ETSI TS 103 859 V7.0.2 (2022-09)

TECHNICAL SPECIFICATION

Publicly Available Specification (PAS); O-RAN Fronthaul Control, User and Synchronization Plane Specification v07.02; (O-RAN-WG4.CUS.0-v07.02)

CAUTION

The present document has been submitted to ETSI as a PAS produced by O-RAN WG4 and approved by the ETSI Technical Committee MSG.

ETSI had been assigned all the relevant copyrights related to the document ORAN-WG4.CUS.0-v07.01 on an "as is basis". Consequently, to the fullest extent permitted by law, ETSI disclaims all warranties whether express, implied, statutory or otherwise including but not limited to merchantability, non-infringement of any intellectual property rights of third parties. No warranty is given about the accuracy and the completeness of the content of the present document.

Reference

DTS/MSG-001140

Keywords

control, fronthaul, interface, PAS, synchronization

ETSI

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

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

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

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx.

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Copyright Notification

Contents

Intellectual Property Rights

Essential patents

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

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

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.

Foreword

This Technical Specification (TS) has been produced by O-RAN Alliance and approved by ETSI Technical Committee 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.

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 release date and an increase in version number as follows:

Release x.y.z

where:

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

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

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

The present document specifies the control plane, user plane and synchronization plane protocols used over the fronthaul interface linking the O-DU (O-RAN Distributed Unit) with the O-RU (O-RAN Radio Unit) with a Lower Layer Functional Split-7-2x based architecture (explained below). The scope of this document includes both LTE and NR (5G). A separate document contains the O-RAN M-Plane (management plane) specification.

In the following, “Layer 1” and “Physical Layer” are assumed to be synonymous.

In the main body of this specification (in any “clause”) the information contained therein is normative meaning binding on any compliant system, unless explicitly described as informative (a capability described as “optional” may or may not be included in a compliant system but if it is included it shall comply with the optional capability description). Information contained in an “Annex” to this specification is always informative.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document in Release 15.

Referenced documents which are not found to be publicly available in the expected location might be found at https://o-ran.org/Reference.

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

The following referenced documents are necessary for the application of the present document.

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

[2] eCPRI Transport Network V1.2 (2018-06-25) "Common Public Radio Interface: Requirements for the eCPRI Transport Network".

[3] IEEE Std 1588-2008 "Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems".

[4] 3GPP TS 38.211 V15.1.0: "NR; Physical channels and modulation".

[5] R1-1800296, "NR OFDM Symbol Generation Option Analysis", Intel, 3GPP TSG RAN WG1 AH#18-01, Vancouver, Canada, Jan. 22-26, 2018

[6] R1-1800802, "OFDM signal generation", Nokia, 3GPP TSG RAN WG1 AH#18-01, Vancouver, Canada, Jan. 22-26, 2018

[7] ORAN-WG4.MP.0-v07.01: "O-RAN Fronthaul Working Group Management Plane Specification, Release 07.01".

[8] 3GPP TS 38.104 "Base Station (BS) radio transmission and reception", Release 15, v15.2.0 (2018-06).

[9] 3GPP TS 36.104 "Base Station (BS) radio transmission and reception", Release 16, v16.7.0 (2020-07).

[10] RFC 1166: "Internet Numbers".

[11] "IEEE Standard for Local and metropolitan area networks -- Time-Sensitive Networking for Fronthaul," in IEEE Std 802.1CM-2018, 8 June 2018.

[12] "IEEE Standard for Local and metropolitan area networks -- Time-Sensitive Networking for Fronthaul - Amendment 1: Enhancements to Fronthaul Profiles to Support New Fronthaul Interface, Synchronization, and Syntonization Standards," in IEEE Std 802.1CMde-2020 (Amendment to IEEE Std 802.1CM-2018, Oct. 2020.

[13] 3GPP. TR 38.801 V14.0.0: "Study on new radio access technology: Radio access architecture and interfaces".

[14] 3GPP TS 36.213: "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures".

[15] ITU-T G.781 (04/2020): "Synchronization layer functions for frequency synchronization based on the physical layer".

[16] ITU-T G.810 (08/1996): "Definitions and terminology for synchronization networks ".

[17] ITU-T G.8260 (03/2020): "Definitions and terminology for synchronization in packet networks".

[18] ITU-T G.8261/Y.1361 (2019) Amendment 2 (10/2020) Timing and synchronization aspects in packet networks

[19] ITU-T G.8262/Y.1362 (2018) Amendment 1 (03/2020) Timing characteristics of a synchronous Ethernet equipment slave clock

[20] ITU-T G.8262.1/Y.1362 (2019) Amendment 1 (08/2019) Timing characteristics of an enhanced synchronous equipment slave clock

[21] ITU-T G.8264/Y.1364 (08/2017) Amendment 1 (03/2018) Distribution of timing information through packet networks

[22] ITU-T G.8271/Y.1366 (03/2020): "Time and phase synchronization aspects of telecommunication networks".

[23] ITU-T G.8271.1/Y.1366.1 (10/2020): "Network limits for time synchronization in packet networks with full timing support from the network".

[24] ITU-T G.8271.2/Y.1366.2 (05/2021): "Network limits for time synchronization in packet networks with partial timing support from the network".

[25] ITU-T G.8272/Y.1367 (11/2018) Amendment 1 (03/2020) Timing characteristics of primary reference time clocks

[26] ITU-T G.8272.1/Y.1367. (11/2016) Amendment 2 (08/2019) Timing characteristics of enhanced primary reference time clocks

[27] ITU-T G.8273/Y.1368 (2018) Corrigendum 1 (10/2020) Framework of phase and time clocks

[28] ITU-T G.8273.2/Y.1368.2 (10/2020): "Timing characteristics of telecom boundary clocks and telecom time slave clocks".

[29] ITU-T G.8273.3/Y.1368.3 (10/2017) Amendment. 1 (11/2018) Timing characteristics of telecom transparent clocks for use with full timing support from the network

[30] ITU-T G.8275/Y.1369 (2020) Amendment 1 (05/2021) Architecture and requirements for packet-based time and phase distribution

[31] ITU-T G8275.1/Y.1369.1 (2020) Amendment 2 (06/2021) Precision time protocol telecom profile for phase/time synchronization with full timing support from the network

[32] ITU-T G8275.2/Y.1369.2 (2020) Amendment 2 (06/2021) Precision time protocol telecom profile for time/phase synchronization with partial timing support from the network

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

2.2 Informative references

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

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 38.901: "Study on channel model for frequencies from 0.5 to 100 GHz".

[i.2] 3GPP TR 36.932: "Scenarios and requirements for small cell enhancements for E-UTRA and E-UTRAN".

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

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

Cascade mode: Mode of Shared cell which is realized by several O-RUs cascaded in chain. (See clause 13)

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. Includes the fields BandSector_ID, CC_ID, RU_Port_ID and DU_Port_ID (see clauses 5.1.3.1.6 and 5.1.3.2.4)

FFO: Fractional Frequency Offset. This is defined as Δf/f_norm which is used to describe frequency error, typically on the output of the T-TSC filter in the O-RU. It is the same as the Fractional Frequency Deviation defined in ITU-T G.810.

FHM mode: Mode of Shared cell which is realized by FHM and several O-RUs. (See clause 13)

Hop: physical link between 2 s-plane nodes (where node can be O-DU, switch or O-RU) as defined in IEEE 802.1CM

LAA: Licensed-assisted access: Carrier aggregation with at least one secondary cell operating in the unlicensed spectrum.

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.

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

O-CU: O-RAN Control Unit – a logical node hosting PDCP, RRC, SDAP and other control functions

O-DU: O-RAN Distributed Unit: a logical node hosting 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).

PLFS : Physical Layer Frequency Support, as per ITU-T G.8271.1. SyncE is one example.

Processing element: See clause 7.5 of the M-Plane specification [7] for a definition of processing element.

PRTC: Primary Reference Time Clock. There are different types of PRTC defined in both G.8272 and G.8272.1. Unless the specific type is indicated, a reference to a PRTC in this document could include any of these types.

Q<I.F>: denotes a signed two's-complement I+F bit fixed point number with I signed integer bits, and F fractional bits.

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. (See clause 11)

Slot: this is a group of 14 symbols, for LTE and NR. LTE has a separate slot definition within 3GPP which is 7 symbols but that definition is not used in this specification. So for NR, "slot" in this document means slot as 3GPP defines it, but for LTE "slot" in this document correlates to the LTE "TTI" as defined by 3GPP.

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.

T-TSC: Telecom Subordinate Clock; this is what ITU-T standards refer to as a "Telecom Slave Clock"

TAE: Time Alignment Error as defined by 3GPP TS36.104 and TS38.104

TE(*t*): Time Error – The difference between the time indicated by a clock or timing signal, and that indicated by a reference clock or timing signal. See ITU-T G.810, clause 4.5.13.

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

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

UL: UpLink : data flow away from the radiating antenna (generally on the LLS interface)

UNI: User Network Interface as defined by eCPRI network requirement specification

UQ<I.F>: denotes an unsigned I+F bit fixed point number with I unsigned integer bits and F fractional bits

3.2 Symbols

For the purposes of the present document, the following symbols apply: None.

