O-RU does not need to process C-Plane and U-plane traffic.

Additionally, when [tr]x-array-carriers are inactive and when **energy-saving-enabled** is set to true, the O-RU may turn off circuitry associated with carrier processing to further reduce power consumption.

NOTE 1: The actual reduced value of power consumption depends on the O-RU’s implementation.

NOTE 2: The operation of an O-RU when **active** is set to SLEEP is not currently defined in the present document.

NOTE 3: For details about how parameters energy-saving-enabled and [tr]x-array-carrier::**active** interact - please see clause 9.1.3 "Modify state".

20.2.2 Synchronization aspects for carrier deactivation for energy saving

When **power-state** of o-ran-hardware is SLEEPING, C/U/S functions on O-RU may be stopped to reduce energy consumption. If O-RU stops S-plane function, a certain period which depends on the O-RU implementation will be required to make carrier activation again. NETCONF client should set **energy-saving-enabled** to FALSE to ensure O-RU is ready to immediately activate a carrier.

Annex A (normative): Common alarm definition

A.1 Introduction

This clause contains common alarms which may be supported.

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 common alarms table has following columns:

NOTE 1: The table columns do not represent the entire set of alarm fields defined in the o-ran-fm.yang model.

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 when fulfilled generates alarm. If filtering time is needed, then it shall be defined in this column.

Cancel condition – Defines conditions which when fulfilled cancels alarm. If filtering time is needed, then it shall 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). Alarm sources may directly reference a defined component or may use a predefined textual description. The following list provides mapping between alarm sources and XML encodings describing the names of components that may be alarm sources:

- **Module:** < hardware xmlns= "urn:o-ran:hardware:1.0"><component><o-ran-name/></component></hardware>
- **Fan supervision:** <fan-tray xmlns= "urn:o-ran:fan:1.0"><fan-state><name/></fan-state></fan-tray>
- **GNSS:** <sync xmlns= "urn:o-ran:sync:1.0"><gnss-status><name/></gnss-status></sync>
- **External input:** <external-io xmlns= "urn:o-ran:externalio:1.0"><input><name/></input></external-io>
- **External output:** <external-io xmlns= "urn:o-ran:externalio:1.0"><output><name/></output></external-io>
- **ALD port:** <ald-ports-io xmlns= "urn:o-ran:ald-port:1.0"><ald-port><name/></ald-port></ald-ports-io>
- **Port transceiver:** <port-transceivers xmlns= "urn:o-ran:transceiver:1.0"><port-transceiver-data><name/></ port-transceiver-data ></port-transceivers>
- **Ethernet interface:** <interfaces xmlns= "urn:ietf:params:xml:ns:yang:ietf-interfaces"><interface><name/></interface></interfaces>

- **Processing element:** <processing-elements xmlns= "urn:o-ran:processing-element.1.0”><ru-elements><name/></ru-elements></processing-elements>
- **Low level tx link:** <o-ran-uplane-conf xmlns= "urn:o-ran:uplane-conf.1.0”><low-level-tx-links><name/></low-level-tx-links></ o-ran-uplane-conf>
- **Low level rx link:** <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>
- **Tx array:** <o-ran-uplane-conf xmlns= "urn:o-ran:uplane-conf.1.0”><tx-arrays><name/></tx-arrays></o-ran-uplane-conf>
- **Rx array:** <o-ran-uplane-conf xmlns= "urn:o-ran:uplane-conf.1.0”><rx-arrays><name/></rx-arrays></o-ran-uplane-conf>
- **O-DU supervision:** <o-ran-supervision xmlns= "urn:o-ran:supervision:1.0"><per-odu-monitoring><odu-id/></per-odu-monitoring></o-ran-supervision>

The following list provides mapping between alarm sources and predefined textual descriptions used to define the alarm sources:

- **Antenna line:** *array’s name*:*array element,* where *array’s name* follows either **Tx array** or **Rx array** name as defined above and *array element* is an ordinal number defining element within an array (see *k* in clause 12.5.3 Identification and Ordering of Array Elements in [2]).

  EXAMPLE 1: "tx-array-1:0" (alarm on the array element k=0 on the tx-array named "tx-array-1")

  EXAMPLE 2: "rx-array-2:15" (alarm on the array element k=15 on the rx-array named "rx-array-2")

If an alarm is caused by an external device (fault is out of the O-RU, e.g., caused by an Antenna Line Device), then fault source may not fit to any of the fault sources described above and additional text in an alarm notification is needed to clearly describe what may be a possible fault source.

NOTE 2: Alarms, especially those defined in vendor specific range, may use other alarm sources, which are not described above.

Severity – Defines severity of the alarm as specified in ITU X.733 [30].

- Critical – sub-unit for which alarm has been generated is not working and cannot be used.
- 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.

Alarm Type - Indicates the type of alarm, as specified in 3GPP 28.5.3.2 [60], table 12.2.1.4.4.6-1.

- The table of alarms in Annex A includes alarm-type values for the common alarms.

Table A.1-1: Common O-RU Alarms

Fault id	Name	Meaning	Start condition	Cancel condition	NETCONF Server actions on detection	NETCONF Server actions on cancel	System recovery actions	Source	Severity	Alarm Type
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	ENVIRONMENTAL-ALARM
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	ENVIRONMENTAL-ALARM
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	ENVIRONMENTAL-ALARM

Fault id	Name	Meaning	Start condition	Cancel condition	NETCONF Server actions on detection	NETCONF Server actions on cancel	System recovery actions	Source	Severity	Alarm Type
4	Temperature too low	During start-up: The temperature inside the unit is too low. Heating of unit is ongoing. Wait until the alarm is cancelled. 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 start-up) is mandatory within x minutes.	HW implementation specific (e.g., enable heating). (NOTE 3)	HW implementation specific (e.g., disable the heating). (NOTE 4)	None.	Module	Critical during start-up Minor during runtime	ENVIRONMENTAL-ALARM
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)	EQUIPMENT-ALARM
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	EQUIPMENT-ALARM
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	EQUIPMENT-ALARM
8	Filter unit faulty	Major failure has been detected by the filter.	HW implementation specific.	HW implementation specific.	None.	None.	None.	Antenna line	Critical	EQUIPMENT-ALARM
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	QUALITY-OF-SERVICE-ALARM
10	RF Module overvoltage protection faulty	Module's overvoltage protection is broken.	HW implementation specific.	None.	None.	None.	None.	Module	Minor	EQUIPMENT-ALARM
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	PROCESSING-ERROR-ALARM

Fault id	Name	Meaning	Start condition	Cancel condition	NETCONF Server actions on detection	NETCONF Server actions on cancel	System recovery actions	Source	Severity	Alarm Type
12	Critical file not found	Critical configuration file is missing.	Critical configuration file is detected missing.	None.	None.	None.	Reset.	Module	Critical	PROCESSING-ERROR-ALARM
13	File not found	Non-critical configuration file is missing.	Non-critical configuration file is detected missing.	None.	None.	None.	None.	Module	Major	PROCESSING-ERROR-ALARM
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	PROCESSING-ERROR-ALARM
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	EQUIPMENT-ALARM
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	PROCESSING-ERROR-ALARM
17	No external sync source	The Unit lost lock to all incoming clocks.	HW implementation specific.	HW implementation specific.	None.	None.	Reset or none.	Module	Major	EQUIPMENT-ALARM
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	Critical	EQUIPMENT-ALARM
19	TX out of order	TX path is not usable.	HW implementation specific.	HW implementation specific.	None.	None.	Reset or none.	Antenna line	Critical	EQUIPMENT-ALARM
20	RX out of order	RX path is not usable.	HW implementation specific.	None.	None.	None.	Reset or none.	Antenna line	Critical	EQUIPMENT-ALARM

Fault id	Name	Meaning	Start condition	Cancel condition	NETCONF Server actions on detection	NETCONF Server actions on cancel	System recovery actions	Source	Severity	Alarm Type
21	Increased BER detected on the optical connection	Increased bit error rate has been detected on the optical link which results in sporadic 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	QUALITY-OF-SERVICE-ALARM
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	PROCESSING-ERROR-ALARM
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	PROCESSING-ERROR-ALARM
24 (NOTE 5)	Unit blocked	Unit is blocked.	Parameter **admin-state** of the Module element is set to "locked".	Parameter **admin-state** of the Module Element is set to "unlocked" or "shutting-down"	Blocked unit shuts down all RF emission and turns off power on antenna lines and ALD ports	None.	None.	Module	Critical	EQUIPMENT-ALARM
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	EQUIPMENT-ALARM
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	EQUIPMENT-ALARM
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	EQUIPMENT-ALARM

Fault id	Name	Meaning	Start condition	Cancel condition	NETCONF Server actions on detection	NETCONF Server actions on cancel	System recovery actions	Source	Severity	Alarm Type
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	COMMUNICATIONS-ALARM
29	Transceiver Fault	Unit has detected a transceiver fault	HW and SW implementation specific.	None.	None	None.	None.	Module	Critical	EQUIPMENT-ALARM
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	EQUIPMENT-ALARM
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 that uses the fault source is disabled/removed	SW implementation specific	None	None	Specific to low-level-[tr]x-endpoint in the C/U-Plane message	Major, Minor or warning	COMMUNICATIONS-ALARM
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	PROCESSING-ERROR-ALARM

Fault id	Name	Meaning	Start condition	Cancel condition	NETCONF Server actions on detection	NETCONF Server actions on cancel	System recovery actions	Source	Severity	Alarm Type
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	EQUIPMENT-ALARM
34	Clock source failure	Clock source has failed	O-RU detects clock source failure	HW implementation specific.	set the system clock to the current time recovered from available external sources	None	None	Module	Critical	EQUIPMENT-ALARM
35	Lost O-DU ID based Supervision	An O-RU has lost supervision from one O-DU NOTE: this alarm is only supported by O-RUs that implement the SHARED-ORU-MULTI-ODU feature	Major – carriers associated with the O-DU are disabled	Re-establishment of NETCONF supervision by O-DU	Disable operation of carrier resources associated with O-DU	None	None	odu-id	Major	COMMUNICATIONS-ALARM
NOTE 3 : Actions taken shall not interfere with normal unit operation if such is commanded by NETCONF Client NOTE 4: Actions taken shall not interfere with normal unit operation if such is commanded by NETCONF Client. NOTE 5: The **admin-state** is conditioned on the optional **hardware-state** feature. The O-RU can support fault id #24 (Unit blocked) only if it supports optional feature **hardware-state.**										

Annex B (normative): Counters

B.1 Counter definition

Table B.1-1: Counters definition

measurement-group	measurement-object	report-info	object-unit	Note
transceiver-stats	RX_POWER 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	max and time min and time first and time latest and time frequency-bin-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-bin-table.
rx-window-stats	RX_ON_TIME 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	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.
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.

Table B.1-1: Counters definition (continued)

measurement-group	measurement-object	report-info	object-unit	Note
epe-stats	POWER TEMPERATURE VOLTAGE CURRENT	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 specified in clause 5.1.1.19 of 3GPP TS 28.552 [57] Unit of power: watts (W) Temperature measured using method specified in clause 5.1.1.19 of 3GPP TS 28.552 [57] Unit of temperature: Celsius Voltage measured using method as specified in clause 5.1.1.19 of 3GPP TS 28.552 [57] Unit of voltage: Volts Current measured using method specified in clause 5.1.1.19 of 3GPP TS 28.552 [57] Unit of current: Amperes
symbol-rssi-stats-object	ALL-UL-SYMBOLS CONFIGURED-SYMBOLS	max min avg frequency-bin-table	rx-array-carrier	Type decimal64 including 1 fraction-digit is used for max, min and avg. Type uint32 is used for frequency-bin-table.

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.2 Transceiver statistics

B.2.1 Transceiver measurements

The transceiver-measurement includes the performance measurement of transceivers as shown in the following table.

Table B.2.1-1: Transceiver Measurements

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.

B.2.2 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 clause B.2.3.

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 clause B.2.3 below.

After updates O-RU waits another interval to elapse.

B.2.3 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.
- If $b \leq x$ and n > 1 then bin h_{n-1} is incremented.

- If $a \le x$ and $x < b$ and $n > 2$ then bin h_k is incremented for k such that
  - $k-1 \le (n-2) * (x-a) / (b-a) < k$.

where the value of a 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 ... n-2, h_k is a number of values x such that
- $a + (b-a) * (k-1) / (n-2) \le x < a + (b-a) * (k) / (n-2)$.
- For k = n-1, h_k is a number of values x such that $b \le x$.

Example:

function = LOG_{10}, **bin-count** = 14, **lower-bound** = -12, **upper-bound** = 0

- parameter value s = 0, x = f(0) = -128, -128 < -12 = a → h_0 is incremented
- parameter value s = $1e^{-12}$, x = f($1e^{-12}$) = -12, (14-2)*(-12-(-12))/(0-(-12)) = 12*0/12 < 1 → 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 → h_1 is incremented
- parameter value s = $1e^{-1}$, x = f($1e^{-1}$) = -1, (14-2)*(-1-(-12))/(0-(-12)) = 12*11/12 < 12 → h_{12} is incremented
- parameter value s = 1, x = f(1) = 0, 0 ≥ 0 = b → h_{13} is incremented

B.3 Rx window statistics

B.3.1 Rx window measurement

The rx-window-measurement includes the performance measurement for the reception window as following table.

Table B.3-1: Rx Window Measurement

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**
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**

B.4 Tx statistics

The tx-measurements include the measurement according to the following table.

Table B.4-1: Tx Measurement

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**)**

B.5 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 shall report its supported measurement objects per hardware component class.

Table B.5-1: Energy, Power and Environmental Measurements

measurement-object	Description
POWER	Value of measured power consumed by identified hardware component
TEMPERATURE	Value of measured temperature of identified hardware component
VOLTAGE	Value of measured voltage of identified hardware component
CURRENT	Value of measured current of identified hardware component

B.6 Symbol RSSI statistics

B.6.1 Statistics calculation

The symbol-rssi-stats is the time domain RSSI per symbol, the reference point for the TD-RSSI shall be the antenna connector of the O-RU. The value of Received Signal Strength Indicator(RSSI) per rx-array-carrier per configured OFDM symbol is measured. The RSSI shall be calculated as the linear average of the total received power observed in the configured OFDM symbol in the measurement bandwidth from all sources including co-channel serving and non-serving cells, adjacent channel interference, thermal noise, etc, over the total number of antenna elements of the array. The unit of the reported RSSI is dBm.

If analogue or hybrid beamforming is enabled, the beamId used for RSSI measurement is:

- When there is allocation of a beamId in this symbol, O-RU use that beamId for RSSI measurement;
- When there is no allocation of a beamId in this symbol, it is up to O-RU implementation, for example, the O-RU can choose to use a common beamId or use a previous allocated beamId;

Table B.6.1-1: Symbol RSSI Measurements

measurement-object	Description
ALL-UL-SYMBOLS	Measure and report symbol-rssi separately for all UL symbols in every configured number of slots (as defined by 'period' in 'symbol-rssi-measurement-objects'. And the UL symbols are decided by 'configurable-tdd-pattern', 'static-srs-configuration', 'static-prach-configuration', and 'dataDirection' in the C-plane messages. This option is recommended for static TDD case. If this option is used in dynamic TDD case, then O-RU measures only the allocates UL symbols because O-RU may not know 'candidate UL symbols' which are not allocated.
CONFIGURED-SYMBOLS	Measure and report symbol-rssi separately for all configured symbols as defined by the leaf-list 'symbol-index'. This can be used for non-dynamic TDD as well as dynamic TDD cases, the O-RU should measure all configured symbols, irrespective of whether the UL symbol is allocated or not. If a c-plane message indicates a symbol within the 'symbol-index' list to be a DL symbol, O-RU shall not measure rssi on this symbol.

B.6.2 Frequency Table Generation

Same as B.2.2.

Annex C (informative): Optional multi-vendor functionality

C.1 Optional multi-vendor 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.

NOTE: Table C.1-1, C.1-2 do not apply for factory default software.