3.3 Abbreviations

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

eNB	e NodeB (applies to LTE)
FHM	Fronthaul Multiplexer
gNB	g NodeB (applies to NR)

O-DU	O-RAN Distributed Unit (see clause 3.1)
O-RU	O-RAN Radio Unit
SMO	Service Management and Orchestration
BMCA	Best Master Clock Algorithm
CA	Carrier Aggregation
eEEC	enhanced Ethernet Equipment Clock
ePRTC	enhanced Primary Reference Time Clock
OTA	Over the Air
OTDOA	Observed Time Delay Of Arrival
PRC	Primary Reference Clock
PTP	Precision Time Protocol
SYNCE	Synchronous Ethernet
T-BC	Telecom Boundary Clock
T-GM	Telecom Grand Master

3.4 Document Conventions

3.4.1 Terminologies

For the purposes of the present document, the following terminologies apply.

ignored Within this specification, it is sometimes stated that certain information elements are to be "ignored" by the receiver (generally for "reserved" fields but in some cases other fields too). In this case, the fields shall be ignored for the purposes of processing at the O-RAN application level, but in the case of certain packet error-checking such as Ethernet-layer CRC or parity calculations, the fields shall be included in the CRC or parity calculations. As well, in the event of packet encryption, the "ignored" fields shall be encrypted along with the other packet payload information.

3.4.2 Fields and Bitmasks in Messages

For the purposes of the present document, the following conventions apply for the format of messages and data structures within messages.

In accordance with RFC 1166 [10], the left most bit of an octet is the most significant bit (msb) and the right most bit is the least significant bit (lsb). The msb is labelled as 0 and lsb is labelled as 7. This is illustrated by a blue ribbon in tables showing the message format. An example is depicted in Table 3-1. Note that this bit labelling convention (specifically the blue ribbon header in some tables) for octets is different from the labelling of bits within a field (bracketed bit numbers shown in tables).

To address specific bits within a field, the following notation is used: "X[k]" represents k^{th} bit in a field X with the convention that bit X[0] is the least significant bit of field X and located in the rightmost bit position. Where applicable, a sequence of bits in a field X can be interpreted as an unsigned integer value X_{val} calculated with formula:

$$X_{val} = \sum_{k=0}^{N-1} X[k] 2^\wedge k$$

where N is number of bits in field X.

Notation "X[a:b]" represents a sequence of bits in field X starting from bit X[a] and ending at bit X[b] inclusive, where a > b.

Table 3-1 below illustrates the format of messages and data structures using this notation.

Table 3-1 Example of table presenting format of a data structure

0 (msb)	1	2	3	4	5	6	7 (lsb)	# of bytes	octet
Y[3:0]				X[11:8]				1	N
X[7:0]								1	N+1

The example data structure presented in Table 3-1 is interpreted as follows:

- Field Y has 4 bits. Bits Y[3] to Y[0] of field Y are in octet N; Y[3], the most significant bit of field Y is in the most significant bit of octet N.
- Field X has 12 bits. Bits X[11] to X[8] of field X are in octet N; X[8] is in the least significant bit of octet N. Bits X[7] to X[0] of field X are in octet N+1; X[7] is in the most significant bit of octet N+1 and X[0] is in the least significant bit of octet N+1.

This corresponds to a structure that maps every bit as presented in Table 3-2 below.

Table 3-2 Interpretation of the example of table presenting format of a data structure

0 (msb)	1	2	3	4	5	6	7 (lsb)	# of bytes	octet
Y[3]	Y[2]	Y[1]	Y[0]	X[11]	X[10]	X[9]	X[8]	1	N
X[7]	X[6]	X[5]	X[4]	X[3]	X[2]	X[1]	X[0]	1	N+1

4 Architecture & Requirements

4.1 Architectural aspects

The architecture of eNB or gNB with O-DU and O-RUs is shown in **Figure 4-1**. LLS-C and LLS-U provide C-plane and U-plane over LLS interface, respectively.

In this architecture, O-DU and O-RU can be defined as follows.

Lower Layer Split Central Unit (O-DU): a logical node that includes the eNB/gNB functions as listed in clause 4.1.1 split option 7-2x, excepting those functions allocated exclusively to the O-RU. The O-DU controls the operation of O-RUs.

O-RAN Radio Unit (O-RU): a logical node that includes a subset of the eNB/gNB functions as listed in clause 4.1.1 split option 7-2x. The real-time aspects of control & user plane communication with the O-RU are controlled by the O-DU.

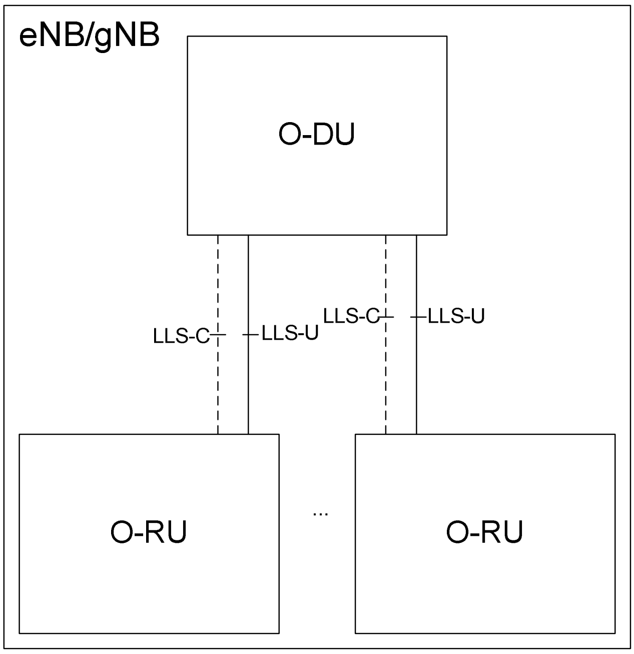


Figure 4-1 : eNB/gNB architecture with O-DU and O-RUs

4.1.1 Functional Split

When considering the functional split defining a fronthaul interface there are two competing interests:

a) There is a benefit in keeping an O-RU as simple as possible because size, weight, and power draw are primary deciding considerations and the more complex an O-RU, the larger, heavier and more power-hungry the O-RU tends to be;

b) There is a benefit in having the interface at a higher level which tends to reduce the interface throughput relative to a lower-level interface – but the higher-level the interface, the more complex the O-RU tends to be.

To resolve this conundrum, O-RAN has selected a single split point, known as “7-2x” but allows a variation, with the precoding function to be located either “above” the interface in the O-DU or “below” the interface in the O-RU. For the most part the interface is not affected by this decision, but there are some impacts namely to provide the necessary information to the O-RU to execute the precoding operation. O-RUs within which the precoding is not done (therefore of lower complexity) are called “Category A” O-RUs while O-RUs within which the precoding is done are called “Category B” O-RUs. See **Figure 4-2** for a depiction of this dual O-RU concept.