Table C.1-1: Optional O-RAN Namespace

No	Optional Functionality	Reference	Namespace
1	Antenna Line Device	Clause 14.4	"urn:o-ran:ald-port:x.y" "urn:o-ran:ald: x.y "
2	External IO Port	Clause 14.5	"urn:o-ran:external-io:x.y "
3	eCPRI delay measurement	Clause 7.7	"urn:o-ran:message5:x.y "
4	UDP Echo functionality for IP based transport verification	Clause 7.6	"urn:o-ran:udpecho:x.y "
5	Beamforming	Clause 15.4	"urn:o-ran:beamforming:x.y "
6	FAN	-	"urn:o-ran:fan:x.y"
7	LAA	Clause 16	"urn:o-ran:laa:x.y " "urn:o-ran:laa-operations:x.y "
8	Antenna calibration	Clause 5.5	"urn:o-ran:antcal: x.y "
9	Shared cell (common to FHM and Cascade modes)	Clause 17	"urn:o-ran:shared-cell:x.y" "urn:o-ran:ethernet-fwd:x.y"
10	Configured subscription transported using VES common header	Clause 18	"urn:o-ran:ves-sn:1.0"
11	Certificates to name mapping	Clause 6.4.3.2	“urn:o-ran:certificates:1.0”

Table C.1-2: Optional Non-O-RAN Namespace

No	Optional Functionality	Reference	Namespace
1	Notification of Updates to Configuration Datastore	Clause 9.4	"urn:ietf:params:xml:ns:yang:ietf-netconf-notifications "
2	(Transport agnostic) subscriptions to YANG notifications	Clause 18	"urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications"
3	Certificates to name mapping	Clause 6.4.3.2	“urn:ietf:params:xml:ns:yang:ietf-x509-cert-to-name”
4	IEEE 802.1X Port based Access Control	Clause 7.12	“urn:ieee:std:802.1X:yang:ieee802-dot1x” “urn:ieee:std:802.1Q:yang:ieee802-types” “urn:ietf:params:xml:ns:yang:iana-crypt-hash” “urn:ietf:params:xml:ns:yang:ietf-system” “urn:ieee:std:802.1X:yang:ieee802-dot1x-types”
5	Trust store	Clause 6.2	"urn:ietf:params:xml:ns:yang:ietf-truststore"
6	Continuity Check Message	Clause 7.6	"urn:ieee:std:802.1Q:yang:ieee802-dot1q-cfm"

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 C.1.3 describes those optional features that when supported result in YANG models becoming optional.

Table C.1-3: Not mandatory O-RAN Namespace for FHM.

No	Optional Functionality	Reference	Namespace
1	FHM in shared cell	Clause 17.5.1	"urn:o-ran:module-cap:x.y " "urn:o-ran:uplane-conf:x.y "

C.2 Optional YANG features

Some of the O-RAN defined YANG models define optional feature support using the ability to tag a portion of the model with a feature name. Theses portions of the model are only valid on O-RUs that indicate they support the specific YANG feature in their YANG library. The definition of the portion of the model tagged with a feature name can include definitions which allow an O-RU Controller to control when a particular O-RU capability associated with a feature tag is activated and/or deactivated. The optional capabilities identified using YANG feature tag names defined in the O-RAN defined YANG models are shown in table C.2-1 below.

Table C.2-1: Optional O-RAN WG4 defined feature support

	Namespace	YANG Feature Name Tag	Description	Optional Feature Control
1	"urn:o-ran:ald-port:x.y "	OVERCURRENT-SUPPORTED	ALD overcurrent reporting	Subscription to over-current notifications
2	"urn:o-ran:antcal:1.0"	O-RU-COORDINATED-ANT-CAL	O-RU needs user traffic to be co-ordinated from O-DU for antenna calibration	/antenna-calibration/self-calibration-policy/coordinated-calibration-allowed
3		O-RU-COORDINATED-ANT-CAL-MULTIPLE-TIME-RESOURCE	Calibration with multiple timing resource sets	/antenna-calibration/self-calibration-policy/coordinated-calibration-multiple-time-resources-allowed
4	"urn:o-ran:beamforming:x.y "	MODIFY-BF-CONFIG	Dynamic Beamforming Configuration	activate-beamforming-config and/or activate-beamforming-config-by-capability-group remote procedure calls
5		BEAM-TILT	Tilting pre-defined beams	modify-predefined-beam-tilt-offset remote procedure call
6	"urn:o-ran:compression-factors:x.y "	CONFIGURABLE-FS-OFFSET	Configurable FS offset	/user-plane-configuration /low-level-tx-endpoints /compression/fs-offset and/or /user-plane-configuration /low-level-tx-endpoints /compression/dynamic-compression-configuration/fs-offset and/or /user-plane-configuration /low-level-rx-endpoints /compression/fs-offset and/or /user-plane-configuration /low-level-rx-endpoints /compression/dynamic-compression-configuration/fs-offset
7	"urn:o-ran:delay:x.y "	ADAPTIVE-RU-PROFILE	Adaptive O-RU delay profile	/delay-management/adaptive-delay-configuration
8	" urn:o-ran:fm:1.0"	HISTORICAL-ALARM-LIST	Historical Alarms List	M-Plane activation not applicable
9	"urn:o-ran:hardware:x.y "	ENERGYSAVING	O-RU Energy saving	/hardware/component/energy-saving-enabled
10	"urn:o-ran:interfaces:x.y "	ALIASMAC-BASED-CU-PLANE	Alias MAC address based C/U transport	/processing-elements/ru-elements/transport-flow/aliasmac
11		UDPIP-BASED-CU-PLANE	UDP/IP based C/U Transport	/processing-elements/ru-elements/transport-flow/udpip-flow
12	"urn:o-ran:module-cap:x.y "	CONFIGURABLE-TDD-PATTERN-SUPPORTED	Configurable TDD pattern	/user-plane-configuration/tx-array-carriers/configurable-tdd-pattern and/or /user-plane-configuration/rx-array-carriers/configurable-tdd-pattern
13		DSS_LTE_NR	Dynamic Spectrum Sharing	/user-plane-configuration/tx-array-carriers/type and/or /user-plane-configuration/rx-array-carriers/type
14		EAXC-ID-GROUP-SUPPORTED	EAXC-ID Grouping	/user-plane-configuration/eaxc-id-group-configuration
15		LAA	LAA Support	/user-plane-configuration/tx-array-carriers/laa-carrier-configuration
16		TRANSPORT-FRAGMENTATION	Transport Fragmentation	M-Plane activation not applicable

17		PRACH-STATIC-CONFIGURATION-SUPPORTED	Static configuration of PRACH pattern	/user-plane-configuration/static-prach-configurations
18		SRS-STATIC-CONFIGURATION-SUPPORTED	Static configuration of SRS pattern	/user-plane-configuration/static-srs-configurations
19	"urn:o-ran:performance-management:x.y "	GRANULARITY-EAXC-ID-MEASUREMENT	EAXC_ID in rx-window-measurement	/performance-measurement-objects/rx-window-measurement-objects/object-unit
20		GRANULARITY-TRANSPORT-MEASUREMENT	TRANSPORT in rx-window-measurement	/performance-measurement-objects/rx-window-measurement-objects/object-unit
21	"urn:o-ran:processing-element:x.y"	SHARED_CELL	Shared cell support	/ru-elements/transport-flow/north-eth-flow and/or /ru-elements/transport-flow/south-eth-flow
22	"urn:o-ran:shared-cell:x.y"	FHM	FHM support, no capability of radio transmission and reception	/shared-cell/shared-cell-config/shared-cell-copy-combine-mode/shared-cell-copy-entities/shared-cell-copy-uplane-config and /shared-cell/shared-cell-config/shared-cell-copy-combine-mode/shared-cell-copy-entities/shared-cell-combine-uplane-config
23		SELECTIVE-BEAM-ID	FHM supports the selective combining function by using beamId	/shared-cell/shared-cell-config/shared-cell-copy-entities-selective-beam-id and /shared-cell/shared-cell-config/shared-cell-combine-entities-for-selective-beam-id
24	"urn:o-ran:sync:x.y "	ANTI-JAM	GNSS Anti Jamming	/sync/gnss-config/anti-jam-enable
25		GNSS	GNSS Support	Not applicable
26	"urn:o-ran:uplane-conf:x.y "	EAXC-GAIN-CORRECTION	eAxC specific gain correction	/user-plane-configuration /low-level-rx-endpoints/eaxc-gain-correction
27		TX-REFERENCE-LEVEL	TX gain reference level control	/user-plane-configuration/tx-array-carriers/reference-level
28	"urn:o-ran:wg4feat:1.0"	BEAM-UPDATE-CONTENTION-CONTROL	O-RU requirements for beam weight update for a given beamId, to avoid beam update contentions.	/user-plane-configuration /low-level-tx-endpoints/eaxc-gain-correction/beam-update-contention-control-enabled and /user-plane-configuration /low-level-rx-endpoints/eaxc-gain-correction/beam-update-contention-control-enabled
29		CHANNEL-INFORMATION-COMPRESSION	Compression for channel information in Section Type 6 (indicate at least static compression is supported)	/user-plane-configuration /low-level-tx-endpoints/channel-information-compressions
30		CHANNEL-INFORMATION-PRB-GROUP	Receiving and processing channel information with PRB group size greater than one	/user-plane-configuration /low-level-tx-endpoints/channel-information-prb-group-configuration
31		CPLANE-MESSAGE-PROCESSING-LIMITS	C-Plane Message Limits	/user-plane-configuration/low-level-tx-endpoints/cplane-message-processing-limits-enabled and /user-plane-configuration /low-level-rx-endpoints/cplane-message-processing-limits-enabled

		UPLANE-ONLY-DL-MODE	O-RU supports U-Plane-only DL mode.	/user-plane-configuration/general-config/uplane-only-dl-mode-enable
32		DYNAMIC-TRANSMISSION-WINDOW-CONTROL	U-plane transmission window control configuration over C-plane	/user-plane-configuration /low-level-rx-endpoints/transmission-window-control
33		EXT-ANT-DELAY-CONTROL	O-RU supports external antenna delay control	/user-plane-configuration/tx-array-carriers/t-da-offset and/or /user-plane-configuration/rx-array-carriers/t-au-offset
34		EXTENDED-PRACH-CONFIGURATION	O-RU is able to support extended number of PRACH patterns and occasions provided by means of static PRACH.	/user-plane-configuration/static-low-level-rx-endpoints/extended-max-prach-patterns /user-plane-configuration/static-prach-configurations/prach-patterns/prach-pattern-id /user-plane-configuration/static-prach-configurations/prach-patterns/number-of-occasions /user-plane-configuration/static-prach-configurations/prach-patterns/occasion-parameters/occasion-id
35		ENHANCED-T-COMBINE	O-RU can support t-combine-net and tx-duration	/shared-cell/shared-cell-config/enhanced-t-combine-enabled
36		INDEPENDENT-TRANSMISSION-WINDOW-CONTROL	Independent U-plane transmission window per endpoint	M-Plane activation not applicable
37		INTEGRITY-CHECK-AT-SW-DOWNLOAD	O-RU can perform integrity check at file download	/software-inventory/integrity-check-at-download-enabled
38		MULTIPLE-SCS-IN-EAXC	FHM supports combining for multiple SCS or multiple c-plane-types/frameStructure in a single eAxC-id	/shared-cell/shared-cell-config/multiple-scs-in-eaxc-used
39		MULTIPLE-TRANSPORT-SESSION-TYPE	Multiple transport-session-type	/processing-elements/additional-transport-session-type-elements
40		NON-PERSISTENT-MPLANE	Optimizations for non-persistent M-Plane	O-RU is configured with an event-collector identity (e.g., using DHCP)
41		NON-SCHEDULED-UEID	O-RU endpoint is able to support non-scheduled -ueid	/user-plane-configuration / low-level-tx-endpoints/ non-scheduled-ueid-enabled and/or /user-plane-configuration / low-level-rx-endpoints/ non-scheduled-ueid-enabled
42		ORDERED-TRANSMISSION	Ordered transmission	/user-plane-configuration /low-level-rx-endpoints/ordered-transmission
43		SE11-WITH-CONTINUITY-BIT-SUPPORT	O-RU is able to support handling 'continuity' bit information in Section Extension 11	/user-plane-configuration/low-level-tx-endpoints/se-11-continuity-flag-enabled
44		SHARED-ORU-MULTI-ODU	O-RU is able to support supervision on a per O-DU basis	/supervision/per-odu-monitoring
45		SHARED-ORU-MULTI-OPERATOR	O-RU is able to support operation with multiple shared resource operator O-DUs	/users/user/sro-id

46		STATIC-TRANSMISSION-WINDOW-CONTROL	U-plane transmission window control configuration over M-plane	/user-plane-configuration /low-level-rx-endpoints/transmission-window-control
47		SUPERVISION-WITH-SESSION-ID	O-RU uses the NETCONF session-id in supervision-notification	Not applicable
48		UNIFORMLY-DISTRIBUTED-TRANSMISSION	Transmission of UL U-plane messages distributed uniformly over transmission window	/user-plane-configuration /low-level-rx-endpoints/transmission-type
49		UPLANE-MESSAGE-PROCESSING-LIMITS	U-Plane message limits	/user-plane-configuration/general-config/ uplane-message-section-header-limit-enabled

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.

Table C.2-2: Optional feature support in common models

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

C.3 Optional features exposed using O-RAN YANG models

In addition to optional namespaces and optional YANG feature tags specified in O-RU supported namespaces, certain O-RAN defined YANG models include read-only YANG leaf nodes used to be able to indicate support by an O-RU of for certain optional capabilities by the O-RU.

Table C.3-1: Optional features where support is indicated by read-only YANG leaf nodes defined in O-RAN defined YANG models

	Namespace	Optional Feature	Read-only YANG leaf indicating feature support	Optional Feature Control
1	"urn:o-ran:ald-port:x.y "	Control of ALD's DC power supply	/ald-ports-io/ald-port/dc-control-support	/ald-ports-io/ald-port/dc-enabled-status

	Namespace	Optional Feature	Read-only YANG leaf indicating feature support	Optional Feature Control
2	"urn:o-ran:antcal:x.y"	O-RU is able to perform self calibration	/antenna-calibration/antenna-calibration-capabilities/self-calibration-support	/antenna-calibration/self-calibration-policy/self-calibration-allowed
3	"urn:o-ran:antcal:x.y"	O-RU supports configuration of the preparedness timer that controls how far in advance of the co-ordinated self calibration procedure the O-RU is required to send the notification of impacted resources	/antenna-calibration/antenna-calibration-capabilities/configured-preparation-timer-supported	Clause 15.5 defines use when supported.
4	"urn:o-ran:beamforming:x.y "	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	/beamforming-config/operational-properties/update-bf-non-delete	**activate-beamforming-config** RPC or activate-beamforming-config-by-capability-group RPC
5	"urn:o-ran:beamforming:x.y "	O-RU supports the capability to store the modified beamforming configuration file in the reset persistent memory	/beamforming-config/operational-properties/persistent-bf-files	M-Plane activation not applicable
6	"urn:o-ran:beamforming:x.y"	O-RU supports dynamic beamforming control mode	/beamforming-config/static-properties/rt-bf-weights-update-support	M-Plane activation not applicable. CUS-Plane [2] clause 7.7.1.2 defines use when supported.
7	"urn:o-ran:beamforming:x.y"	O-RU supports attributes based dynamic beamforming control mode	beamforming-config/beamforming-trough-attributes-supported	M-Plane activation not applicable. CUS-Plane [2] clause 7.7.2.1 defines use when supported.
8	"urn:o-ran:beamforming:x.y"	O-RU supports beamforming based on UE channel information	beamforming-config /beamforming-trough-ue-channel-info-supported	M-Plane activation not applicable. CUS-Plane [2] clause 7.2.7 defines use when supported.
9	"urn:o-ran:beamforming:x.y"	O-RU supports dynamic channel information compression	/beamforming-config/ue-specific-beamforming/dynamic-channel-information-compression-supported	M-Plane activation not applicable CUS-Plane [2] clause 7.5.2.15 defines use when supported.
10	"urn:o-ran:message5:x.y"	O-RU supports eCPRI message 5 one-step procedure for T34 measurements.	/ecpri-delay-message/one-step-t34-supported	M-Plane activation not applicable CUS-Plane [2] clause 4.4.4.4 defines use when supported.
11	"urn:o-ran:message5:x.y"	O-RU supports eCPRI message 5 two-step procedure for T34 measurements.	/ecpri-delay-message/two-step-t34-supported	M-Plane activation not applicable CUS-Plane [2] clause 4.4.4.4 defines use when supported.
12	"urn:o-ran:hardware:x.y"	O-RU supports the dying gasp alarm	/hardware/component/dying-gasp-supported	M-Plane activation not applicable
13	"urn:o-ran:interfaces:x.y"	O-RU's rate able to be supported by an interface is less than nominal bit rate indicated by its transceiver model	/interfaces/interface-grouping	M-Plane activation not applicable