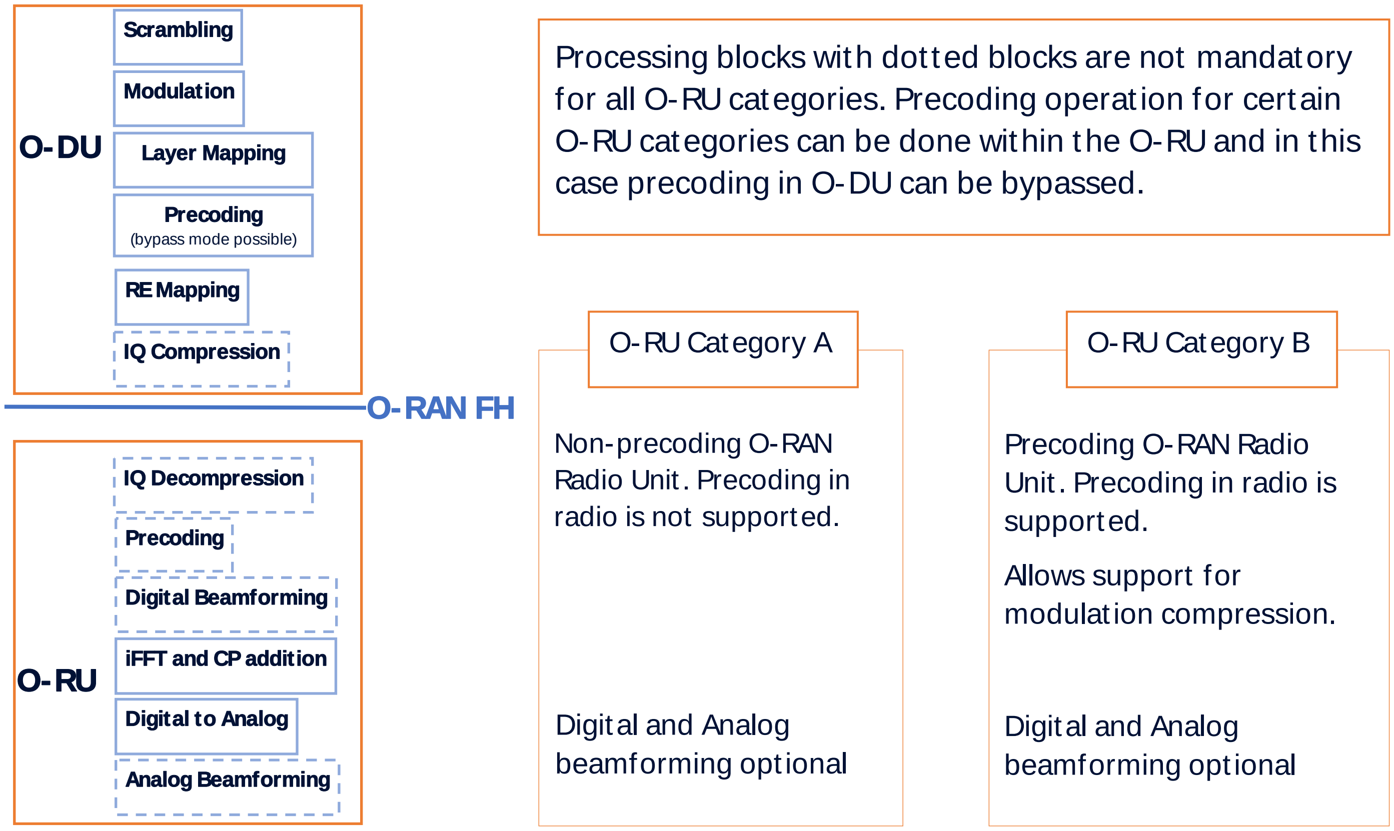

O-DU
Scrambling
Modulation
Layer Mapping
Precoding
(bypass mode possible)
RE Mapping
IQ Compression
O-RAN FH
O-RU
IQ Decompression
Precoding
Digital Beamforming
iFFT and CP addition
Digital to Analog
Analog Beamforming
Processing blocks with dotted blocks are not mandatory for all O-RU categories. Precoding operation for certain O-RU categories can be done within the O-RU and in this case precoding in O-DU can be bypassed.
O-RU Category A
Non-precoding O-RAN Radio Unit. Precoding in radio is not supported.
Digital and Analog beamforming optional
O-RU Category B
Precoding O-RAN Radio Unit. Precoding in radio is supported.
Allows support for modulation compression.
Digital and Analog beamforming optional

Figure 4-2 : Split Point and Category A and Category B O-RAN Radio Units

The inclusion of these two O-RU categories has certain implications for the LTE and NR functional splits in both DL and UL which are reflected in the following clauses. In particular, for a Category B O-RU to implement precoding for LTE TM2-TM4 some special C-Plane instructions need to be provided to the O-RU from the O-DU; this is described in a later clause of this document. For LTE TM5-10 and NR, no special instructions are needed because the precoding may be included in a digital beamforming processing block within the O-RU for a Category B O-RU (even for analog beamforming O-RUs), while for a Category A O-RU, the precoding would be executed in the O-DU and any beamforming in the O-RU, if present, would exclude the precoding calculation.

4.1.2 Selected Split 7-2x (DL)

DL functional split for various physical layer channels is illustrated in **Figure 4-3** (LTE Category A O-RUs), **Figure 4-4** (LTE Category B O-RUs), **Figure 4-5** (NR Category A O-RUs), and **Figure 4-6** (NR Category B O-RUs).

When O-RU Category A is supported by O-DU it is mandatory to support a total number of precoded streams of up to 8. Support for more than 8 precoded streams is optional.

For LTE (e.g. TM9) and NR PDSCH with UE specific reference signals, the DL processing chain specified by 3GPP does not include a precoding operation. The detailed precoding operation referred to in **Figure 4-3** and **Figure 4-4** for PDSCH with UE specific reference signals is not further described within this specification.

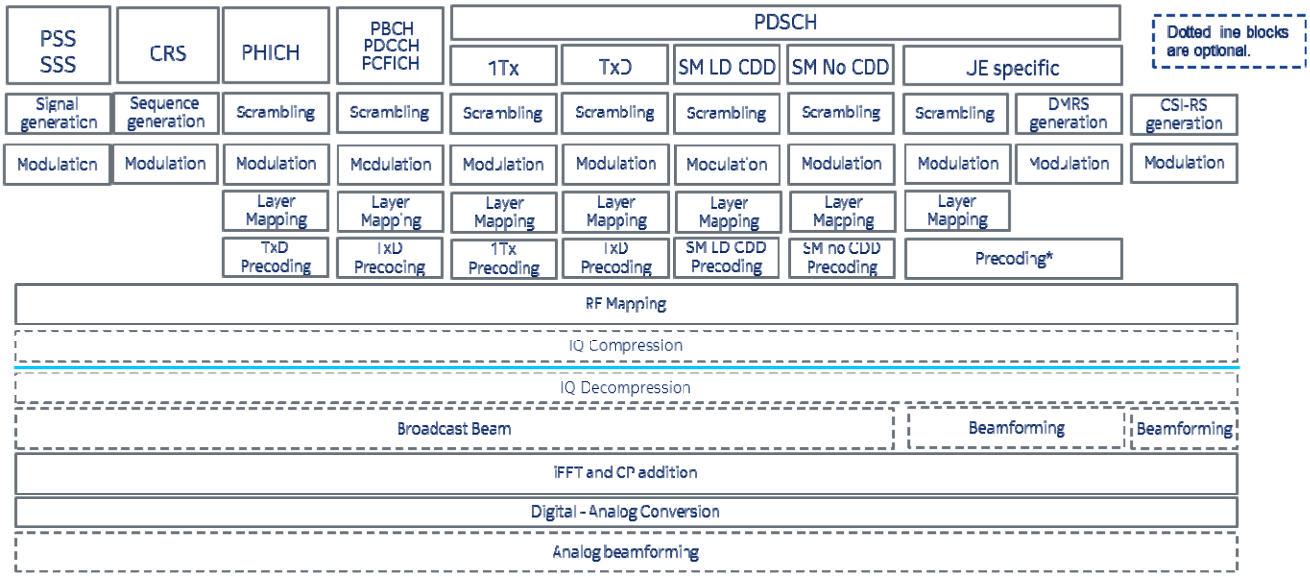


Figure 4-3 : Lower layer DL split description, LTE, Category A O-RUs

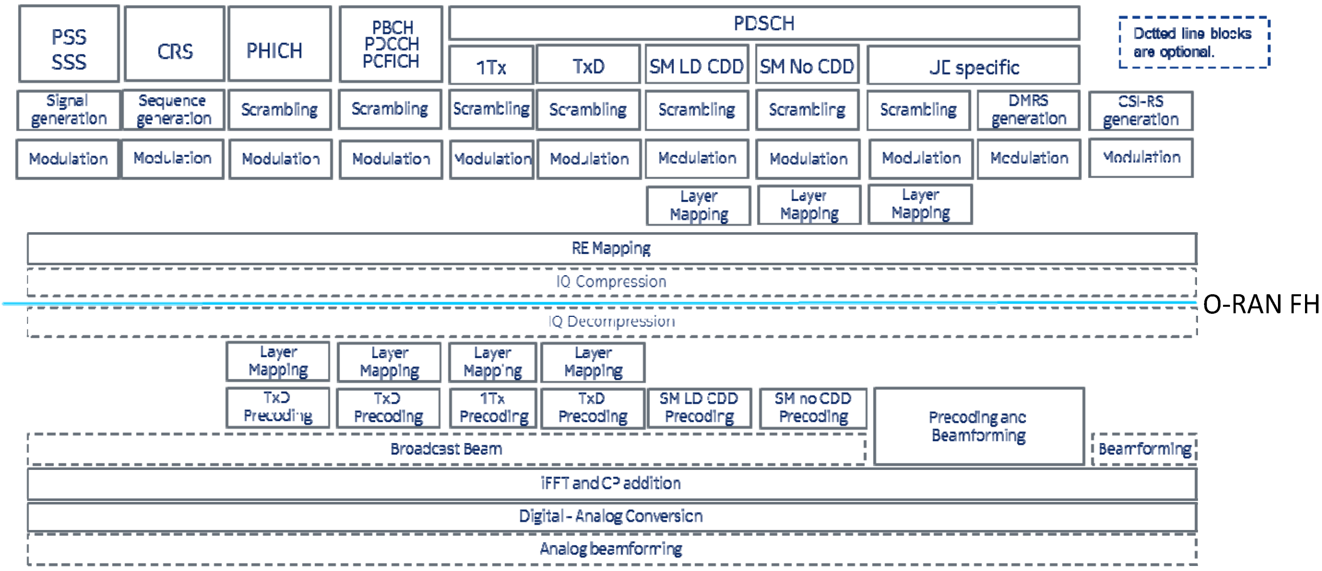


Figure 4-4 : Lower layer DL split description, LTE, Category B O-RUs

Although not explicitly indicated in Figure 2-4, RE mapping operation is separated into two parts for precoding support at O-RU for transmit diversity and spatial multiplexing modes in Category B radios:

1) RE mapping to frequency resources is performed at O-DU

2) RE mapping to antenna ports is performed at O-RU after precoding

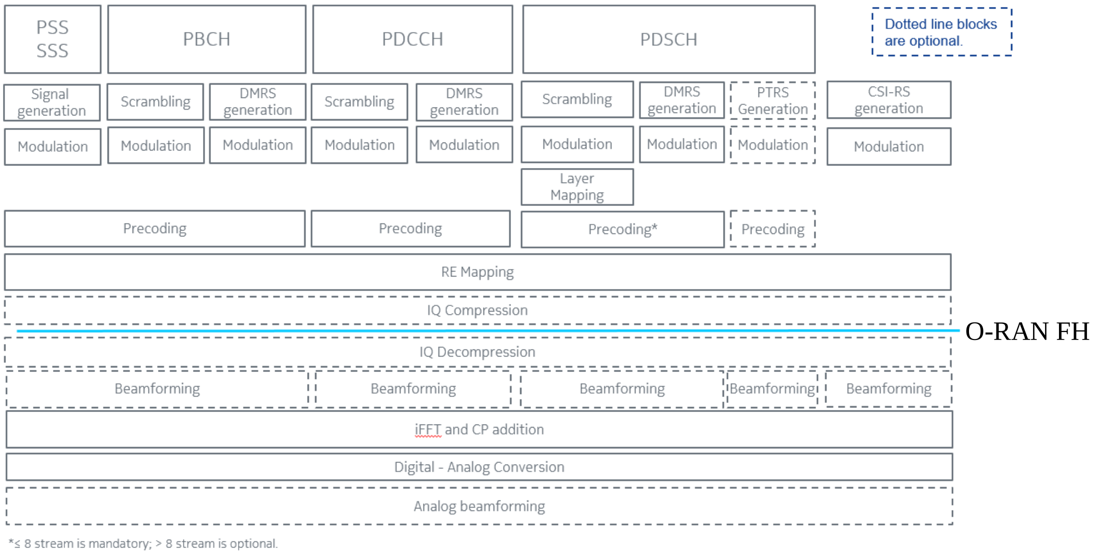


Figure 4-5 : Lower layer DL split description, NR, Category A Radio

NOTE: the above figure illustrates some of the high level functionalities inside the O-RU and O-DU, and do not bind the internal design of each of the O-RU and O-DU

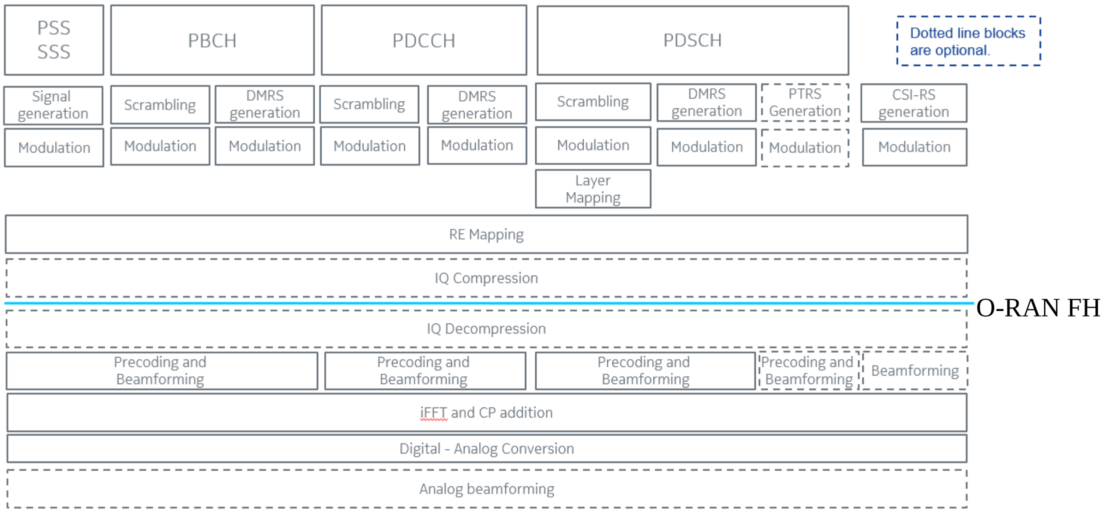


Figure 4-6 : Lower layer DL split description, NR, Category B Radio

NOTE: the above figure illustrates some of the high level functionalities inside the O-RU and O-DU, and do not bind the internal design of each of the O-RU and O-DU

4.1.2.1 Description

In the DL, OFDM phase compensation (clause 5.4 of [4]), iFFT, CP addition, and digital beamforming functions reside in the O-RU as well as precoding for Category B O-RUs. The rest of the PHY functions including resource element mapping, precoding, layer mapping, modulation, scrambling, rate matching and coding reside in the O-DU. Precoding shall be in the O-DU for Category A O-RU support but this only applies if the number of precoder output spatial streams is 8 or less, otherwise precoding shall be in the (Category B) O-RU.

The option of including some PHY functionality in a radio unit was not included in [13].

4.1.2.2 Benefits and Justification

- Interface simplicity: Transfer of user plane data is based on Resource Elements / Physical Resource Blocks, which simplifies the data mapping and limits the required associated control messages
- Transport Bandwidth Scalability: Lower split options (e.g., splits 7-1 and 8) scale based on number of antennas. In contrast, 7-2x interface scales based on “streams”, which allows using high number of antennas without higher transport bandwidth. Further, user data transfer can be optimized to send only PRBs that contain user data for purpose of reducing transport bandwidth
- Beamforming Support: The same interface design can support different beamforming techniques (digital, analog, hybrid) as well as different beamforming algorithms. Likewise, deployments using only analog beamforming are also possible with the same interface design.
- Interoperability: Less user specific parameters are used at split 7-2x (when compared to higher split options), which can simplify specification.
- Advanced receivers and inter-cell coordination: this option allows implementation of advanced receivers and coordination features, which are also easier to implement and less restricted when most functions are placed at the O-DU. For example, UL CoMP is not possible when the UL upper-PHY processing is in the O-RU.
- Lower O-RU complexity: Less functions at O-RU (when compared to higher split options) allow limiting the number of required real time calculations as well as required memory requirement, especially for Category A O-RUs.
- Future proof-ness: Placing most functions at O-DU will allow introduction of new features via software upgrades without inflicting HW changes at O-RU (e.g., specification changes due to URLLC or new modulation schemes).
- Interface and functions symmetry: If the same interface and split point is used for DL and UL, specification effort can be reduced.

4.1.3 Selected Split 7-2x (UL)

UL functional split for various physical layer channels and transmission modes are illustrated in **Figure 4-7**. Likewise, digital beamforming in this context, is the function of antenna port selection or antenna port combining.

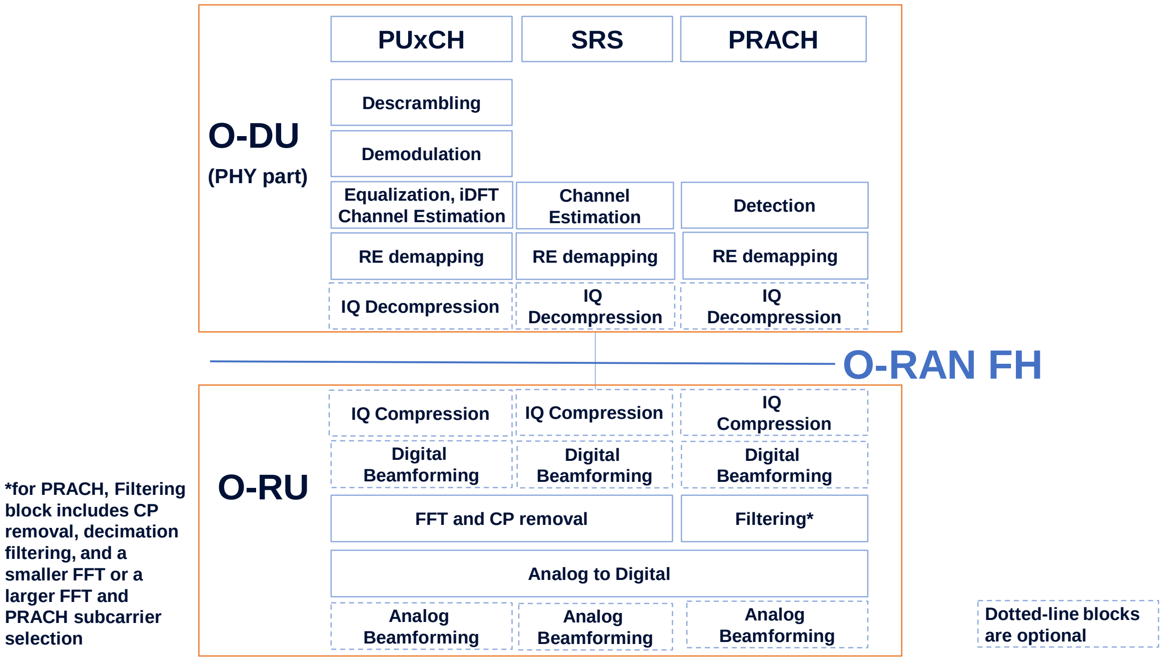


Figure 4-7 : Lower layer UL split description for LTE and NR

NOTE: the above figure illustrates some of the highlevel functionalities inside the O-RU and O-DU, and do not bind the internal design of each of the O-RU and O-DU.