	Namespace	Optional Feature	Read-only YANG leaf indicating feature support	Optional Feature Control
14	"urn:o-ran:module-cap:x.y"	O-RU supports Category B operation – precoding in the O-RU	/module-capability/ru-capabilities/ru-supported-category	M-Plane activation not applicable. CUS-Plane [2] clause 7.2.4 defines use when supported.
15	"urn:o-ran:module-cap:x.y"	O-RU supports dynamic compression method	/module-capability/ru-capabilities/format-of-iq-samples/dynamic-compression-supported	M-Plane activation not applicable. CUS-Plane[2] clause 7.5.2.10 defines use when supported.
16	"urn:o-ran:module-cap:x.y"	O-RU supports real-time variable bit with	/module-capability/ru-capabilities/format-of-iq-samples/realtime-variable-bit-width-supported	M-Plane activation not applicable. CUS-Plane[2] clause 7.5.2.10 defines use when supported.
17	"urn:o-ran:module-cap:x.y"	O-RU supports block scaling compression	O-RU sets a list entry /module-capability/ru-capabilities/format-of-iq-samples/compression-method-supported with compression-method set to "BLOCK_SCALING"	M-Plane activation not applicable. CUS-Plane[2] clause 8.3.3.13 defines use when supported.
18	"urn:o-ran:module-cap:x.y"	O-RU supports u-law compression	O-RU sets a list entry /module-capability/ru-capabilities/format-of-iq-samples/compression-method-supported with compression-method set to "U_LAW"	M-Plane activation not applicable. CUS-Plane[2] clause 8.3.3.13 defines use when supported.
19	"urn:o-ran:module-cap:x.y"	O-RU supports beamspace compression	O-RU sets a list entry /module-capability/ru-capabilities/format-of-iq-samples/compression-method-supported with compression-method set to "BEAMSPACE"	M-Plane activation not applicable. CUS-Plane[2] clause 8.3.3.13 defines use when supported.
20	"urn:o-ran:module-cap:x.y"	O-RU supports modulation compression	O-RU sets a list entry /module-capability/ru-capabilities/format-of-iq-samples/compression-method-supported with compression-method set to "MODULATION"	M-Plane activation not applicable. CUS-Plane[2] clause 8.3.3.13 defines use when supported.
21	"urn:o-ran:module-cap:x.y"	O-RU supports block floating-point with selective re sending compression	O-RU sets a list entry /module-capability/ru-capabilities/format-of-iq-samples/compression-method-supported with compression-method set to "BLOCK-FLOATING-POINT-SELECTIVE-RE-SENDING"	M-Plane activation not applicable. CUS-Plane[2] clause 8.3.3.13 defines use when supported.
22	"urn:o-ran:module-cap:x.y"	O-RU supports modulation compression with selective re sending compression	O-RU sets a list entry /module-capability/ru-capabilities/format-of-iq-samples/compression-method-supported with compression-method set to "MODULATION-COMPRESSION-SELECTIVE-RE-SENDING"	M-Plane activation not applicable. CUS-Plane[2] clause 8.3.3.13 defines use when supported.

	Namespace	Optional Feature	Read-only YANG leaf indicating feature support	Optional Feature Control
23	"urn:o-ran:module-cap:x.y"	O-RU supports variable bit width per channel	/module-capability/ru-capabilities/format-of-iq-samples/variable-bit-width-per-channel-supported	M-Plane activation not applicable. CUS-Plane[2] clause 7.5.2.10 defines use when supported.
24	"urn:o-ran:module-cap:x.y"	O-RU supports symbol number increment command in a C-Plane	/module-capability/ru-capabilities/format-of-iq-samples/syminc-supported	M-Plane activation not applicable.
25	"urn:o-ran:module-cap:x.y"	O-RU supports regularizationFactor in section type 5	/module-capability/ru-capabilities/format-of-iq-samples/regularization-factor-se-supported	/user-plane-configuration/general-config/**regularization-factor-se-configured**
26	"urn:o-ran:module-cap:x.y"	O-RU supports little endian	/module-capability/ru-capabilities/format-of-iq-samples/little-endian-supported	/user-plane-configuration/general-config/**little-endian-byte-order**
27	"urn:o-ran:module-cap:x.y"	O-RU requires 4-byte aligned Section Type 6	/module-capability/ru-capabilities/st6-4byte-alignment-required	M-Plane activation not applicable. CUS-Plane[2] clause 7.3.1 defines use when supported.
28	"urn:o-ran:module-cap:x.y"	O-RU supports energy saving by transmission blanking	/module-capability/ru-capabilities/energy-saving-by-transmission-blanks	M-Plane activation not applicable. CUS-Plane[2] clause 8.4.2.3 defines use when supported.
29	"urn:o-ran:module-cap:x.y"	O-RU supports dynamic transport delay management through eCPRI Msg 5	/module-capability/ru-capabilities/dynamic-transport-delay-management-supported	M-Plane activation not applicable. CUS-Plane[2] clause 4.4.4.4 defines use when supported.
30	"urn:o-ran:module-cap:x.y"	O-RU expects unique eCPRI sequence id for eAxC_IDs serving for UL and DL for the same Component Carrier	/module-capability/ru-capabilities/support-only-unique-ecpri-seqid-per-eaxc	M-Plane activation not applicable. CUS-Plane[2] defines use when supported.
31	"urn:o-ran:module-cap:x.y"	O-RU supports coupling of C and U-plane messages by frequency and time	/module-capability/coupling-methods/coupling-via-frequency-and-time	M-Plane activation not applicable. CUS-Plane[2] clause 8.4.1 defines use when supported.
32	"urn:o-ran:module-cap:x.y"	O-RU supports coupling of C and U-plane messages by frequency and time with priorities	/module-capability/coupling-methods/coupling-via-frequency-and-time-with-priorities	M-Plane activation not applicable. CUS-Plane[2] clause 7.9.7 defines use when supported.
33	"urn:o-ran:module-cap:x.y"	O-RU supports coupling of C and U-plane messages by frequency and time with priorities optimized	/module-capability/coupling-methods/coupling-via-frequency-and-time-with-priorities-optimized	M-Plane activation not applicable. CUS-Plane[2] clause 7.9.8 defines use when supported.
34	"urn:o-ran:module-cap:x.y"	O-RU supports optional field udCompLen in U-Plane messages	/module-capability/ud-comp-len-supported	M-Plane activation not applicable. CUS-Plane[2] clause 8.3.3.19 defines use when supported.
35	"urn:o-ran:module-cap:x.y"	O-RU supports configuration of different t-da-offset on different tx-array-carriers, and different t-au-offset on different rx-array-carriers	/module-capability/ext-ant-delay-capability set to "PER-ARRAY-CARRIER"	M-Plane activation not applicable. CUS-Plane[2] clause 4.7 defines use when supported.

	Namespace	Optional Feature	Read-only YANG leaf indicating feature support	Optional Feature Control
36	"urn:o-ran:module-cap:x.y"	O-RU supports the configuration of different t-da-offset on different tx-array-carriers only when those tx-array-carriers belong to different tx-arrays	/module-capability/ext-ant-delay-capability set to "PER-ARRAY"	M-Plane activation not applicable. CUS-Plane[2] clause 4.7 defines use when supported.
37	"urn:o-ran:module-cap:x.y"	O-RU supports the configuration of a common t-da-offset across all tx-array-carriers	/module-capability/ext-ant-delay-capability set to "PER-O-RU"	M-Plane activation not applicable. CUS-Plane[2] clause 4.7 defines use when supported.
38	"urn:o-ran:module-cap:x.y"	O-RU supports sending NACK feedback if a section extension for ACK/NACK request is received	/module-capability/nack-supported	M-Plane activation not applicable. CUS-Plane[2] defines use when supported.
39	"urn:o-ran:module-cap:x.y"	O-RU local management of the LAA contention window	/module-capability/band-capabilities/sub-band-info/self-configure	M-Plane activation not applicable.
40	"urn:o-ran:operations:x.y "	IEEE 1914.3 header support	/operational-info/declarations/supported-header-mechanism/protocol	M-Plane activation not applicable. CUS-Plane [2] clause 5.1.3.3 defines use when supported.
41	"urn:o-ran:operations:x.y "	eCPRI Concatenation support	/operational-info/declarations/supported-header-mechanism/ecpri-concatenation-support	M-Plane activation not applicable. CUS-Plane [2] clause 5.1.3.2.4 defines use when supported.
42	"urn:o-ran:performance-management:x.y "	O-RU supports RX power transceiver measurement	/performance-management-objects/measurement-capabilities/transceiver-objects/measurement-object set to "RX_POWER"	/performance-measurement/objects/transceiver-measurement-objects
43	"urn:o-ran:performance-management:x.y "	O-RU supports TX power transceiver measurement	/performance-management-objects/measurement-capabilities/transceiver-objects/measurement-object set to "TX_POWER"	/performance-measurement/objects/transceiver-measurement-objects
44	"urn:o-ran:performance-management:x.y "	O-RU supports TX Bias Current transceiver measurement	/performance-management-objects/measurement-capabilities/transceiver-objects/measurement-object set to "TX_BIAS_COUNT"	/performance-measurement/objects/transceiver-measurement-objects
45	"urn:o-ran:performance-management:x.y "	O-RU supports transceiver voltage measurement	/performance-management-objects/measurement-capabilities/transceiver-objects/measurement-object set to "VOLTAGE"	/performance-measurement/objects/transceiver-measurement-objects
46	"urn:o-ran:performance-management:x.y "	O-RU supports transceiver temperature measurement	/performance-management-objects/measurement-capabilities/transceiver-objects/measurement-object set to "TEMPERATURE"	/performance-measurement/objects/transceiver-measurement-objects

	Namespace	Optional Feature	Read-only YANG leaf indicating feature support	Optional Feature Control
47	"urn:o-ran:performance-management: x.y "	O-RU supports RX Window on Time measurement	/performance-management-objects/measurement-capabilities/rx-window-objects/measurement-object set to "RX_ON_TIME"	/performance-measurement/objects/rx-window-measurement-objects
48	"urn:o-ran:performance-management: x.y "	O-RU supports RX Window too early Time measurement	/performance-management-objects/measurement-capabilities/rx-window-objects/measurement-object set to "RX_EARLY"	/performance-measurement/objects/rx-window-measurement-objects
49	"urn:o-ran:performance-management: x.y "	O-RU supports RX Window too late Time measurement	/performance-management-objects/measurement-capabilities/rx-window-objects/measurement-object set to "RX_LATE"	/performance-measurement/objects/rx-window-measurement-objects
50	"urn:o-ran:performance-management: x.y "	O-RU supports RX Window corrupt measurement	/performance-management-objects/measurement-capabilities/rx-window-objects/measurement-object set to "RX_CORRUPT"	/performance-measurement/objects/rx-window-measurement-objects
51	"urn:o-ran:performance-management: x.y "	O-RU supports RX Window duplicate measurement	/performance-management-objects/measurement-capabilities/rx-window-objects/measurement-object set to RX_DUPL	/performance-measurement/objects/rx-window-measurement-objects
52	"urn:o-ran:performance-management: x.y "	O-RU supports RX Window on Time measurement for control packets	/performance-management-objects/measurement-capabilities/rx-window-objects/measurement-object set to "RX_ON_TIME_C"	/performance-measurement/objects/rx-window-measurement-objects
53	"urn:o-ran:performance-management: x.y "	O-RU supports RX Window too early Time measurement for control packets	/performance-management-objects/measurement-capabilities/rx-window-objects/measurement-object set to "RX_EARLY_C"	/performance-measurement/objects/rx-window-measurement-objects
54	"urn:o-ran:performance-management: x.y "	O-RU supports RX Window too late Time measurement for control packets	/performance-management-objects/measurement-capabilities/rx-window-objects/measurement-object set to "RX_LATE_C"	/performance-measurement/objects/rx-window-measurement-objects
55	"urn:o-ran:performance-management: x.y "	O-RU supports RX Window sequence error measurements	/performance-management-objects/measurement-capabilities/rx-window-objects/measurement-object set to "RX_SEQID_ERR"	/performance-measurement/objects/rx-window-measurement-objects

	Namespace	Optional Feature	Read-only YANG leaf indicating feature support	Optional Feature Control
56	"urn:o-ran:performance-management: x.y "	O-RU supports RX Window sequence error measurements for control packets	/performance-management-objects/measurement-capabilities/rx-window-objects/measurement-object set to "RX_SEQID_ERR_C"	/performance-measurement/objects/rx-window-measurement-objects
57	"urn:o-ran:performance-management: x.y "	O-RU supports RX Window packet drop measurements	/performance-management-objects/measurement-capabilities/rx-window-objects/measurement-object set to "RX _ERR_DROP"	/performance-measurement/objects/rx-window-measurement-objects
58	"urn:o-ran:performance-management: x.y "	O-RU supports TX Window measurements for all outbound packets	/performance-management-objects/measurement-capabilities/tx-stats-objects/measurement-object set to "TX_TOTAL"	/performance-measurement/objects/tx-measurement-objects
59	"urn:o-ran:performance-management: x.y "	O-RU supports TX Window measurements for outbound controlpackets	/performance-management-objects/measurement-capabilities/tx-stats-objects/measurement-object set to "TX_TOTAL_C"	/performance-measurement/objects/tx-measurement-objects
60	"urn:o-ran:performance-management: x.y "	O-RU supports temperature measurements of specific hardware components	/performance-management-objects/measurement-capabilities/epe-stats-objects/measurement-object set to "TEMPERATURE"	/performance-measurement/objects/epe-measurement-objects
61	"urn:o-ran:performance-management: x.y "	O-RU supports power measurements of specific hardware components	/performance-management-objects/measurement-capabilities/epe-stats-objects/measurement-object set to "POWER"	/performance-measurement/objects/epe-measurement-objects
62	"urn:o-ran:performance-management: x.y "	O-RU supports voltage measurements of specific hardware components	/performance-management-objects/measurement-capabilities/epe-stats-objects/measurement-object set to "VOLTAGE"	/performance-measurement/objects/epe-measurement-objects
63	"urn:o-ran:performance-management: x.y "	O-RU supports current measurements of specific hardware components	/performance-management-objects/measurement-capabilities/epe-stats-objects/measurement-object set to "CURRENT"	/performance-measurement/objects/epe-measurement-objects
64	"urn:o-ran:shared-cell:x.y"	O-RU supports multi cell operation in shared cell cascade mode	/shared-cell/shared-cell-module-cap/multi-cell-in-cascade-mode-supported	M-Plane activation not applicable. CUS-Plane[2] clause 13.4 defines use when supported.
65	"urn:o-ran:software-management: x.y"	O-RU requires separate download procedures to be re-used for downloading individual files in a software build, instead of a single archived package	/software-inventory/build-content-download	software-download rpc

	Namespace	Optional Feature	Read-only YANG leaf indicating feature support	Optional Feature Control
66	"urn:o-ran:sync:x.y"	O-RU supports PTP	O-RU includes "PTP" enumeration in /sync/:sync-status/supported-reference-type list item	/sync/ptp-config
67	"urn:o-ran:sync:x.y"	O-RU supports Synchronous Ethernet	O-RU includes "SYNCE" enumeration in /sync/:sync-status/supported-reference-type list item	/sync/synce-config
68	"urn:o-ran:sync:x.y"	O-RU supports GNSS	O-RU includes "GNSS" enumeration in /sync/:sync-status/supported-reference-type list item	/sync/gnss-config
69	"urn:o-ran:sync:x.y"	O-RU supports the T-BC profiles in ITU-T G.8275.1	/sync/sync-capability/boundary-clock-supported	M-Plane activation not applicable.
70	"urn:o-ran:sync:x.y"	O-RU supports enhanced accuracy for sync as per IEEE802.1CM clause 6.4.1	O-RU sets /sync/sync-capability/sync-t-tsc to **"**ENHANCED**"**	M-Plane activation not applicable.
71	"urn:o-ran:uplane-conf:x.y"	Section types supported by the O-RU endpoint	/user-plane-configuration/endpoint-types/supported-section-types/section-type	M-Plane activation not applicable. CUS-Plane[2] clause 7.3.1 defines use when supported.
72	"urn:o-ran:uplane-conf:x.y"	Section type extensions supported by the O-RU endpoint	/user-plane-configuration/endpoint-types/supported-section-types/supported-section-extensions	M-Plane activation not applicable. CUS-Plane[2] clause 7.3.2 defines use when supported.
73	"urn:o-ran:uplane-conf:x.y"	Supported frame structures	/user-plane-configuration/endpoint-types/supported-frame-structures	M-Plane activation not applicable. CUS-Plane[2] clause 7.5.2.13 defines use when supported.
74	"urn:o-ran:uplane-conf:x.y"	O-RU endpoint supports time managed delays	/user-plane-configuration/endpoint-types/managed-delay-support set to "MANAGED" or "BOTH"	/user-plane-configuration/low-level-rx-endpoints/non-time-managed-delay-enabled
75	"urn:o-ran:uplane-conf:x.y"	O-RU endpoint supports multiple numerologies	/user-plane-configuration/endpoint-types/multiple-numerology-supported set to "true"	M-Plane activation not applicable. CUS-Plane[2] table 7.4-3 defines use when supported.
76	"urn:o-ran:uplane-conf:x.y"	O-RU endpoint supports non-scheduled-ueid	/user-plane-configuration/endpoint-type/non-scheduled-ueid-supported set to "true"	/user-plane-configuration / low-level-tx-endpoints/ non-scheduled-ueid-enabled and/or /user-plane-configuration / low-level-rx-endpoints/ non-scheduled-ueid-enabled

Annex D (informative): YANG module graphical representation

D.1 Introduction

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 [i.4].

D.2 System folder

D.2.1 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
     +--rw per-odu-monitoring! {or-feat:SHARED-ORU-MULTI-ODU or or-feat:SHARED-ORU-MULTI-OPERATOR}?
        +--rw odu-ids* [odu-id] {or-feat:SHARED-ORU-MULTI-ODU}?
        |  +--rw odu-id    string
        +--rw sro-ids-and-odu-ids* [odu-id sro-id] {or-feat:SHARED-ORU-MULTI-ODU and or-feat:SHARED-
ORU-MULTI-OPERATOR}?
           +--rw odu-id    string
           +--rw sro-id    string

  rpcs:
    +---x supervision-watchdog-reset
       +---w input
       |  +---w supervision-notification-interval?   uint16
       |  +---w guard-timer-overhead?                uint16
       |  +---w context {or-feat:SHARED-ORU-MULTI-ODU or or-feat:SHARED-ORU-MULTI-OPERATOR}?
       |     +---w odu-id?   string {or-feat:SHARED-ORU-MULTI-ODU}?
       |     +---w sro-id?   -> /or-user:users/user/sro-id {or-feat:SHARED-ORU-MULTI-OPERATOR}?
       +--ro output
          +--ro next-update-at?   yang:date-and-time
          +--ro error-message?    string

  notifications:
    +---n supervision-notification
       +--ro session-id    -> /ncm:netconf-state/sessions/session/session-id {or-feat:SUPERVISION-
WITH-SESSION-ID}?
```

D.2.2 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
        +--rw sro-id*         string {feat:SHARED-ORU-MULTI-OPERATOR}?

  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.2.3 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
    +--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/hw:component:
    +--ro connector-label?   string
  augment /hw:hardware-state-oper-enabled:
    +--ro availability-state?   -> /hw:hardware/component/state/o-ran-hw:availability-state
{hw:hardware-state}?
  augment /hw:hardware-state-oper-disabled:
    +--ro availability-state?   -> /hw:hardware/component/state/o-ran-hw:availability-state
{hw:hardware-state}?
```

D.2.4 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.2.5 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 identifier?   yang:xpath1.0
  |     +--ro fault-severity            enumeration
  |     +--ro is-cleared                boolean
  |     +--ro fault-text?               string
  |     +--ro probable-cause?           string
  |     +--ro specific-problem?         string
  |     +--ro proposed-repair-actions?   string
  |     +--ro event-time                yang:date-and-time
  |     +--ro additional-information* [identifier]
  |     |  +--ro identifier     string
  |     |  +--ro information?   string
  |     +--ro alarm-type?               enumeration
  +--rw historical-alarm-list {HISTORICAL-ALARM-LIST}?
     +--ro historical-alarms* []
        +--ro fault-id                  uint16
        +--ro fault-source              string
        +--ro affected-objects* []
        |  +--ro name    string
        |  +--ro identifier?   yang:xpath1.0
        +--ro fault-severity            enumeration
        +--ro is-cleared                boolean
        +--ro fault-text?               string
        +--ro probable-cause?           string
        +--ro specific-problem?         string
```

```
        +--ro proposed-repair-actions?   string
        +--ro event-time                 yang:date-and-time
        +--ro additional-information* [identifier]
        |  +--ro identifier     string
        |  +--ro information?   string
        +--ro alarm-type?                enumeration

  notifications:
    +---n alarm-notif
       +--ro fault-id                   uint16
       +--ro fault-source               string
       +--ro affected-objects* []
       |  +--ro name    string
       |  +--ro identifier?   yang:xpath1.0
       +--ro fault-severity             enumeration
       +--ro is-cleared                 boolean
       +--ro fault-text?                string
       +--ro probable-cause?            string
       +--ro specific-problem?          string
       +--ro proposed-repair-actions?   string
       +--ro event-time                 yang:date-and-time
       +--ro additional-information* [identifier]
       |  +--ro identifier     string
       |  +--ro information?   string
       +--ro alarm-type?                enumeration
```

D.2.6 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:uri
```

D.2.7 o-ran-certificates.yang module

The format for the certificates module is provided below.

```
module: o-ran-certificates
  +--rw certificate-parameters
     +--rw cert-maps
        +--rw cert-to-name* [id]
           +--rw id             uint32
           +--rw fingerprint    x509c2n:tls-fingerprint
           +--rw map-type       identityref
           +--rw name           string
```

D.3 Operations folder

D.3.1 o-ran-operations.yang module

The format for the operations module is provided below.

```
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 maximum-simultaneous-netconf-sessions?       uint8
     +--ro operational-state
     |  +--ro restart-cause?      enumeration
     |  +--ro restart-datetime?   yang:date-and-time
     |  +--ro current-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
     +--rw max-call-home-attempts?      uint8

  rpcs:
    +---x reset
```

D.3.2 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]
    |  |  |  |        +---w algorithm     asymmetric-key-algorithm-ref
    |  |  |  |        +---w public-key?   binary
    |  |  |  +--:(certificate)
    |  |  |     +---w certificate!
    |  |  +---w application-layer-credential
    |  |     +---w appl-password?   string
    |  +--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]
       |  |  |        +---w algorithm     asymmetric-key-algorithm-ref
       |  |  |        +---w public-key?   binary
       |  |  +--:(certificate)
       |  |     +---w certificate!
       |  +---w application-layer-credential
       |     +---w appl-password?   string
       +--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
    |  +--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.3.3 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
     +--ro build-content-download?   Empty
     +--ro integrity-check-at-download-enabled?   empty {or-feat:INTEGRITY-CHECK-AT-SW-DOWNLOAD}?

  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]
    |  |  |  |        +---w algorithm     asymmetric-key-algorithm-ref
    |  |  |  |        +---w public-key?   binary
    |  |  |  +--:(certificate)
    |  |  |     +---w certificate!
    |  |  +---w application-layer-credential
    |  |     +---w appl-password?   string
    |  +--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
    |     +--ro sw-install-timeout?   int32
    +---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
```

D.3.4 o-ran-lbm.yang module

The format for the (Ethernet) loopback 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.3.5 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.3.6 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
     +--ro one-step-t34-supported?   boolean
     +--ro two-step-t34-supported?   boolean
     +--rw message5-sessions
        +--rw session-parameters* [session-id]
           +--rw session-id                 uint32
           +--rw processing-element-name?   -> /element:processing-elements/ru-elements/name
           +--rw transport-session-type?                enumeration {feat:MULTIPLE-TRANSPORT-
SESSION-TYPE}?
           +--rw transport-qualified-processing-element-name?   -> /element:processing-
elements/additional-transport-session-type-elements[element:transport-session-type = current()/../
           +--ro flow-state
              +--ro responses-transmitted?   uint32
              +--ro requests-transmitted?    uint32
              +--ro followups-transmitted?   uint32
```

D.3.7 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]
     |  |  +--ro measurement-object    -> /performance-measurement-objects/epe-measurement-
objects/measurement-object
     |  |  +--ro component-class*      identityref
     |  +--ro symbol-rssi-stats-objects* [measurement-object]
```

```
     |     +--ro measurement-object    -> /performance-measurement-objects/symbol-rssi-measurement-
objects/measurement-object
     x--rw enable-SFTP-upload?                 boolean
     +--rw enable-file-upload?                 boolean
     +--rw enable-random-file-upload?          boolean
     x--rw remote-SFTP-uploads* [remote-SFTP-upload-path]
     |  +--rw remote-SFTP-upload-path    inet:uri
     |  +--rw (credentials)?
     |  |  +--:(password)
     |  |  |  +--rw password!
     |  |  |  |  +--rw password    string
     |  |  |  +--rw server
     |  |  |     +--rw keys* [algorithm]
     |  |  |        +--rw algorithm     asymmetric-key-algorithm-ref
     |  |  |        +--rw public-key?   binary
     |  |  +--:(certificate)
     |  |     +--rw certificate!
     |  +--rw application-layer-credential
     |     +--rw appl-password?   string
     +--rw remote-file-uploads* [remote-file-upload-path]
     |  +--rw remote-file-upload-path    inet:uri
     |  +--rw (credentials)?
     |  |  +--:(password)
     |  |  |  +--rw password!
     |  |  |  |  +--rw password    string
     |  |  |  +--rw server
     |  |  |     +--rw keys* [algorithm]
     |  |  |        +--rw algorithm     asymmetric-key-algorithm-ref
     |  |  |        +--rw public-key?   binary
     |  |  +--:(certificate)
     |  |     +--rw certificate!
     |  +--rw application-layer-credential
     |     +--rw appl-password?   string
     +--rw transceiver-measurement-interval?   uint16
     +--rw epe-measurement-interval?           uint16
     +--rw rx-window-measurement-interval?     uint16
     +--rw tx-measurement-interval?            uint16
     +--rw symbol-rssi-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
     |     x--ro frequeny-table*        uint32
     |     +--ro frequency-bin-table* [bin-id]
     |        +--ro bin-id    uint32
     |        +--ro value?    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 transport-session-type?     enumeration {feat:MULTIPLE-TRANSPORT-SESSION-
TYPE}?
     |     |     +--ro transport-qualified-name?   -> /o-ran-elements:processing-
elements/additional-transport-session-type-elements[o-ran-elements:transport-session-type =
current()/../transport-session-type]/ru-elements/name {feat:MULTIPLE-TRANSPORT-SESSION-TYPE}?
     |     |     +--ro count                      uint64
     |     +--:(EAXC_ID)
     |        +--ro eaxc-measured-result* []
     |           +--ro eaxc-id?                   uint16
     |           +--ro count                      uint64
     |           +--ro data-direction?            enumeration
     |           +--ro transport-name?            -> /o-ran-elements:processing-elements/ru-
elements/name
     |           +--ro transport-session-type?     enumeration {feat:MULTIPLE-TRANSPORT-SESSION-
TYPE}?
     |           +--ro transport-qualified-name?   -> /o-ran-elements:processing-
elements/additional-transport-session-type-elements[o-ran-elements:transport-session-type =
current()/../transport-session-type]/ru-elements/name {feat:MULTIPLE-TRANSPORT-SESSION-TYPE}?
     +--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 transport-session-type?     enumeration {feat:MULTIPLE-TRANSPORT-SESSION-
TYPE}?
     |     |     +--ro transport-qualified-name?   -> /o-ran-elements:processing-
elements/additional-transport-session-type-elements[o-ran-elements:transport-session-type =
current()/../transport-session-type]/ru-elements/name {feat:MULTIPLE-TRANSPORT-SESSION-TYPE}?
     |     |     +--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 transport-session-type?     enumeration {feat:MULTIPLE-TRANSPORT-SESSION-
TYPE}?
     |           +--ro transport-qualified-name?   -> /o-ran-elements:processing-
elements/additional-transport-session-type-elements[o-ran-elements:transport-session-type =
current()/../transport-session-type]/ru-elements/name {feat:MULTIPLE-TRANSPORT-SESSION-TYPE}?
     +--rw epe-measurement-objects* [measurement-object]
     |  +--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
     +--rw symbol-rssi-measurement-objects* [measurement-object]
        +--rw measurement-object                    enumeration
        +--rw object-unit                           enumeration
        +--rw per-rx-array-carrier-configuration* [rx-array-carrier]
        |  +--rw rx-array-carrier    -> /up:user-plane-configuration/rx-array-carriers/name
        |  +--rw period?             uint16
        |  +--rw symbol-index*       uint16
        |  +--rw active?             boolean
        |  +--rw report-info*        enumeration
        |  +--rw bin-count?          uint16
        |  +--rw lower-bound?        int16
        |  +--rw upper-bound?        int16
        +--ro symbol-rssi-measurement-result* [object-unit-id]
           +--ro object-unit-id             -> /up:user-plane-configuration/rx-array-carriers/name
           +--ro per-symbol-index-result* [symbol-index]
              +--ro symbol-index           uint16
              +--ro min
```

```
             |  +--ro value?   decimal64
             +--ro max
             |  +--ro value?   decimal64
             +--ro avg
             |  +--ro value?   decimal64
             x--ro frequency-table*       uint32
             +--ro frequency-bin-table* [bin-id]
                +--ro bin-id    uint32
                +--ro value?    uint32

  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
       |  |  x--ro frequeny-table*        uint32
       |  |  +--ro frequency-bin-table* [bin-id]
       |  |     +--ro bin-id    uint32
       |  |     +--ro value?    uint32
       |  +--ro multiple-transceiver-measurement-result* []
       |     +--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
       |        x--ro frequeny-table*        uint32
       |        +--ro frequency-bin-table* [bin-id]
       |           +--ro bin-id    uint32
       |           +--ro value?    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 transport-session-type?     enumeration {feat:MULTIPLE-TRANSPORT-SESSION-
TYPE}?
       |  |  |     +--ro transport-qualified-name?   -> /o-ran-elements:processing-
elements/additional-transport-session-type-elements[o-ran-elements:transport-session-type =
current()/../transport-session-type]/ru-elements/name {feat:MULTIPLE-TRANSPORT-SESSION-TYPE}?
       |  |  |     +--ro count                       uint64
       |  |  +--:(EAXC_ID)
```