4.1.3.1 Description

In the UL, OFDM phase compensation (for all channels except PRACH when Section Type “3” message is used) according to clause 5.4 of [4], FFT, CP removal and digital beamforming functions reside in the O-RU. The rest of the PHY functions including resource element de-mapping, equalization, de-modulation, de-scrambling, rate de-matching and de-coding reside in the O-DU.

The option of including some PHY functionality in a radio unit was not included in [13].

4.1.3.2 Benefits and Justification

The benefits defined for Option 7-2x for DL are also applicable for Uplink (See clause 4.1.2.2).

4.2 Data Flows

An overall reference of required inputs for some major functional blocks, their granularity and input originating source is compiled in **Table 4-1** and **Table 4-2** for DL and UL respectively.

Table 4-1 : Required information for each functional block (DL)

Function	Required Information	Signaling Granularity	Information source
Digital beamforming	• Digital beamforming information • Number of TRXs or Beams	Per RE per beam per slot	Scheduler (O-DU)
iFFT and CP addition	• FFT size • SC spacing • CP length	Per TRX per slot	Scheduler (O-DU)
Analog beamforming	• Analog beamforming info	Per frequency band per slot	Scheduler (O-DU)
Listen Before Talk (LBT)	• LBT parameters	Per MCOT	Scheduler (O-DU)

Table 4-2 : Required Information for each functional block (UL)

Function	Required Information	Signaling Granularity	Information source
Digital beamforming	• Digital beamforming information • Number of TRXs or Beams	Per RE per TRX per slot	Scheduler (O-DU)
FFT and CP removal	• FFT size • SC spacing • CP length	Per TRX per slot	Scheduler (O-DU)
Analog beamforming	• Analog beamforming info	Per frequency band per slot	Scheduler (O-DU)

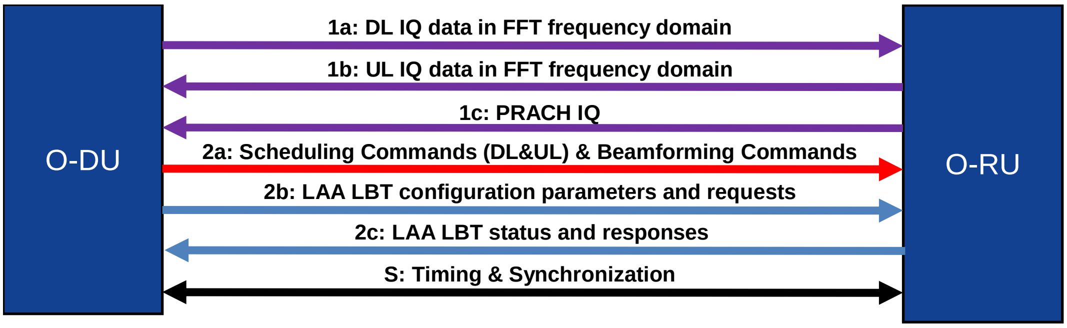
In case of lower layer fronthaul based on split option 7-2x for DL and UL, the required external data flows (excluding M-plane) to exchange information between O-DU and O-RU can be categorized as follows. Further, the data flow contents mapping is described in **Table 4-3.**

- User-Plane

- Data Flow 1a: Flows of IQ Data in FFT frequency domain on DL
- Data Flow 1b: Flows of IQ Data in FFT frequency domain on UL
- Data Flow 1c: Flow of PRACH IQ data in FFT frequency domain

- C-Plane
  - Data Flow 2a: Scheduling commands (DL and UL) & Beamforming commands
  - Data Flow 2b: LAA Listen-Before-Talk (LBT) configuration commands and requests
  - Data Flow 2c: LAA LBT status and response messages

- Synchronization-Plane
  - Data Flow S: Timing and Synchronization data


O-DU
1a: DL IQ data in FFT frequency domain
1b: UL IQ data in FFT frequency domain
1c: PRACH IQ
2a: Scheduling Commands (DL&UL) & Beamforming Commands
2b: LAA LBT configuration parameters and requests
2c: LAA LBT status and responses
S: Timing & Synchronization
O-RU

Note: M-Plane flows not represented here

Figure 4-8 : Lower layer fronthaul data flows

Table 4-3 : Data Flow Information Mapping

Plane	ID	Name	Contents	Periodicity
U-Plane	1a	DL Frequency Domain IQ Data	DL user data (PDSCH), control channel data (PDCCH, etc.), …	< slot
	1b	UL Frequency Domain IQ Data	UL user data (PUSCH), control channel data (PUCCH, etc.), …	< slot
	1c	PRACH Frequency Domain IQ Data	UL PRACH data	< slot
C-Plane	2a	Scheduling Commands & Beamforming Commands	Scheduling information, FFT size, CP length, Subcarrier spacing, UL PRACH scheduling DL and UL Beamforming commands (e.g., beam index) and scheduling	~ slot
	2b	LAA LBT configuration parameters and requests	LBT Configuration parameters such as lbtHandle, lbtDeferFactor, lbtBackoffCounter, lbtOffset, MCOT, lbtMode, sfnSf, lbtCWconfig_H, lbtCWconfig_T, lbtTrafficClass.	per MCOT/DRS
	2c	LAA LBT status and responses	LBT DL indication parameters such as lbtHandle, lbtResult, initialPartialSFs, bufferError, lbtCWR_Result	
S-Plane	S	Timing and Synchronization	SyncE SSM & IEEE 1588 PTP packets	

4.3 Latency Requirements

Intra-PHY lower layer fronthaul split has characteristic of a stringent bandwidth and tight latency requirement. This implies use of a special “Fronthaul Service Profile” to be supported by the transport network, and which may differ depending on the operating environment, topology and target use cases. The general concept and latency model is based on eCPRI reference points for delay management definitions (See **Figure 4-9**). However, this specification provides additional details for both DL and UL (**Table 4-4**) latency. The delay parameters and how these are determined are explained below; note that in general it is expected a separate set of timing parameters applies to each O-RU attached to an O-DU. This clause defines multiple approaches which can be used to provide an inter-operable timing solution.

The reference points defined for eCPRI are reflected below in **Figure 4-9.** The reference points are:

- O-DU: R1/ R4 – Transmit/ Receive interface at O-DU
- O-RU: R2/ R3 – Receive/ Transmit interface at O-RU
- Ra: Antenna interface at O-RU

NOTE: When an external antenna is used with a cable imposing a negligible delay, then the O-RU connector can be assumed as Ra.

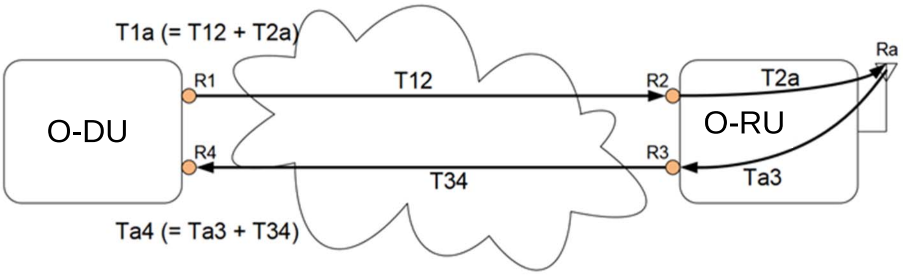


Figure 4-9 : Definition of reference points for delay management (adapted from [2])

Transmission delay between O-DU and O-RU are specified as T12 (downlink) and T34 (uplink). The transmission delay encompasses only the time from when a bit leaves the sender (R1/ R3) until it is received at the receiver (R2/ R4). In an ethernet transport network, these delays may not be constant due to switching delays (i.e. PDV). To account for this, transport delay shall be considered as a range with upper and lower bounds:

- Downlink transport delay: T12min/ T12max
- Uplink transport delay: T34min/ T34max

However, fixed timing at Ra is still required. Therefore, Ra is used as a reference point for delay management in the eCPRI model. Therefore, transmission and reception at the reference points are measured relative to Ra, resulting in the following parameters:

Table 4-4 : eCPRI O-DU/ O-RU Delay Model Latency Parameters

		eCPRI	Latency	eCPRI	
				Minimum	Maximum
Downlink	O-DU	T1a	Measured from output at O-DU (R1) to transmission over the air.	T1amin	T1amax
	O-RU	T2a	Measured from reception at O-RU (R2) to transmission over the air	T2amin	T2amax
Uplink	O-DU	Ta4	Measured from reception at O-RU antenna to reception at O-DU port (R4).	Ta4min	Ta4max
	O-RU	Ta3	Measured from reception at O-RU antenna to output at O-RU port (R3)	Ta3min	Ta3max

It is expected that a common timescale is used on both ends of the link. The relative time error of the S-plane measurement signals between the O-DU and O-RU, for the purposes of latency requirements management, should be within a limit of 3µs (±1.5 µsec).