```
       |  |     +--ro eaxc-measured-result* []
       |  |        +--ro eaxc-id?                    uint16
       |  |        +--ro count                       uint64
       |  |        +--ro data-direction?             enumeration
       |  |        +--ro transport-name?             -> /o-ran-elements:processing-elements/ru-
elements/name
       |  |        +--ro transport-session-type?     enumeration {feat:MULTIPLE-TRANSPORT-SESSION-
TYPE}?
       |  |        +--ro transport-qualified-name?   -> /o-ran-elements:processing-
elements/additional-transport-session-type-elements[o-ran-elements:transport-session-type =
current()/../transport-session-type]/ru-elements/name {feat:MULTIPLE-TRANSPORT-SESSION-TYPE}?
       |  +--ro multiple-rx-window-measurement-result* []
       |     +--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 transport-session-type?     enumeration {feat:MULTIPLE-TRANSPORT-
SESSION-TYPE}?
       |        |     +--ro transport-qualified-name?   -> /o-ran-elements:processing-
elements/additional-transport-session-type-elements[o-ran-elements:transport-session-type =
current()/../transport-session-type]/ru-elements/name {feat:MULTIPLE-TRANSPORT-SESSION-TYPE}?
       |        |     +--ro count                      uint64
       |        +--:(EAXC_ID)
       |           +--ro eaxc-measured-result* []
       |              +--ro eaxc-id?                   uint16
       |              +--ro count                      uint64
       |              +--ro data-direction?            enumeration
       |              +--ro transport-name?            -> /o-ran-elements:processing-elements/ru-
elements/name
       |              +--ro transport-session-type?     enumeration {feat:MULTIPLE-TRANSPORT-
SESSION-TYPE}?
       |              +--ro transport-qualified-name?   -> /o-ran-elements:processing-
elements/additional-transport-session-type-elements[o-ran-elements:transport-session-type =
current()/../transport-session-type]/ru-elements/name {feat:MULTIPLE-TRANSPORT-SESSION-TYPE}?
       +--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 transport-session-type?     enumeration {feat:MULTIPLE-TRANSPORT-SESSION-
TYPE}?
       |  |  |     +--ro transport-qualified-name?   -> /o-ran-elements:processing-
elements/additional-transport-session-type-elements[o-ran-elements:transport-session-type =
current()/../transport-session-type]/ru-elements/name {feat:MULTIPLE-TRANSPORT-SESSION-TYPE}?
       |  |  |     +--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 transport-session-type?     enumeration {feat:MULTIPLE-TRANSPORT-SESSION-
TYPE}?
       |  |        +--ro transport-qualified-name?   -> /o-ran-elements:processing-
elements/additional-transport-session-type-elements[o-ran-elements:transport-session-type =
current()/../transport-session-type]/ru-elements/name {feat:MULTIPLE-TRANSPORT-SESSION-TYPE}?
       |  +--ro multiple-tx-measurement-result* []
       |     +--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 transport-session-type?     enumeration {feat:MULTIPLE-TRANSPORT-
SESSION-TYPE}?
       |        |     +--ro transport-qualified-name?   -> /o-ran-elements:processing-
elements/additional-transport-session-type-elements[o-ran-elements:transport-session-type =
current()/../transport-session-type]/ru-elements/name {feat:MULTIPLE-TRANSPORT-SESSION-TYPE}?
       |        |     +--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 transport-session-type?     enumeration {feat:MULTIPLE-TRANSPORT-
SESSION-TYPE}?
       |              +--ro transport-qualified-name?   -> /o-ran-elements:processing-
elements/additional-transport-session-type-elements[o-ran-elements:transport-session-type =
current()/../transport-session-type]/ru-elements/name {feat:MULTIPLE-TRANSPORT-SESSION-TYPE}?
       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
       +--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
       |  +--ro multiple-epe-measurement-result* []
       |     +--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
       +--ro symbol-rssi-stats* [measurement-object]
          +--ro measurement-object                         -> /performance-measurement-
objects/symbol-rssi-measurement-objects/measurement-object
          +--ro start-time?                                yang-types:date-and-time
          +--ro end-time?                                  yang-types:date-and-time
          +--ro symbol-rssi-measurement-result* [object-unit-id]
          |  +--ro object-unit-id             -> /up:user-plane-configuration/rx-array-carriers/name
          |  +--ro per-symbol-index-result* [symbol-index]
          |     +--ro symbol-index           uint16
          |     +--ro min
          |     |  +--ro value?   decimal64
          |     +--ro max
          |     |  +--ro value?   decimal64
          |     +--ro avg
          |     |  +--ro value?   decimal64
          |     x--ro frequency-table*       uint32
          |     +--ro frequency-bin-table* [bin-id]
          |        +--ro bin-id    uint32
          |        +--ro value?    uint32
          +--ro multiple-symbol-rssi-measurement-result* []
             +--ro start-time?                      yang-types:date-and-time
             +--ro end-time?                        yang-types:date-and-time
             +--ro symbol-rssi-measurement-result* [object-unit-id]
                +--ro object-unit-id             -> /up:user-plane-configuration/rx-array-
carriers/name
                +--ro per-symbol-index-result* [symbol-index]
                   +--ro symbol-index           uint16
                   +--ro min
                   |  +--ro value?   decimal64
                   +--ro max
                   |  +--ro value?   decimal64
                   +--ro avg
```

```
                  |  +--ro value?   decimal64
                  x--ro frequency-table*       uint32
                  +--ro frequency-bin-table* [bin-id]
                     +--ro bin-id    uint32
                     +--ro value?    uint32
```

D.3.8 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 sro-id?                                   -> /or-user:users/user/sro-id {feat:SHARED-
ORU-MULTI-OPERATOR}?
     |  +--rw processing-element                        -> /o-ran-pe:processing-elements/ru-
elements/name
     |  +--rw transport-session-type?                   enumeration {feat:MULTIPLE-TRANSPORT-
SESSION-TYPE}?
     |  +--rw transport-qualified-processing-element?   -> /o-ran-pe:processing-elements/additional-
transport-session-type-elements[o-ran-pe:transport-session-type = current()/../transport-session-
type]/ru-elements/name {feat:MULTIPLE-TRANSPORT-SESSION-TYPE}?
     |  +--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 sro-id?                                   -> /or-user:users/user/sro-id {feat:SHARED-
ORU-MULTI-OPERATOR}?
     |  +--rw processing-element                        -> /o-ran-pe:processing-elements/ru-
elements/name
     |  +--rw transport-session-type?                   enumeration {feat:MULTIPLE-TRANSPORT-
SESSION-TYPE}?
     |  +--rw transport-qualified-processing-element?   -> /o-ran-pe:processing-elements/additional-
transport-session-type-elements[o-ran-pe:transport-session-type = current()/../transport-session-
type]/ru-elements/name {feat:MULTIPLE-TRANSPORT-SESSION-TYPE}?
     |  +--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
     |  +--rw enhanced-user-plane-uplink-marking?       -> /o-ran-pe:processing-elements/additional-
transport-session-type-elements[o-ran-pe:transport-session-type = current()/../transport-session-
type]/enhanced-uplane-mapping/uplane-mapping/up-marking-name {feat:MULTIPLE-TRANSPORT-SESSION-TYPE}?
     +--ro endpoint-types* [id]
     |  +--ro id                                                          uint16
     |  +--ro supported-section-types* [section-type]
     |  |  +--ro section-type                    uint8
     |  |  +--ro supported-section-extensions*   uint8
     |  +--ro st4-supported-commands* [st4-command-type] {feat:ST4-SLOT-CONFIG-MSG-SUPPORT}?
     |  |  +--ro st4-command-type    enumeration
     |  +--ro st4-reception-mask* [cmd-scope] {feat:ST4-SLOT-CONFIG-MSG-SUPPORT}?
     |  |  +--ro cmd-scope               enumeration
     |  |  +--ro st4-reception-bitmask?   uint16
     |  +--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 max-prb-ranges-per-symbol?                                  uint32
     |  +--ro max-mcscaleremask-per-prb?                                  uint16
     |  +--ro max-prb-ranges-per-sec-ext-12?                              uint16
     |  +--ro max-freq-hops-per-sec-ext-13?                               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                             compression-type-def
     |  |  x--ro bitwidth?                                    uint8
     |  |  +--ro compression-method?                          compression-method-def
     |  |  x--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 cplane-message-processing-limits-required?                           boolean
{feat:CPLANE-MESSAGE-PROCESSING-LIMITS}?
     |  +--ro max-beams-per-cplane-message?                                        uint16
{feat:CPLANE-MESSAGE-PROCESSING-LIMITS}?
     |  +--ro max-highest-priority-sec-per-cplane-message?                         uint16
{feat:CPLANE-MESSAGE-PROCESSING-LIMITS}?
     |  +--ro max-beams-per-slot-with-cplane-limits?                               uint16
{feat:CPLANE-MESSAGE-PROCESSING-LIMITS}?
     |  +--ro max-highest-priority-sections-per-slot-with-cplane-limits?           uint16
{feat:CPLANE-MESSAGE-PROCESSING-LIMITS}?
     |  +--ro max-num-se22-per-cplane-message?                                     uint16
{feat:CPLANE-MESSAGE-PROCESSING-LIMITS}?
     |  +--ro max-prb-ranges-per-hp-section-sec-ext-12?                            uint16
{feat:CPLANE-MESSAGE-PROCESSING-LIMITS}?
     |  +--ro uplane-message-processing-limits-required?                           boolean
{feat:UPLANE-MESSAGE-PROCESSING-LIMITS}?
     |  +--ro max-section-headers-per-uplane-message?                              uint16
{feat:UPLANE-MESSAGE-PROCESSING-LIMITS,feat:UPLANE-MESSAGE-PROCESSING-LIMITS}?
     |  +--ro beam-update-contention-control-limits-required?                      boolean
{feat:BEAM-UPDATE-CONTENTION-CONTROL}?
     |  +--ro max-beams-per-symbol-with-beam-contention-control?                   uint16 {feat:BEAM-
UPDATE-CONTENTION-CONTROL}?
     |  +--ro max-beams-updates-per-symbol-with-beam-contention-control?           uint16 {feat:BEAM-
UPDATE-CONTENTION-CONTROL}?
     |  +--ro max-ack-nack-per-symbol?                                             uint16
     |  +--ro non-scheduled-ueid-supported?                                        boolean {feat:NON-
SCHEDULED-UEID}?
     |  +--ro se-11-continuity-flag-supported?                                     boolean
{feat:SE11-WITH-CONTINUITY-BIT-SUPPORT}?
     |  +--ro supported-configuration-combinations* [combination-id]
     |  |  +--ro combination-id    uint32
     |  |  +--ro set* [set-id]
     |  |     +--ro set-id                        uint32
     |  |     +--ro max-overlapping-instances?   uint32
     |  |     +--ro config* [config-id]
     |  |        +--ro config-id                   uint32
```

```
     |  |        +--ro scs*                         mcap:scs-config-type
     |  |        +--ro carrier-types*               enumeration
     |  |        +--ro filter-pass-bandwidth?       uint64
     |  |        +--ro max-prb-range?               uint32
     |  |        +--ro center-from-freqoffset?      boolean
     |  |        +--ro supported-filter-indices*    uint32
     |  +--ro center-from-freqoffset-supported?                              boolean
     |  +--ro supported-filter-indices*                                      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-prb-ranges-per-symbol?                              uint32
     |  +--ro max-numerologies-per-symbol?                            uint16
     |  +--ro max-mcscaleremask-per-prb?                              uint16
     |  +--ro max-prb-ranges-per-sec-ext-12?                          uint16
     |  +--ro max-freq-hops-per-sec-ext-13?                           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                            compression-type-def
     |  |  x--ro bitwidth?                                   uint8
     |  |  +--ro compression-method?                         compression-method-def
     |  |  x--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 max-beams-per-cplane-message?                           uint16 {feat:CPLANE-
MESSAGE-PROCESSING-LIMITS}?
     |  +--ro max-highest-priority-sec-per-cplane-message?            uint16 {feat:CPLANE-
MESSAGE-PROCESSING-LIMITS}?
     |  +--ro max-beams-per-slot-with-cplane-limits?                  uint16 {feat:CPLANE-
MESSAGE-PROCESSING-LIMITS}?
     |  +--ro max-highest-priority-sections-per-slot-with-cplane-limits?   uint16 {feat:CPLANE-
MESSAGE-PROCESSING-LIMITS}?
     |  +--ro max-num-se22-per-cplane-message?                        uint16 {feat:CPLANE-
MESSAGE-PROCESSING-LIMITS}?
     |  +--ro max-prb-ranges-per-hp-section-sec-ext-12?               uint16 {feat:CPLANE-
MESSAGE-PROCESSING-LIMITS}?
     |  +--ro max-ack-nack-per-symbol?                                uint16
     |  +--ro supported-configuration-combinations* [combination-id]
     |  |  +--ro combination-id    uint32
```

```
     |  |  +--ro set* [set-id]
     |  |     +--ro set-id                        uint32
     |  |     +--ro max-overlapping-instances?   uint32
     |  |     +--ro config* [config-id]
     |  |        +--ro config-id                   uint32
     |  |        +--ro scs*                        mcap:scs-config-type
     |  |        +--ro carrier-types*              enumeration
     |  |        +--ro filter-pass-bandwidth?      uint64
     |  |        +--ro max-prb-range?              uint32
     |  |        +--ro center-from-freqoffset?     boolean
     |  |        +--ro supported-filter-indices*   uint32
     |  +--ro center-from-freqoffset-supported?                           boolean
     +--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-method?                     compression-method-def
     |     x--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 extended-max-prach-patterns?          uint32 {feat:EXTENDED-PRACH-CONFIGURATION}?
     |  +--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 sro-id?                                        -> /or-user:users/user/sro-id
{feat:SHARED-ORU-MULTI-OPERATOR}?
     |  +--rw compression!
     |  |  +--rw iq-bitwidth?                              uint8
     |  |  +--rw compression-type                          compression-type-def
     |  |  x--rw bitwidth?                                 uint8
     |  |  +--rw compression-method?                       compression-method-def
     |  |  x--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?               compression-method-def
     |  |  |  +--rw fs-offset?                        uint8 {cf:CONFIGURABLE-FS-OFFSET}?
     |  |  |  x--rw channel-information-iq-bitwidth?   uint8
     |  |  +--rw channel-information-bitwidth?              uint8
     |  |  +--rw channel-information-compressions* [id] {feat:CHANNEL-INFORMATION-COMPRESSION}?
     |  |  |  +--rw id                                        uint16
     |  |  |  +--rw channel-information-compression-method?   cf:ci-compression-method-def
     |  |  |  +--rw iq-bitwidth?                              uint8
     |  |  +--rw bf-weights-compressions* [id]
     |  |     +--rw id                        uint16
     |  |     +--rw bf-weights-compression?   cf:bf-compression-method-def
     |  |     +--rw iq-bitwidth?              uint8
     |  +--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
{mcap:CONFIGURABLE-TDD-PATTERN-SUPPORTED}?
     |  +--rw cplane-message-processing-limits-enabled?      boolean {feat:CPLANE-MESSAGE-
PROCESSING-LIMITS}?
     |  +--rw uplane-message-section-header-limit-enabled?   boolean {feat:UPLANE-MESSAGE-
PROCESSING-LIMITS}?
     |  +--rw beam-update-contention-control-enabled?        boolean {feat:BEAM-UPDATE-CONTENTION-
CONTROL}?
     |  +--rw channel-information-prb-group-configuration {feat:CHANNEL-INFORMATION-PRB-GROUP}?
     |  |  +--rw enable-ci-prb-group?   boolean
     |  |  +--rw ci-prb-group-size?     uint8
     |  +--rw non-scheduled-ueid-enabled?                    boolean {feat:NON-SCHEDULED-UEID}?
```

```
     |  +--rw se-11-continuity-flag-enabled?                 boolean {feat:SE11-WITH-CONTINUITY-BIT-
SUPPORT}?
     |  +--rw combination-configuration
     |     +--rw endpoint-type?    -> /user-plane-configuration/static-low-level-tx-
endpoints[name=current()/../../name]/endpoint-type
     |     +--rw combination-id?   -> /user-plane-configuration/endpoint-types[id =
current()/../endpoint-type]/supported-configuration-combinations/combination-id
     |     +--rw configurations* [id]
     |        +--rw id           uint32
     |        +--rw set-id?      -> /user-plane-configuration/endpoint-types[id =
current()/../../endpoint-type]/supported-configuration-combinations[combination-id =
current()/../../combination-id]/set/set-id
     |        +--rw config-id?   -> /user-plane-configuration/endpoint-types[id =
current()/../../endpoint-type]/supported-configuration-combinations[combination-id =
current()/../../combination-id]/set[set-id = current()/../set-id]/config/config-id
     +--rw low-level-rx-endpoints* [name]
     |  +--rw name                                        -> /user-plane-configuration/static-low-
level-rx-endpoints/name
     |  +--rw sro-id?                                     -> /or-user:users/user/sro-id
{feat:SHARED-ORU-MULTI-OPERATOR}?
     |  +--rw compression!
     |  |  +--rw iq-bitwidth?                                uint8
     |  |  +--rw compression-type                            compression-type-def
     |  |  x--rw bitwidth?                                   uint8
     |  |  +--rw compression-method?                         compression-method-def
     |  |  x--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?   compression-method-def
     |  |  |  +--rw fs-offset?            uint8 {cf:CONFIGURABLE-FS-OFFSET}?
     |  |  +--rw bf-weights-compressions* [id]
     |  |     +--rw id                        uint16
     |  |     +--rw bf-weights-compression?   cf:bf-compression-method-def
     |  |     +--rw iq-bitwidth?              uint8
     |  +--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
     |  |  +--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 {mcap:PRACH-STATIC-CONFIGURATION-SUPPORTED}?
     |  +--rw static-srs-configuration?                   -> /user-plane-configuration/static-srs-
configurations/static-srs-config-id {mcap:SRS-STATIC-CONFIGURATION-SUPPORTED}?
     |  +--rw configurable-tdd-pattern-supported?         -> /user-plane-configuration/static-low-
level-rx-endpoints[name=current()/../name]/configurable-tdd-pattern-supported {mcap: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 cplane-message-processing-limits-enabled?   boolean {feat:CPLANE-MESSAGE-PROCESSING-
LIMITS}?
     |  +--rw beam-update-contention-control-enabled?     boolean {feat:BEAM-UPDATE-CONTENTION-
CONTROL}?
     |  +--rw non-scheduled-ueid-enabled?                 boolean {feat:NON-SCHEDULED-UEID}?
     |  +--rw center-from-freqoffset-enabled?             boolean
     |  +--rw combination-configuration
     |     +--rw endpoint-type?    -> /user-plane-configuration/static-low-level-rx-
endpoints[name=current()/../../name]/endpoint-type
     |     +--rw combination-id?   -> /user-plane-configuration/endpoint-types[id =
current()/../endpoint-type]/supported-configuration-combinations/combination-id
     |     +--rw configurations* [id]
     |        +--rw id           uint32
     |        +--rw set-id?      -> /user-plane-configuration/endpoint-types[id =
current()/../../endpoint-type]/supported-configuration-combinations[combination-id =
current()/../../combination-id]/set/set-id
     |        +--rw config-id?   -> /user-plane-configuration/endpoint-types[id =
current()/../../endpoint-type]/supported-configuration-combinations[combination-id =
current()/../../combination-id]/set[set-id = current()/../set-id]/config/config-id
     +--rw tx-array-carriers* [name]
     |  +--rw name                          string
     |  +--rw odu-ids* [odu-id] {feat:SHARED-ORU-MULTI-ODU}?
     |  |  +--rw odu-id    string
     |  +--rw sro-ids-and-odu-ids* [odu-id sro-id] {feat:SHARED-ORU-MULTI-ODU and feat:SHARED-ORU-
MULTI-OPERATOR}?
     |  |  +--rw odu-id    string
     |  |  +--rw sro-id    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
     |  x--rw rw-duplex-scheme?              -> /user-plane-configuration/tx-array-
carriers[name=current()/../name]/duplex-scheme
     |  x--rw rw-type?                       -> /user-plane-configuration/tx-array-
carriers[name=current()/../name]/type
     |  +--rw occupied-bandwidth
     |  |  +--rw lower-bound?   uint64
     |  |  +--rw upper-bound?   uint64
     |  +--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 t-da-offset?                   uint32 {feat:EXT-ANT-DELAY-CONTROL}?
     |  +--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
     |  +--rw odu-ids* [odu-id] {feat:SHARED-ORU-MULTI-ODU}?
     |  |  +--rw odu-id    string
     |  +--rw sro-ids-and-odu-ids* [odu-id sro-id] {feat:SHARED-ORU-MULTI-ODU and feat:SHARED-ORU-
MULTI-OPERATOR}?
     |  |  +--rw odu-id    string
     |  |  +--rw sro-id    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 occupied-bandwidth
     |  |  +--rw lower-bound?   uint64
     |  |  +--rw upper-bound?   uint64
     |  +--rw downlink-radio-frame-offset    uint32
     |  +--rw downlink-sfn-offset            int16
     |  +--rw gain-correction                decimal64
     |  +--rw n-ta-offset                    uint32
     |  +--rw t-au-offset?                   uint32 {feat:EXT-ANT-DELAY-CONTROL}?
     |  +--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 related-o-ru-connectors* [name]
     |  |  +--ro name                -> /hw:hardware/component/name
     |  |  +--ro array-element-id*   uint16
     |  +--ro min-gain?                                 decimal64
     |  +--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 supported-filter-pass-bandwidths-dl
     |        +--ro supported-filter-pass-bandwidths* [id]
     |           +--ro id                        uint32
     |           +--ro type?                     enumeration
     |           +--ro carrier-bandwidth?        uint64
     |           +--ro filter-pass-bandwidth?    uint64
     +--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 related-o-ru-connectors* [name]
     |  |  +--ro name                -> /hw:hardware/component/name
     |  |  +--ro array-element-id*   uint16
     |  +--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 supported-filter-pass-bandwidths-ul
     |        +--ro supported-filter-pass-bandwidths* [id]
     |           +--ro id                      uint32
     |           +--ro type?                   enumeration
     |           +--ro carrier-bandwidth?      uint64
     |           +--ro filter-pass-bandwidth?  uint64
     +--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)
     |  |        +--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 sfn-offset?               uint32
     |  +--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         uint32
     |     +--rw number-of-repetitions    uint8
     |     +--rw number-of-occasions      uint32
```

```
     |     +--rw re-offset                uint32
     |     +--rw occasion-parameters* [occasion-id]
     |     |  +--rw occasion-id    uint32
     |     |  +--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
        +--rw uplane-only-dl-mode-enable?                        boolean {feat:UPLANE-ONLY-DL-MODE}?
        +--rw st4-for-time-domain-beamforming-weights-enabled?   boolean {feat:ST4-SLOT-CONFIG-MSG-
SUPPORT}?

  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
```