- This requirement allows for the measurement of the transport delay (i.e., between O-DU and O-RU UNIs) with a measurement error that is sufficiently lower than the delay that is to be measured (e.g., 10 times lower).
- In case of LLS-C1 and LLS-C2, due to clause 11.3.2 requirements, the relative time error of the S-plane measurement signals between the O-DU and O-RU would always be within the limit of 3µs (±1.5 µsec).

- In case of LLS-C3 and LLS-C4, the synchronization network would need to meet G.8271.1 or G.8271.2 network limits in order to meet similar requirements (relative time error of the S-plane measurement signals between the O-DU and O-RU within a limit of 3µs).
- The upper bound on the absolute time error requirement at the O-DU S-Plane is dictated by the O-RU's receive window, the delay and PDV in the transport network and the O-DU's internal delays.

4.3.1 Timing Parameter Relationships

To ensure proper reception of data at the receiver over the packet interface, there are several relationships between the parameters defined above which shall be met. First it is important to understand the nature of the transmission itself. In either direction (downlink/ uplink) it takes some amount of time for the sender to actually transmit the packets onto the transmission media. However, the amount of data for any interval (e.g symbol) can vary thus resulting differing transmit times. This transmission time can be affected by several factors including (but not limited to) transport media rate, air interface bandwidth, and amount of data compression. The maximum amount of time allowed for the transmitter to send all data for an interval (Transmission Window) is defined by T1amax – T1amin. It is important to note that this is the allowed time, based on transport and O-RU characteristics. This is explained in greater detail, and its impacts on O-DU in clause 4.3.1.1.

To account for transport variation and transmission time the receiver implements a reception window. This allows packets containing samples for a specific symbol to be received within the window and still be transmitted at Ra at the required time. The size of the Reception Window shall account for both the maximum transmission time at the sender and the transport variation through the fronthaul network. The result is the first of the delay relationships which shall be met to ensure a working delay solution.

Reception Window ≥ Transmission Window + Transport Variation

Table 4-5 : eCPRI Delay Windows

	Reception Window	Transmission Window	Transport Variation
Downlink	T2amax – T2amin	T1amax – T1amin	T12max – T12min
Uplink	Ta4max – Ta4min	Ta3max – Ta3min	T34max – T34min

The position (in time) of the reception/ transmission windows at the O-RU is fixed relative to the air interface. However, the position of the corresponding windows at the O-DU is a function of the O-RU and transport parameters. For guaranteed reception of packets sent from O-DU to O-RU within the O-RU reception window, the following relationships shall also be met (refer to Annex B for explanation of these inequalities):

Table 4-6 : O-DU transmission/ reception window position (in time)

O-DU Timing		Parameter	O-DU Transmit Boundary Relationships
Downlink (Transmit)	No earlier than	T1amax	T1amax ≤ T2amax + T12min
	No later than	T1amin	T1amin ≥ T2amin + T12max
Uplink (Receive)	No earlier than	Ta4min	Ta4min ≤ Ta3min + T34min
	No later than	Ta4max	Ta4max ≥ Ta3max + T34max

4.3.1.1 O-DU Transmission Window

The U-Plane O-DU transmission window (T1amax – T1amin) is defined by the relationships above based on the O-RU reception window and max transport variation. It does not define the exact timing of transmission from the O-DU. Rather, it defines the boundaries that the U-Plane O-DU transmission shall operate within. The window merely represents the mathematical boundaries imposed on the O-DU as a result of the O-RU and Transport constraints. It is possible to define the constraints for any one of O-DU, transport and O-RU based on knowledge of the other two. However, typically the O-RU constraint is predefined based on the equipment and transport is part of the overall network goals.

The window resulting from the relationships shall be greater than or equal to the actual maximum time required by the O-DU to transmit all data for a symbol ($TXmax_{O\text{-}DU}$). That is, the window shall be at least large enough that the O-DU can transmit in the worst case within the window. Where, within the window, the O-DU transmits (e.g. beginning, middle, end) and how much of the window is consumed by the O-DU transmission is a matter of O-DU design.

The following downlink example illustrates the concept:

- O-RU parameters: T2amin = 100 usec, T2amax = 260 usec
- Transport Parameters (direct fiber of known length): T12min = 50 usec, T12max = 51 usec

The result indicates an O-DU transmission window as follows:

- T1amax ≤ 260 + 50
- T1amin ≥ 100 + 51

This provides a very large transmit window available to the O-DU. If, for example, the $TXmax_{O-DU}$ is only 30 usec, then the O-DU can determine where within the window to start its transmission, so long as the transmission completes prior to T1amin.

If, however, this same O-DU were paired with an O-RU with smaller reception window (e.g. T2amin = 100, T2amax = 150) using a transport network with the same T12min, but with 15 usec of PDV (T12max = 65), the result is:

- T1amax ≤ 150 + 50
- T1amin ≥ 100 + 65

The delay solution still works (200 – 165 ≥ 30), but with far less margin, and far less flexibility as to where within the window the O-DU may begin transmission.

The C-plane O-DU transmission window follows the same concept as the U-plane O-DU transmit window. However, the C-plane O-DU transmission window can be larger in size and hence, can start much earlier in time compared to the U-plane O-DU transmission window.

In certain scenarios such as sending the beamforming weights in real-time for mMIMO application using Section Extension 11, the C-plane messages need to be spread over time to avoid peak throughput. **Figure 4-10** illustrates this use case.

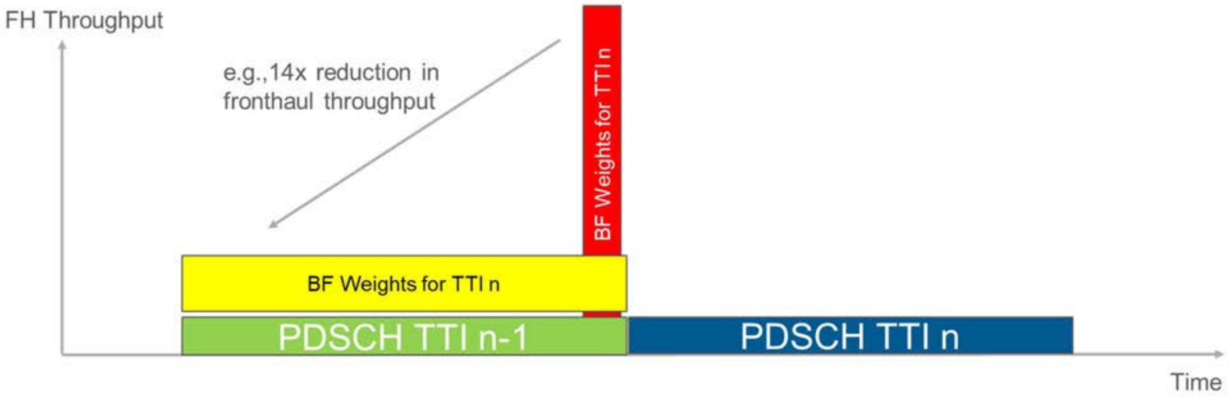


Figure 4-10 : Example of the benefit of using a longer C-Plane tx/rx window compared to the U-Plane windows

4.3.2 U-Plane/ C-Plane Timing

The basic delay parameters above describe the general delay model and characteristics of the O-RAN interface. However, the ORAN interface is divided into C-Plane and U-Plane parts. The C-Plane shall be available in order to process the corresponding U-Plane packets. To support coordination of C-Plane and U-Plane timing, the O-RAN interface specifies that C-Plane messages shall arrive at the O-RU some amount of time in advance (Tcp_adv_dl) of the latest possible time the first corresponding U-Plane messages may arrive. (Note that it is possible for Tcp_adv_dl to be set to 0, but the O-RAN delay model does not assume this.) As a result, O-RAN has defined the downlink timing relationships and parameters shown in **Figure 4-11**.

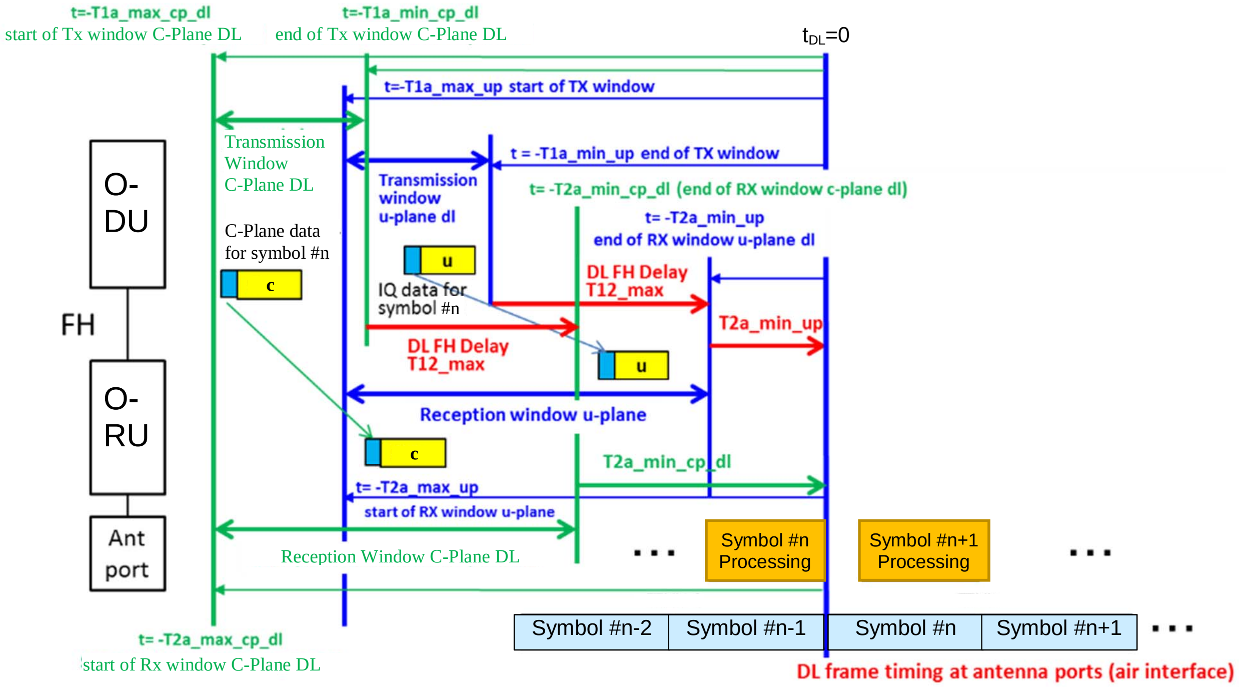

t=-T1a_max_cp_dl
start of Tx window C-Plane DL
t=-T1a_min_cp_dl
end of Tx window C-Plane DL
tDL=0
t=-T1a_max_up start of TX window
O-DU
Transmission Window C-Plane DL
t = -T1a_min_up end of TX window
Transmission window u-plane dl
t= -T2a_min_cp_dl (end of RX window c-plane dl)
t= -T2a_min_up
end of RX window u-plane dl
C-Plane data for symbol #n
u
c
IQ data for symbol #n
DL FH Delay T12_max
T2a_min_up
FH
DL FH Delay T12_max
u
O-RU
Reception window u-plane
c
T2a_min_cp_dl
t= -T2a_max_up
start of RX window u-plane
Ant port
Reception Window C-Plane DL
Symbol #n Processing
Symbol #n+1 Processing
Symbol #n-2
Symbol #n-1
Symbol #n
Symbol #n+1
t= -T2a_max_cp_dl
start of Rx window C-Plane DL
DL frame timing at antenna ports (air interface)

Figure 4-11 : Timing relations per symbol IQ in DL direction (U-Plane and C-Plane)

O-RAN does not assume that the transmission windows for C-Plane and U-Plane are of the same size. Tcp_adv_dl allows for different alignment of the respective transmission/ reception windows as illustrated in the figure. The reference (denoted as "t_{DL}=0") is the transmission of the earliest IQ sample in time domain within a symbol (including cyclic prefix) which is generated from the IQ data received in a U-plane messages specific to a symbol identified by **symbolId** (optionally adjusted by **timeOffset**).

A downlink C-plane message can refer to one or more symbols; transmission and reception windows for a downlink C-plane message referencing multiple symbols are relative to the start of the earliest symbol referenced by the message (identified by **startSymbolId**; optionally adjusted by **timeOffset**).

For uplink, the C-Plane is sent from O-DU to O-RU, while the U-Plane is from O-RU to O-DU. Since uplink C-Plane is from O-DU to O-RU, it uses the R1/ R2 reference points, and corresponding timing parameters. However, the uplink C-Plane timing is not characterized relative to the downlink C-Plane or U-Plane timing, but relative to Ra. The resulting O-RAN delay model for non-PRACH channels is shown in **Figure 4-12**.

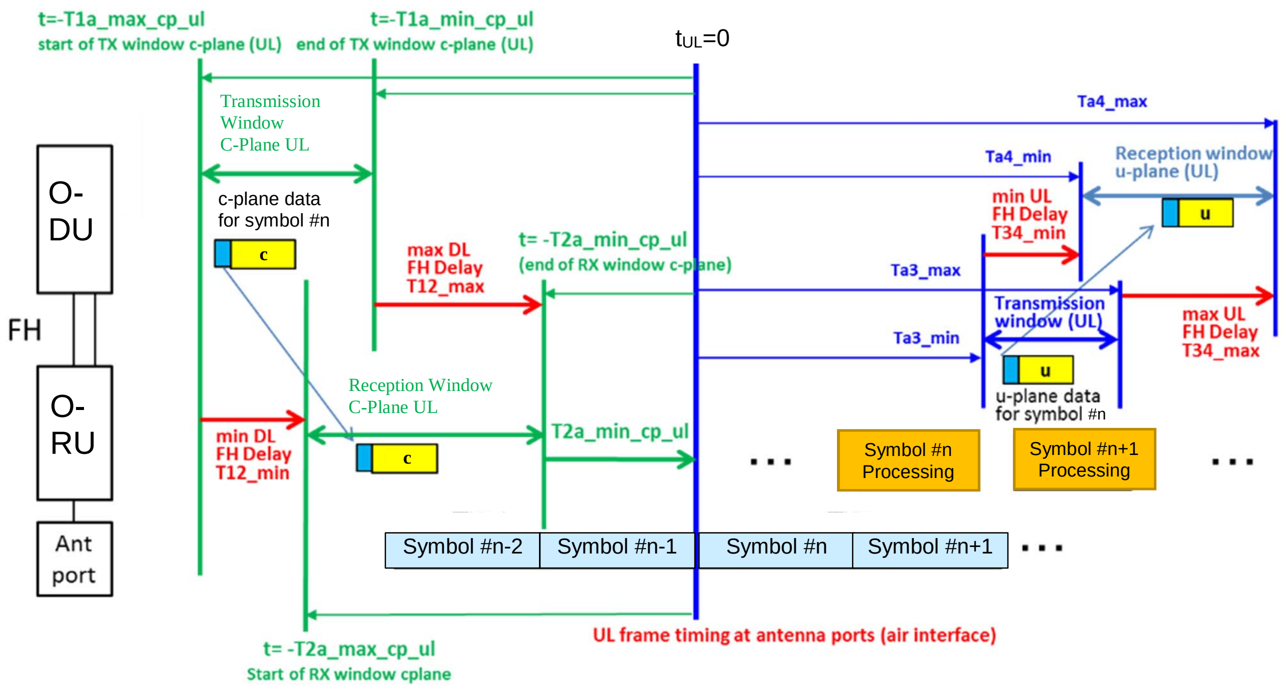


Figure 4-12 : Timing relations per symbol IQ in UL direction (U-Plane and C-Plane)

An uplink C-plane message not describing PRACH channels can refer to one or more symbols. The reference point for uplink C-plane message (denoted as "t_{UL}=0") is the reception of the earliest IQ sample in time domain (start of cyclic prefix) that is specific to the earliest symbol referenced by the message (pointed by **startSymbolId**; if Section Type 3 message is used, **timeOffset** will also point to this reference point).

The reference point for uplink U-plane messages related to non-PRACH channels (denoted as "t_{UL}=0") is the reception of the earliest IQ sample in time domain within a symbol (start of cyclic prefix) referenced by the message (pointed by **symbolId**; if Section Type 3 message is used, **timeOffset** will also point to this reference point). Note that "t_{UL}=0" is not affected by FFT sampling offset configured over M-plane since configured FFT sampling offset cannot be longer than CP-length.

Due to the characteristics regarding CP-length and symbol-length of PRACH-symbols, PRACH is handled in a separate way. Especially for the long preamble PRACH formats 0..3 (see reference 3GPP TS 38.211) the length of both the CP and the length of the repeatable part is much larger than corresponding parts for e.g. PUSCH symbols. For the other PRACH formats (A1..C2) the length of the CP could be very long compared to the CP for the PUSCH symbols (up to roughly 14 times longer (C2)). The consequence of either long CP and/or long repeatable part of specific PRACH formats is that the values of Ta3_min and Ta3_max are increased. Due to this, PRACH CP duration and number of repetitive parts are excluded from the values for these parameters provided by O-RU.

The O-DU is aware of the characteristics of the PRACH format being used (e.g., length of CP, number of repetitions), and with this information the position and size of the reception window is calculated by the O-DU. The used values for Ta3_min and Ta3_max are the ones being retrieved from the O-RU for the SCS being used for PRACH.

An uplink C-plane message describing PRACH can refer to one or more repeatable parts of PRACH sequence (PRACH repetitions) depending on the PRACH format. For PRACH formats with repetition, only a single C-plane message is used to refer to all PRACH repetitions for a particular PRACH occasion.