D.3.9 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.3.10 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.3.11 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
```

D.3.12 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.3.13 o-ran-ieee802-dot1q-cfm module

The format for the o-ran-ieee802-dot1q-cfm module is provided below

```
module: o-ran-ieee802-dot1q-cfm

  augment /dot1q-cfm:cfm/dot1q-cfm:maintenance-group/dot1q-cfm:mep:
    +--rw interface?     if:interface-ref
    +--rw primary-vid?   -> /if:interfaces/interface/o-ran-int:vlan-id
```

D.4 Interfaces folder

D.4.1 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:
    +--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.4.2 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 {o-ran-int:UDPIP-BASED-CU-PLANE}?
     |        +--:(ipv6)
     |           +--rw upv6-dscp-name?   -> /if:interfaces/interface/ip:ipv6/o-ran-int:diffserv-
markings/enhanced-uplane-markings/up-marking-name {o-ran-int:UDPIP-BASED-CU-PLANE}?
```

```
     +--rw ru-elements* [name]
     |  +--rw name              string
     |  +--rw sro-id?           -> /or-user:users/user/sro-id {feat:SHARED-ORU-MULTI-OPERATOR}?
     |  +--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
     +--rw additional-transport-session-type-elements* [transport-session-type] {feat:MULTIPLE-
TRANSPORT-SESSION-TYPE}?
        +--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 sro-id?           -> /or-user:users/user/sro-id {feat:SHARED-ORU-MULTI-OPERATOR}?
           +--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
```

D.4.3 o-ran-transceiver.yang module

The format for the (SFP) transceiver 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 interface-names*                    -> /if:interfaces/interface/name
        +--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 interface-names*   -> /if:interfaces/interface/name
           +--ro rx-power?          decimal64
           +--ro tx-bias-current?   decimal64
           +--ro tx-power?          decimal64
```

D.4.4 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 scan-interval?            uint16
     +--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
```

D.4.5 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 type-code?                           uint16
     |  |  +--ro (duid-type)?
     |  |  |  +--:(duid-llt)
     |  |  |  |  +--ro duid-llt-hardware-type?        uint16
     |  |  |  |  +--ro duid-llt-time?                 yang:timeticks
     |  |  |  |  +--ro duid-llt-link-layer-address?   yang:mac-address
     |  |  |  +--:(duid-en)
     |  |  |  |  +--ro duid-en-enterprise-number?     uint32
     |  |  |  |  +--ro duid-en-identifier?            string
     |  |  |  +--:(duid-ll)
     |  |  |  |  +--ro duid-ll-hardware-type?         uint16
     |  |  |  |  +--ro duid-ll-link-layer-address?    yang:mac-address
     |  |  |  +--:(duid-uuid)
     |  |  |  |  +--ro uuid?                          yang:uuid
     |  |  |  +--:(duid-unstructured)
     |  |  |     +--ro data?                          binary
     |  |  +--ro active-duid?                         binary
     |  |  +--ro identity-association
     |  |  |  +--ro iaid?      uint32
     |  |  |  +--ro ia-type?   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
     |     x--ro dhcp-client-identifier
     |     |  +--ro type-code?                           uint16
     |     |  +--ro (duid-type)?
     |     |     +--:(duid-llt)
     |     |     |  +--ro duid-llt-hardware-type?      uint16
     |     |     |  +--ro duid-llt-time?               yang:timeticks
     |     |     |  +--ro duid-llt-link-layer-addr?    yang:mac-address
     |     |     +--:(duid-en)
     |     |     |  +--ro duid-en-enterprise-number?   uint32
     |     |     |  +--ro duid-en-identifier?          string
     |     |     +--:(duid-ll)
     |     |     |  +--ro duid-ll-hardware-type?       uint16
     |     |     |  +--ro duid-ll-link-layer-addr?     yang:mac-address
     |     |     +--:(duid-uuid)
     |     |     |  +--ro uuid?                        yang:uuid
     |     |     +--:(duid-unknown)
     |     |        +--ro data?                        binary
     |     +--ro dhcpv6-client-identifier
     |     |  +--ro type-code?                           uint16
```

```
     |      |  +--ro (duid-type)?
     |      |  |  +--:(duid-llt)
     |      |  |  |  +--ro duid-llt-hardware-type?        uint16
     |      |  |  |  +--ro duid-llt-time?                 yang:timeticks
     |      |  |  |  +--ro duid-llt-link-layer-address?   yang:mac-address
     |      |  |  +--:(duid-en)
     |      |  |  |  +--ro duid-en-enterprise-number?     uint32
     |      |  |  |  +--ro duid-en-identifier?            string
     |      |  |  +--:(duid-ll)
     |      |  |  |  +--ro duid-ll-hardware-type?         uint16
     |      |  |  |  +--ro duid-ll-link-layer-address?    yang:mac-address
     |      |  |  +--:(duid-uuid)
     |      |  |  |  +--ro uuid?                          yang:uuid
     |      |  |  +--:(duid-unstructured)
     |      |  |     +--ro data?                          binary
     |      |  +--ro active-duid?                         binary
     |      |  +--ro identity-association
     |      |     +--ro iaid?      uint32
     |      |     +--ro ia-type    string
     |      x--ro dhcp-server-identifier
     |      |  +--ro type-code?                         uint16
     |      |  +--ro (duid-type)?
     |      |     +--:(duid-llt)
     |      |     |  +--ro duid-llt-hardware-type?      uint16
     |      |     |  +--ro duid-llt-time?               yang:timeticks
     |      |     |  +--ro duid-llt-link-layer-addr?    yang:mac-address
     |      |     +--:(duid-en)
     |      |     |  +--ro duid-en-enterprise-number?   uint32
     |      |     |  +--ro duid-en-identifier?          string
     |      |     +--:(duid-ll)
     |      |     |  +--ro duid-ll-hardware-type?       uint16
     |      |     |  +--ro duid-ll-link-layer-addr?     yang:mac-address
     |      |     +--:(duid-uuid)
     |      |     |  +--ro uuid?                        yang:uuid
     |      |     +--:(duid-unknown)
     |      |        +--ro data?                        binary
     |      +--ro dhcpv6-server-identifier
     |      |  +--ro type-code?                           uint16
     |      |  +--ro (duid-type)?
     |      |  |  +--:(duid-llt)
     |      |  |  |  +--ro duid-llt-hardware-type?        uint16
     |      |  |  |  +--ro duid-llt-time?                 yang:timeticks
     |      |  |  |  +--ro duid-llt-link-layer-address?   yang:mac-address
     |      |  |  +--:(duid-en)
     |      |  |  |  +--ro duid-en-enterprise-number?     uint32
     |      |  |  |  +--ro duid-en-identifier?            string
     |      |  |  +--:(duid-ll)
     |      |  |  |  +--ro duid-ll-hardware-type?         uint16
     |      |  |  |  +--ro duid-ll-link-layer-address?    yang:mac-address
     |      |  |  +--:(duid-uuid)
     |      |  |  |  +--ro uuid?                          yang:uuid
     |      |  |  +--:(duid-unstructured)
     |      |  |     +--ro data?                          binary
     |      |  +--ro active-duid?                         binary
     |      +--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.4.6 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
```

D.4.7 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.4.8 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.5 Sync folder

D.5.1 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
     |  +--ro boundary-clock-supported?   boolean
     +--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.6 Radio folder

D.6.1 o-ran-module-cap.yang module

The format for the module capabilities module is provided below

```
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 number-of-spatial-streams-dl?                   uint8
     |  +--ro number-of-spatial-streams-ul?                   uint8
     |  +--ro max-num-bands?                                  uint16
     |  x--ro max-power-per-pa-antenna?                       decimal64
     |  x--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                          compression-type-def
     |  |  |  x--ro bitwidth?                                 uint8
     |  |  |  +--ro compression-method?                       compression-method-def
     |  |  |  x--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 st6-4byte-alignment-required?               boolean
     |  |  +--ro se6-rb-bit-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 ext-ant-delay-capability?                       enumeration {or-feat:EXT-ANT-DELAY-
CONTROL}?
     |  +--ro nack-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 supported-filter-pass-bandwidths-dl
     |  |  +--ro supported-filter-pass-bandwidths* [id]
     |  |     +--ro id                          uint32
     |  |     +--ro type?                       enumeration
     |  |     +--ro carrier-bandwidth?          uint64
     |  |     +--ro filter-pass-bandwidth?      uint64
     |  +--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 supported-filter-pass-bandwidths-ul
     |  |  +--ro supported-filter-pass-bandwidths* [id]
     |  |     +--ro id                          uint32
     |  |     +--ro type?                       enumeration
     |  |     +--ro carrier-bandwidth?          uint64
     |  |     +--ro filter-pass-bandwidth?      uint64
     |  +--ro max-num-component-carriers?            uint8
     |  x--ro max-num-bands?                         uint16
     |  +--ro max-num-sectors?                       uint8
     |  x--ro max-power-per-antenna?                 decimal64
     |  x--ro min-power-per-antenna?                 decimal64
     |  x--ro codebook-configuration_ng?             uint8
     |  x--ro codebook-configuration_n1?             uint8
     |  x--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.6.2 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
     |     +--ro ta3-min-ack?     int32
     |     +--ro ta3-max-ack?     int32
     +--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.6.3 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
     |  |  +--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                            compression-type-def
     |  |  |  |     x--ro bitwidth?                                   uint8
     |  |  |  |     +--ro compression-method?                         bf-compression-method-def
     |  |  |  |     x--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 additional-compression-method-supported* []
     |  |  |  |        +--ro iq-bitwidth?                                uint8
     |  |  |  |        +--ro compression-type                            compression-type-def
     |  |  |  |        x--ro bitwidth?                                   uint8
     |  |  |  |        +--ro compression-method?                         bf-compression-method-def
     |  |  |  |        x--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
     |  |  |  +--:(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                           compression-type-def
     |  |  |  |     x--ro bitwidth?                                  uint8
     |  |  |  |     +--ro compression-method?                        bf-compression-method-def
     |  |  |  |     x--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 additional-compression-method-supported* []
     |  |  |  |        +--ro iq-bitwidth?                               uint8
     |  |  |  |        +--ro compression-type                           compression-type-def
     |  |  |  |        x--ro bitwidth?                                  uint8
     |  |  |  |        +--ro compression-method?                        bf-compression-method-def
     |  |  |  |        x--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
     |  |  |  +--:(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                           compression-type-def
     |  |  |        x--ro bitwidth?                                  uint8
     |  |  |        +--ro compression-method?                        bf-compression-method-def
     |  |  |        x--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 additional-compression-method-supported* []
   |  |  |        |  +--ro iq-bitwidth?                                uint8
   |  |  |        |  +--ro compression-type                            compression-type-def
   |  |  |        |  x--ro bitwidth?                                   uint8
   |  |  |        |  +--ro compression-method?                         bf-compression-method-def
   |  |  |        |  x--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 fd-weights-mapping* [fd-weight-number]
   |  |  |           +--ro fd-weight-number    uint16
   |  |  |           +--ro array-elements*     uint16
   |  |  +--ro number-of-beams?                uint16
   |  |  +--ro p-dash?                         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)
   |  |  |  |  +--ro frequency-domain-beams
   |  |  |  |     +--ro max-number-of-beam-ids                    uint16
   |  |  |  |     +--ro initial-beam-id                           uint16
   |  |  |  |     +--ro iq-bitwidth?                              uint8
   |  |  |  |     +--ro compression-type                          compression-type-def
   |  |  |  |     x--ro bitwidth?                                 uint8
   |  |  |  |     +--ro compression-method?                       bf-compression-method-def
   |  |  |  |     x--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 additional-compression-method-supported* []
     |  |  |  |        +--ro iq-bitwidth?                            uint8
     |  |  |  |        +--ro compression-type                        compression-type-def
     |  |  |  |        x--ro bitwidth?                               uint8
     |  |  |  |        +--ro compression-method?                     bf-compression-method-def
     |  |  |  |        x--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
     |  |  |  +--:(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                        compression-type-def
     |  |  |  |     x--ro bitwidth?                               uint8
     |  |  |  |     +--ro compression-method?                     bf-compression-method-def
     |  |  |  |     x--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 additional-compression-method-supported* []
     |  |  |  |        +--ro iq-bitwidth?                            uint8
     |  |  |  |        +--ro compression-type                        compression-type-def
     |  |  |  |        x--ro bitwidth?                               uint8
     |  |  |  |        +--ro compression-method?                     bf-compression-method-def
     |  |  |  |        x--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
     |  |  |  +--:(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                           compression-type-def
     |  |  |        x--ro bitwidth?                                  uint8
     |  |  |        +--ro compression-method?                        bf-compression-method-def
     |  |  |        x--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 additional-compression-method-supported* []
     |  |  |        |  +--ro iq-bitwidth?                               uint8
     |  |  |        |  +--ro compression-type                           compression-type-def
     |  |  |        |  x--ro bitwidth?                                  uint8
     |  |  |        |  +--ro compression-method?                        bf-compression-method-def
     |  |  |        |  x--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 fd-weights-mapping* [fd-weight-number]
     |  |  |           +--ro fd-weight-number    uint16
     |  |  |           +--ro array-elements*     uint16
     |  |  +--ro number-of-beams?                uint16
     |  |  +--ro p-dash?                         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 ue-specific-beamforming!
     |  x--ro max-number-of-ues?                                  uint8
     |  +--ro max-number-of-ues-15bit?                            uint16
     |  +--ro channel-information-compression-method-supported*    cf:ci-compression-method-def
{feat:CHANNEL-INFORMATION-COMPRESSION}?
     |  +--ro dynamic-channel-information-compression-supported?   boolean {feat:CHANNEL-
INFORMATION-COMPRESSION}?
     +--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* [capabilities-group]
        |  +--ro capabilities-group                         -> /beamforming-config/capabilities-
groups/capabilities-group
        |  +--ro elevation-tilt-offset-granularity          uint8
        |  +--ro azimuth-tilt-offset-granularity            uint8
        |  +--ro minimum-supported-elevation-tilt-offset    int16
        |  +--ro maximum-supported-elevation-tilt-offset    int16
        |  +--ro minimum-supported-azimuth-tilt-offset      int16
        |  +--ro maximum-supported-azimuth-tilt-offset      int16
        |  +--ro run-time-tilt-offset-supported             boolean
        +--ro predefined-beam-tilt-state* [capabilities-group]
           +--ro capabilities-group             -> /beamforming-config/capabilities-
groups/capabilities-group
           +--ro elevation-tilt-offset-angle    int16
           +--ro azimuth-tilt-offset-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* [capabilities-group] {BEAM-TILT}?
       |     +---w capabilities-group             -> /beamforming-config/capabilities-
groups/capabilities-group
       |     +---w elevation-tilt-offset-angle?   int16
       |     +---w azimuth-tilt-offset-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* [capabilities-group]
          +--ro capabilities-group             -> /beamforming-config/capabilities-
groups/capabilities-group
          +--ro elevation-tilt-offset-angle    int16
          +--ro azimuth-tilt-offset-angle      int16
```

D.6.4 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-scarriers?                 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.6.5 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}?
     |  +--ro configured-preparation-timer-supported?       boolean {O-RU-COORDINATED-ANT-CAL}?
     +--rw self-calibration-policy
     |  +--rw self-calibration-allowed?                                 boolean
     |  +--rw coordinated-calibration-allowed?                          boolean {O-RU-COORDINATED-
ANT-CAL}?
     |  +--rw coordinated-ant-calib-prep-timer?                         uint8 {O-RU-COORDINATED-
ANT-CAL}?
     |  +--rw coordinated-calibration-multiple-time-resources-allowed?   boolean {O-RU-COORDINATED-
ANT-CAL and O-RU-COORDINATED-ANT-CAL-MULTIPLE-TIME-RESOURCE}?
     +--ro antenna-calibration-multiple-time-resource {O-RU-COORDINATED-ANT-CAL and O-RU-
COORDINATED-ANT-CAL-MULTIPLE-TIME-RESOURCE}?
        +--ro antenna-calibration-multiple-time-resource-list* [antenna-calibration-time-resource-
index]
           +--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}?
           +--ro antenna-calibration-time-resource-index       uint8

  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
    +---n antenna-calibration-multiple-time-resource-params {O-RU-COORDINATED-ANT-CAL and O-RU-
COORDINATED-ANT-CAL-MULTIPLE-TIME-RESOURCE}?
       +--ro antenna-calibration-time-resource-index?   uint8
       +--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
```

D.6.6 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 t-combine-net?                        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 eaxc-id-group-capabilities {FHM}?
     |  |  +--ro max-num-rx-eaxc-id-groups?       uint8
     |  |  +--ro max-num-rx-eaxc-ids-per-group?   uint8
     |  +--ro scs-supported*                        o-ran-cmn:scs-config-type {FHM}?
     |  +--ro compression-method-supported* [] {FHM}?
     |  |  +--ro iq-bitwidth?                                 uint8
     |  |  +--ro compression-type                             compression-type-def
     |  |  x--ro bitwidth?                                    uint8
     |  |  +--ro compression-method?                          compression-method-def
     |  |  x--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 multi-cell-in-cascade-mode-supported?   boolean
     +--rw shared-cell-config
        +--rw (shared-cell-copy-combine-mode)?
        |  +--:(COMMON)
```

```
        |  |  +--rw shared-cell-copy-entities* [name]
        |  |  |  +--rw name                              string
        |  |  |  +--rw odu-id?                           string {feat:SHARED-ORU-MULTI-ODU}?
        |  |  |  +--rw sro-id?                           -> /or-user:users/user/sro-id {feat:SHARED-
ORU-MULTI-OPERATOR}?
        |  |  |  +--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 odu-id?                              string {feat:SHARED-ORU-MULTI-ODU}?
        |  |     +--rw sro-id?                              -> /or-user:users/user/sro-id
{feat:SHARED-ORU-MULTI-OPERATOR}?
        |  |     +--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 scs?                                 o-ran-cmn:scs-config-type
        |  |     +--rw ta3-prime-max?                       uint32
        |  |     +--rw tx-duration?                         uint32
        |  |     +--rw shared-cell-combine-uplane-config {FHM}?
        |  |        +--rw rx-eaxc-id* [eaxc-id]
        |  |        |  +--rw eaxc-id               uint16
        |  |        |  +--rw number-of-prb?        uint16
        |  |        |  +--rw cp-ul-section-type*   enumeration
        |  |        |  +--rw comression-method
        |  |        |     +--rw iq-bitwidth?                             uint8
        |  |        |     +--rw compression-type                         compression-type-def
        |  |        |     x--rw bitwidth?                                uint8
        |  |        |     +--rw compression-method?                      compression-method-def
        |  |        |     x--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
        |  |        x--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 odu-id?                               string {feat:SHARED-ORU-MULTI-ODU}?
        |  |  |  +--rw sro-id?                               -> /or-user:users/user/sro-id
{feat:SHARED-ORU-MULTI-OPERATOR}?
        |  |  |  +--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
        |  |  |     x--rw downlink-radio-frame-offset?   uint32
        |  |  |     x--rw downlink-sfn-offset?           int16
        |  |  +--rw shared-cell-combine-entities-for-selective-beam-id* [name]
        |  |     +--rw name                                 string
        |  |     +--rw odu-id?                              string {feat:SHARED-ORU-MULTI-ODU}?
        |  |     +--rw sro-id?                              -> /or-user:users/user/sro-id
{feat:SHARED-ORU-MULTI-OPERATOR}?
        |  |     +--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 scs?                                 o-ran-cmn:scs-config-type
        |  |     +--rw ta3-prime-max?                       uint32
        |  |     +--rw tx-duration?                         uint32
        |  |     +--rw shared-cell-combine-uplane-config {FHM}?
        |  |        +--rw rx-eaxc-id* [eaxc-id]
        |  |        |  +--rw eaxc-id              uint16
        |  |        |  +--rw number-of-prb?       uint16
        |  |        |  +--rw cp-ul-section-type*   enumeration
        |  |        |  +--rw comression-method
        |  |        |     +--rw iq-bitwidth?                              uint8
        |  |        |     +--rw compression-type                          compression-type-def
        |  |        |     x--rw bitwidth?                                 uint8
        |  |        |     +--rw compression-method?                       compression-method-def
        |  |        |     x--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
        |  |        x--rw number-of-prb                  uint16
        |  +--:(SELECTIVE)
        +--rw max-num-rx-eaxc-ids-per-group?                              -> /shared-cell/shared-
cell-module-cap/eaxc-id-group-capabilities/max-num-rx-eaxc-ids-per-group {FHM}?
        +--rw max-num-rx-eaxc-id-groups?                                  -> /shared-cell/shared-
cell-module-cap/eaxc-id-group-capabilities/max-num-rx-eaxc-id-groups {FHM}?
        +--rw rx-eaxc-id-group* [representative-rx-eaxc-id] {FHM}?
        |  +--rw representative-rx-eaxc-id    uint16
        |  +--rw member-rx-eaxc-id*           uint16
        +--rw enhanced-t-combine-enabled?                                 boolean {feat:ENHANCED-T-
COMBINE}?
        +--rw multiple-scs-in-eaxc-used?                                  boolean {feat:MULTIPLE-
SCS-IN-EAXC}?
```

Annex E (normative): Corresponding YANG Module Definition

The definitions for each method and resource defined in the clauses of the present document include normative references to schema nodes, notifications and remote procedure calls (RPCs) defined in a set of corresponding YANG models. The YANG models for present document are available in the file O-RAN.WG4.MP-YANGs-R003-v12.00.zip which can be downloaded from the O-RAN Alliance website http://www.o-ran.org/specifications/.

If there is any conflict between the schema nodes, notifications and remote procedure calls (RPCs) defined in the YANG models and the accompanying description in the present document, the definition of the YANG models shall take precedence.

Table E-1 lists all of the O-RAN Alliance YANG models included in the O-RAN.WG4.MP-YANGs-R003-v12.00.zip file and their corresponding version numbers and revision dates.

Table E-1: O-RAN Alliance defined YANG model versions to be used with this present document.

YANG Module Name	Namespace	Revision Date	Description Version
o-ran-ald	urn:o-ran:ald:1.0	2021-12-01	1.2.0
o-ran-ald-port	urn:o-ran:ald-port:1.0	2021-12-01	1.3.0
o-ran-beamforming	urn:o-ran:beamforming:1.0	2023-04-10	11.1.0
o-ran-antenna-calibration	urn:o-ran:antcal:1.0	2021-12-01	7.1.0
o-ran-certificates	urn:o-ran:certificates:1.0	2022-08-15	10.0.0
o-ran-common-identity-refs	urn:o-ran:wg1identityref:1.0	2020-11-01	1.0.0
o-ran-common-yang-types	urn:o-ran:common-yang-types:1.0	2022-08-15	1.1.0
o-ran-compression-factors	urn:o-ran:compression-factors:1.0	2021-12-01	8.0.0
o-ran-delay-management	urn:o-ran:delay:1.0	2022-08-15	10.0.0
o-ran-dhcp	urn:o-ran:dhcp:1.0	2022-08-15	10.0.0
o-ran-ecpri-delay	urn:o-ran:message5:1.0	2021-12-01	8.0.0
o-ran-ethernet-forwarding	urn:o-ran:ethernet-fwd:1.0	2021-12-01	3.1.0
o-ran-externalio	urn:o-ran:external-io:1.0	2019-07-03	1.1.0
o-ran-fan	urn:o-ran:fan:1.0	2021-12-01	1.2.0
o-ran-file-management	urn:o-ran:file-management:1.0	2023-04-10	10.1.0
o-ran-fm	urn:o-ran:fm:1.0	2022-08-15	10.0.0
o-ran-hardware	urn:o-ran:hardware:1.0	2022-12-05	10.1.0
o-ran-ieee802-dot1q-cfm	urn:o-ran:o-ran-ieee802-dot1q-cfm:1.0	2023-04-10	12.0.0
o-ran-interfaces	urn:o-ran:interfaces:1.0	2021-12-01	5.2.0
o-ran-laa	urn:o-ran:laa:1.0	2022-08-15	1.2.0
o-ran-laa-operations	urn:o-ran:laa-operations:1.0	2021-12-01	1.3.0
o-ran-lbm	urn:o-ran:lbm:1.0	2021-12-01	1.2.0
o-ran-module-cap	urn:o-ran:module-cap:1.0	2023-04-10	12.0.0
o-ran-mplane-int	urn:o-ran:mplane-interfaces:1.0	2021-12-01	7.1.0
o-ran-operations	urn:o-ran:operations:1.0	2023-04-10	10.1.0
o-ran-performance-management	urn:o-ran:performance-management:1.0	2022-08-15	8.1.0
o-ran-processing-element	urn:o-ran:processing-element:1.0	2022-08-15	10.0.0
o-ran-shared-cell	urn:o-ran:shared-cell:1.0	2023-04-10	12.0.0
o-ran-software-management	urn:o-ran:software-management:1.0	2022-12-05	11.0.0
o-ran-supervision	urn:o-ran:supervision:1.0	2021-12-05	11.0.0
o-ran-sync	urn:o-ran:sync:1.0	2022-08-15	8.1.0
o-ran-trace	urn:o-ran:trace:1.0	2022-08-15	1.2.0
o-ran-transceiver	urn:o-ran:transceiver:1.0	2023-04-10	11.1.0
o-ran-troubleshooting	urn:o-ran:troubleshooting:1.0	2022-08-15	1.2.0
o-ran-udp-echo	urn:o-ran:udpecho:1.0	2019-02-04	1.0.0
o-ran-uplane-conf	urn:o-ran:uplane-conf:1.0	2023-04-10	12.0.0
o-ran-usermgmt	urn:o-ran:user-mgmt:1.0	2022-08-15	10.0.0
o-ran-ves-subscribed-notifications	urn:o-ran:ves-sn:1.0	2020-12-10	5.0.0
o-ran-wg4-features	urn:o-ran:wg4feat:1.0	2023-04-10	12.0.0

Table E-2 lists all of the externally defined YANG models included in the O-RAN.WG4.MP-YANGs-R003-v12.00.zip file and their corresponding revision dates.

Table E-2: Externally defined YANG model versions to be used with the present document.

YANG Module Name	Namespace	Revision Date
iana-crypt-hash	urn:ietf:params:xml:ns:yang:iana-crypt-hash	2014-08-06
iana-hardware	urn:ietf:params:xml:ns:yang:iana-hardware	2018-03-13
iana-if-type	urn:ietf:params:xml:ns:yang:iana-if-type	2017-01-19
ieee802-dot1x	urn:ieee:std:802.1X:yang:ieee802-dot1x	2020-02-18
Ieee802-dot1x-types	urn:ieee:std:802.1X:yang:ieee802-dot1x-types	2020-02-18
ieee802-types	urn:ieee:std:802.1Q:yang:ieee802-types	2020-06-04
ietf-crypto-types	urn:ietf:params:xml:ns:yang:ietf-crypto-types	2022-12-12
ietf-datastores	urn:ietf:params:xml:ns:yang:ietf-datastores	2018-02-14
ietf-dhcpv6-common	urn:ietf:params:xml:ns:yang:ietf-dhcpv6-common	2021-01-29
ietf-dhcpv6-types	urn:ietf:params:xml:ns:yang:ietf-dhcpv6-types	2018-09-04
ietf-hardware	urn:ietf:params:xml:ns:yang:ietf-hardware	2018-03-13
ietf-inet-types	urn:ietf:params:xml:ns:yang:ietf-inet-types	2013-07-15
ietf-interfaces	urn:ietf:params:xml:ns:yang:ietf-interfaces	2018-02-20
ietf-ip	urn:ietf:params:xml:ns:yang:ietf-ip	2018-02-22
ietf-netconf-acm	urn:ietf:params:xml:ns:yang:ietf-netconf-acm	2018-02-14
ietf-netconf-monitoring	urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring	2010-10-04
ietf-netconf-notifications	urn:ietf:params:xml:ns:yang:ietf-netconf-notifications	2012-02-06
ietf-network-instance	urn:ietf:params:xml:ns:yang:ietf-network-instance	2019-01-21
ietf-restconf	urn:ietf:params:xml:ns:yang:ietf-restconf	2017-01-26
ietf-subscribed-notifications	urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications	2019-09-09
ietf-system	urn:ietf:params:xml:ns:yang:ietf-system	2014-08-06
ietf-truststore	urn:ietf:params:xml:ns:yang:ietf-truststore	2022-10-19
ietf-yang-library	urn:ietf:params:xml:ns:yang:ietf-yang-library	2019-01-04
ietf-yang-schema-mount	urn:ietf:params:xml:ns:yang:ietf-yang-schema-mount	2019-01-14
ietf-yang-types	urn:ietf:params:xml:ns:yang:ietf-yang-types	2013-07-15
ietf-x509-cert-to-name	urn:ietf:params:xml:ns:yang:ietf-x509-cert-to-name	2014-12-10
ieee802-dot1q-cfm	urn:ieee:std:802.1Q:yang:ieee802-dot1q-cfm	2022-01-19
ieee802-dot1q-cfm-types	urn:ieee:std:802.1Q:yang:ieee802-dot1q-cfm-types	2020-06-04
ieee802-dot1q-types	urn:ieee:std:802.1Q:yang:ieee802-dot1q-types	2018-03-07

Editor's Note: Use of ietf-dhcpv6-types with a revision date of 2018-09-04 is deprecated. The reference to ietf-crypto-types with a revision date of 2022-12-12, ietf-dhcpv6-common with a revision date of 2021-01-29 and ietf-truststore with a revision date of 2022-10-19 will be superseded once the corresponding RFCs are published.

Annex F (informative): Out of scope functionality

F.1 Out of scope functionality

The present document does not include specific functions that, during the specification definition process, have been highlighted as useful enhancements. The list of identified functionalities is shown below. This can be used to prioritize future specification work within O-RAN Alliance.

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) Shared cell topology discovery for architectures that do not ensure Ethernet frames sent between a south-node and a north-node are bridged by an O-RU

6) Enabling an O-RU to be provisioned with multiple trust anchors

7) Dynamic sharing of O-RU carriers between multiple O-DUs

8) Enhanced adaptive delay operation that allows O-RU delay adaptation without requiring active carriers to be disabled.

9) The role of a Shared Resource Operator's SMO when operating with a multi-operator O-RU operated by a third party Shared O-RU Host.

Annex G (informative): o-ran-lbm.yang and o-ran-ieee802-dot1q-cfm.yang co-ordination

G.1 Model structure

In o-ran-lbm.yang model, the YANG structure is illustrated in Figure G-1:

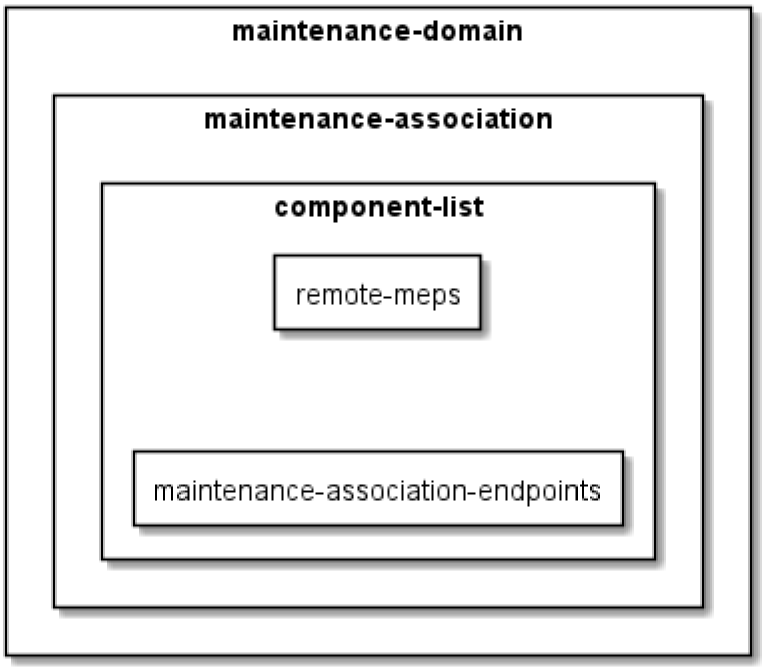


Figure G-1: o-ran-lbm YANG structure

In contrast, the ieee-802-dot1q-cfm YANG model has the structure illustrated in Figure G-2

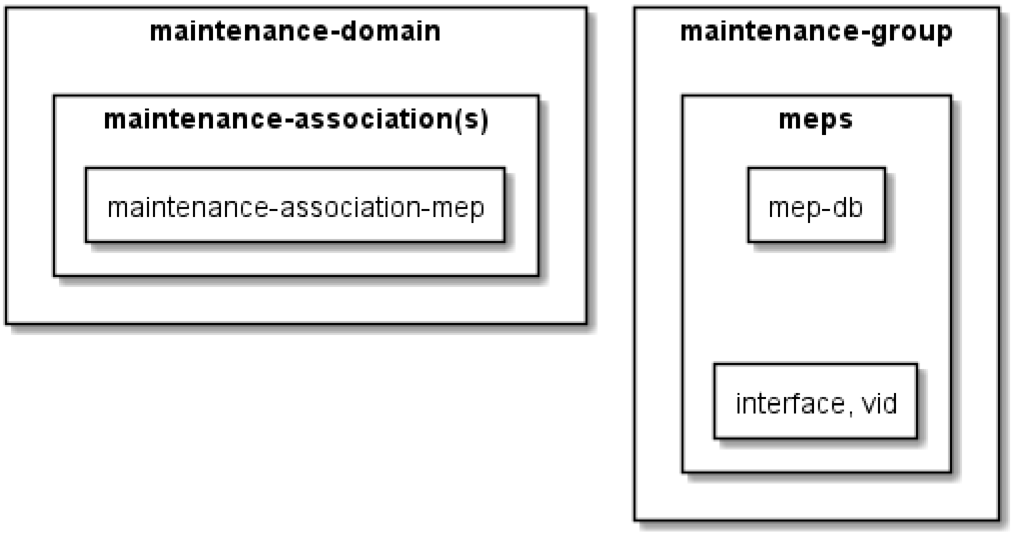


Figure G-2: ieee-802-dot1q-cfm YANG structure

The ieee802-dot1q-cfm.yang model is not standalone model. It needs to be extended to work with ieee-dot1q-cfm-bridge model or some equvaliant. In o-ran-ieee802-dot1q-cfm.yang, **interface** and **primary-vid** are used to map to o-ran-interfaces (where an l2vlan interface implicitly maps to an IEEE 802.1Q bridge equivalent).

In both models, the maintenance association endpoint (mep) is mapped to a l2vlan interface (bridge interface) and the l2vlan interface is mapped to a bridge equivalent. In the /cfm/maintenance-group/mep, **interface** and **primary-vid** represents an ieee equivalent bridge intance. Association towards an ieee equivalent bridge is represented by parameter on MEP in maintenance -group.

In o-ran-lbm.yang model, **remote-meps** and **maintenance-association-end-points** are defined on the same level under schema node **component-list**. **Remote-meps** contains MEP IDs that are in the MA but are located remotely (opposite to local) and **maintenance-association-endpoint** contains mep configurations.

In ieee-802-dot1q-cfm model, information of meps is in two lists, **maintenance-domain** and **maintenance-group**.

The ieee-802-dot1q-cfm **maintenance-association-mep** list contains **mep-id**s for all configured meps that are in the MA. The **mep** list in **maintenance-group** list has mep configurations for example mac address, enable paramter etc. And in ieee-802-dot1q-cfm model, each mep contains a **mep-db** list used to store information of its remote-meps. In ieee-802-dot1q, if a maintenance-group is defined, remote MEP state machine is instantiated to all remote MEPs that are presented in **maintenance-association-mep** list (that is active).

G.2 Mapping between models

To map the a o-ran-lbm.yang configuration to ieee-802-dot1q-cfm, the level of **component** which represent a bridge will be mapped to ieee equivalent bridge instancewhich is represented by the combination of value **interface** and **primary-vid**. The **remote-meps** leaf-list in o-ran-lbm.yang will be mapped to the **maintenance-association-mep** list in ieee-802-dot1q-cfm model.

To map the a o-ran-lbm.yang configuration to ieee-802-dot1q-cfm, **maintenance-group** will be generated for each MD, MA combination. **maintenance-association-end-point** configuration in o-ran-lbm.yang will be mapped as /**maintenance-group/MEP** in ieee-802-dot1q-cfm model, as illustrated in Table G-1. Table G-2, Table G-3, and Table G-4 describe the mapping between the two YANG schemas.

Table G-1 High level relationship between o-ran-lbm.yang and ieee-802-dot1q-cfm.yang

Model: o-ran-lbm-.yang	Model: ieee-802-dot1q-cfm.yang	Comment
/md-data-definiton/maintenance-domain	/cfm/maintenance-domain	
/md-data-definiton/maintenance-domain/maintenance-association	/cfm/maintenance-domain /maintenance-association	
/md-data-definiton/maintenance-domain/maintenance-association /component-list	To parameter interface and vid in /cfm/maintenance-group/mep	
/md-data-definiton/maintenance-domain/maintenance-association /component-list/Remote-meps	/cfm/maintenance-domain /maintenance-association /maintenance-association-mep	The local mep ids in lbm model need to be added as well.
/md-data-definiton/maintenance-domain/maintenance-association /component-list /maintenance-association-end-point	/cfm/maintenance-group/mep	For each MD MA combination, a new maintance-group can be created.

Table G-2 Mapping between maintenance-domain in o-ran-lbm.yang and ieee-802-dot1q-cfm.yang

List: /md-data-definiton/maintenance-domain	List: /cfm/maintenance-domain	Comment
id	md-id	
name	md-name	
md-level	md-level	

Table G-3 Mapping between maintenance-association lists in o-ran-lbm.yang and ieee-802-dot1q-cfm.yang

/md-data-definiton/maintenance-domain/maintenance-association		/cfm/maintenance-domain /maintenance-association	Comment
parameter	sub-parameter	parameter	
id		ma-id	
name		ma-name	
Component-list			Not mapped
	Component-id		Not mapped
	name	/cfm/maintenance-group/mep/interface	
	vid	/cfm/maintenance-group/mep/vid	
	Remote-meps	cfm/maintenance-domain /maintenance-association /maintenance-association-mep	
	Maitaince-association-meps	/cfm/maintenance-group/mep	

Table G-4 Mapping MEP configuration between o-ran-lbm.yang and ieee-802-dot1q-cfm.yang

/md-data-definiton/maintenance-domain/maintenance-association /component-list /maintenance-association-end-point		/cfm/maintenance-group/mep	Comment
parameter	sub-parameter	parameter	
Mep-identifier		mep-id	
interface		interface	Augmented in ieee-dot1q-cfm
Primary-vid		vid	Augmented in ieee-dot1q-cfm
Administrative-state		enabled	
Mac-address		Mac-address	
loopback		N/A	
	Reply-transmitted	State/mep-lbr-out	

Change 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
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 document - 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

Date	Revision	Description
2020.12.10	05.00	Bug fixes and correction to v04.00: - Clarify operation of default account 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 centre 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
2021.07.26	07.00	Bug fixes and correction to v06.00: - Clarify operation of configured subscriptions - Clarify username syntax - Correction to log management sequence diagrams - Clarify delay management operation - Clarify DHCP operation - Clarify operation of antenna calibration - Correct errors in text that describes low-level-[tr]x-endpoint creation - Clarify modify parameter section - Clarify revision and namespace compatibility handling - Clarify certificate enrolment - Clarify non-persistent operation with software management - Clarify Fault Management Activation - Correction to enable multiple measurements to be included in a notification - Clarify operation of NETCONF supervision with multiple NETCONF clients Addition of new functionality, including: - External antenna delay handling - Optional capability to optimize VLAN discovery - Capability to support C-Plane limits for packet processing - FTPES based file transfer - TD-RSSI measurement capability - EPE measurements for current and voltage - Configurable timer for co-ordinated antenna calibration - Enhanced antenna calibration using different resource sets
2021.10.28	07.01	Updates to align with ETSI PAS Process

Date	Revision	Description
2021.12.01	08.00	Bug fixes and correction to v07.01: - Deprecate leaf-list(frequency-table) and to add a new list(frequency-bin-table) - Correction for C-plane message limits - Antenna Calibration schema node reference corrections - Clarification for fault ID 24 - DHCP Clarifications - Clarifications to log management - Adding SCS information for FHM Combine operations - Shared cell performance management corrections Addition of new functionality, including: - Introduction of alarm-type - Enable IANA Private Enterprise Number to be used as Vendor Code in software management - Boundary Clock function - Capability for supporting multi cell operation - Compression support for Section Type 6 - Mandatory support of TLS, PKIX, and FTPES - Supporting multiple transport session types simultaneously - Shared cell enhancements introducing t-combine-net and tx-duration
2022.04.18	09.00	Bug fixes and correction to v08.00: - Updates to align with new O-RAN template - Updates to references to align with DFT guidelines - Clarify rejection of invalid configuration - Alignment of eventName field and “remote-file-path” format - Clarification of handling of an O-RU capable of BF and non-BF modes - Update clause cases - Correction Clarification of S-Plane interactions with M-Plane during Startup, and Loss/Recovery of Synchronization - Normative Reference to YANG models - Clarifiy NACM and user management - Clarify NETCONF monitoring and <get-schema> - Reset handling clarification - NETCONF event stream clarification - Software Management Clarification - SSH and SFTP host key clarification - SFTP host key correction Addition of new functionality, including: - New feature flag for O-RU to indicate support of 4-Byte aligned section type 6 - New build-content-download leaf and associated software management definition - New list for historical alarms - Defining loopback message to group destination address
2022.08.15	10.00	Bug fixes and correction to v09.00: - Clarify contradicting statements with regard to TLS 1.2 - Array element fault source and fault source improvements - Software management clarification - Correction for module privileges - Persistence of configured subscription - Correction for O-RU call home port - Clarify operation of filter based on xpath - Clarify operation with factory default software - Clarification to path definition for file management - Remove LAA “capability” option from Table C.3-1 - Password for FTPES server - Call home with multiple A/AAAA records - PKI Clarifications - Hybrid management clarifications Addition of new functionality, including: - Identifying default NETCONF accounts - Introduction of O-RU connectors - Configuration to map from certificates to NETCONF usernames - Update SFP compliance codes - DHCPv4 Client DUID/IAID definition - IEEE 802.1X Port based access control - Shared O-RU

Date	Revision	Description
2022.12.05	11.00	Bug fixes and correction to v10.00: - Correcting hardware access privileges for SMO - Correct fault ID 31 cancel condition and source - Correct o-ran-laa YANG tree - Add voltage and current EPE stats to Annex B.5 - Deprecation of power parameters for shared O-RU - Annex C.2 and C.3 reformat tables - Future topics annex - Clarification for handling of multiple interfaces per optical port - Clarify writable-running datastore - Clarification for PM measurement objects modification - Energy saving state clarification - Clarification of active parameter - Adding IEEE 802.1X recommendation Addition of new functionality, including: - U-Plane-only DL mode - SW build level file integrity check - Session supervision with session-id - Network energy savings
2023.04.10	12.00	Bug fixes and correction to v11.00: - Correct inconsistency in clause 9.1.3 - Update reference to DHCPv6 Options - Remove import of ietf-crypto-types@2019-04-29 - Confirmation of backwards compatibility - Clarification of the accounts name - DHCP options clarification - Clarify default operation of 802.1X - Correction Radio Timing for Copy - Correction to shared cell counters - Clarification of PRACH repetition - Clarified operational state for layer 3 configuration - Remove erroneous use of SZTP Addition of new functionality, including: - 802.1X as Mandatory - Add supported SCSes of FHM - Allow all NETCONF clients to change their own password - NB IoT - Advanced endpoint capability reporting for NB-IoT & other scenarios - Allow CA/RA server on non-production VLAN - Introduce ietf-truststore YANG model - Introduce SE11-WITH-CONTINUITY-BIT-SUPPORT - Addition of feature non-scheduled-ueid - Continuity Check Message support - U-plane message processing limits
2024.03.14	12.01	Corrections to v12.00: - Fixing typographical and cross reference errors

History

Document history		
V12.01	May 2024	Publication