The reference point for PRACH C-plane message is pointed by **startSymbolId** and:

- Is the reception of the earliest IQ sample in time domain (start of cyclic prefix) that is specific to the PRACH repetition referenced by the message;
    - This is allowed only for PRACH formats without repetitions and with only one PRACH occasion in a slot, e.g. Format 0;
    - In this case, **cpLength** is set to a non-zero value in the PRACH C-plane (Section Type 3) message, which should indicate the actual CP duration;
- Or is the start of the latest symbol timing that starts right at or before the reception of the earliest IQ sample in time domain (after PRACH cyclic prefix) that is specific to the earliest PRACH repetition referenced by the message;

- This is allowed for all PRACH formats;
- In this case, **cpLength** is set to zero in the PRACH C-plane (Section Type 3) message;
- If the SCS value provided by "frameStructure" is equal to or greater than 15 kHz, then the symbol timing used to determine **startSymbolId** is based on the numerology of the SCS value provided by "frameStructure". Else, if the SCS value provided by "frameStructure" is less than 15 kHz (e.g. for long preamble PRACH formats), then the symbol timing used to determine **startSymbolId** is based on the numerology of 15 kHz SCS;
- When **ul-fft-sampling-offset** for PRACH is configured (via M-plane) to a non-zero value, the timing of the earliest IQ sample is advanced by that value. However, the reference symbol timing that is used to determine the **startSymbolId** value in the PRACH C-plane message is unaffected by **ul-fft-sampling-offset**. Therefore, with a non-zero **ul-fft-sampling offset** for PRACH, it is possible that "the latest symbol timing that starts right at or before the reception of the earliest IQ sample in time domain (after PRACH cyclic prefix) that is specific to the earliest PRACH repetition" may change to an earlier symbol timing compared to the case where **ul-fft-sampling-offset** for PRACH is zero. So, the O-DU needs to take **ul-fft-sampling-offset** for PRACH into account when determining the **startSymbolId** value. However, it is noted that even if **ul-fft-sampling-offset** for PRACH is configured to a non-zero value, it is expected to be in the order of nano seconds (i.e., less than 1 microsecond).

The reference point for PRACH U-plane message is the reception of the earliest IQ sample in time domain within a PRACH repetition (after PRACH cyclic prefix) referenced by the message.

- For the first PRACH repetition:
    - If **cpLength** is set to a non-zero value in the PRACH C-plane (Section Type 3) message referencing the PRACH repetition, this reference point for PRACH U-plane message is pointed by **timeOffset** (pointing to the start of PRACH cyclic prefix) + **cpLength** – **ul-fft-sampling-offset** (for PRACH);
    - Else, if **cpLength** is set to zero in the PRACH C-plane (Section Type 3) message referencing the PRACH repetition, this reference point for PRACH U-plane message is pointed by **timeOffset** (pointing to the start of the PRACH repetition (after PRACH cyclic prefix)) – **ul-fft-sampling-offset** (for PRACH);
- For subsequent PRACH repetitions (for PRACH formats with repetition), this reference point for PRACH U-plane message is shifted by the PRACH sequence duration.

Figures 4-13–4-16 illustrate timing relationships for PRACH taking some specific PRACH formats (Format 0, Format 1, Format B4 and Format C0) as examples.

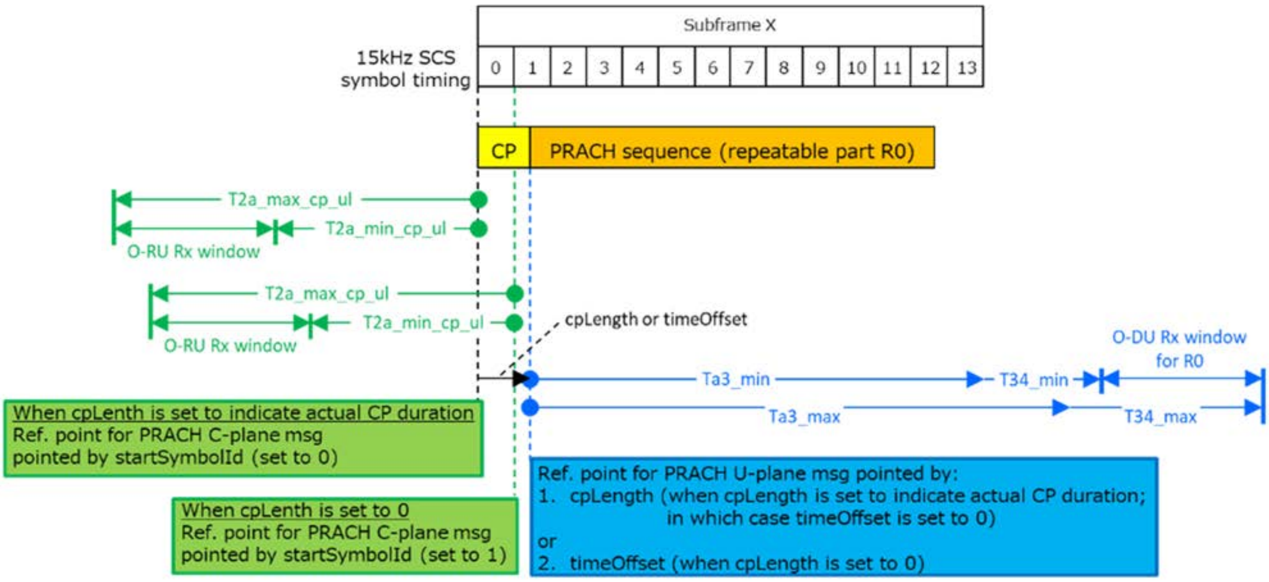

Subframe X
15kHz SCS symbol timing
0
1
2
3
4
5
6
7
8
9
10
11
12
13
CP
PRACH sequence (repeatable part R0)
T2a_max_cp_ul
T2a_min_cp_ul
O-RU Rx window
T2a_max_cp_ul
T2a_min_cp_ul
O-RU Rx window
cpLength or timeOffset
O-DU Rx window for R0
Ta3_min
T34_min
Ta3_max
T34_max
When cpLenth is set to indicate actual CP duration
Ref. point for PRACH C-plane msg pointed by startSymbolId (set to 0)
When cpLenth is set to 0
Ref. point for PRACH C-plane msg pointed by startSymbolId (set to 1)
Ref. point for PRACH U-plane msg pointed by:
1. cpLength (when cpLength is set to indicate actual CP duration; in which case timeOffset is set to 0)
or
2. timeOffset (when cpLength is set to 0)

Figure 4-13 : Timing relations for PRACH (Example 1: PRACH Format 0)

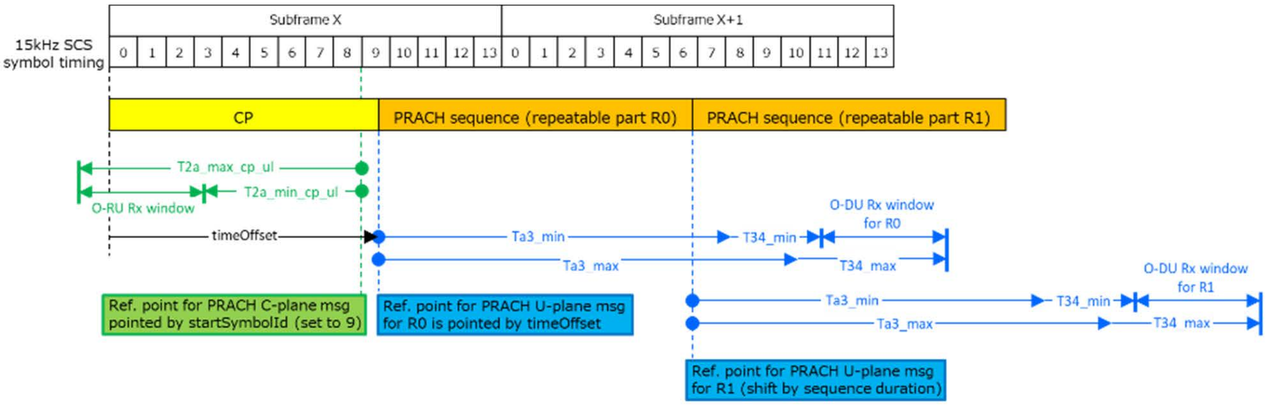

Subframe X
Subframe X+1
15kHz SCS symbol timing
0 1 2 3 4 5 6 7 8 9 10 11 12 13 0 1 2 3 4 5 6 7 8 9 10 11 12 13
CP
PRACH sequence (repeatable part R0)
PRACH sequence (repeatable part R1)
T2a_max_cp_ul
T2a_min_cp_ul
O-RU Rx window
timeOffset
Ta3_min
T34_min
O-DU Rx window for R0
Ta3_max
T34_max
O-DU Rx window for R1
Ta3_min
T34_min
Ta3_max
T34_max
Ref. point for PRACH C-plane msg pointed by startSymbolId (set to 9)
Ref. point for PRACH U-plane msg for R0 is pointed by timeOffset
Ref. point for PRACH U-plane msg for R1 (shift by sequence duration)

Figure 4-14 : Timing relations for PRACH (Example 2: PRACH Format 1)

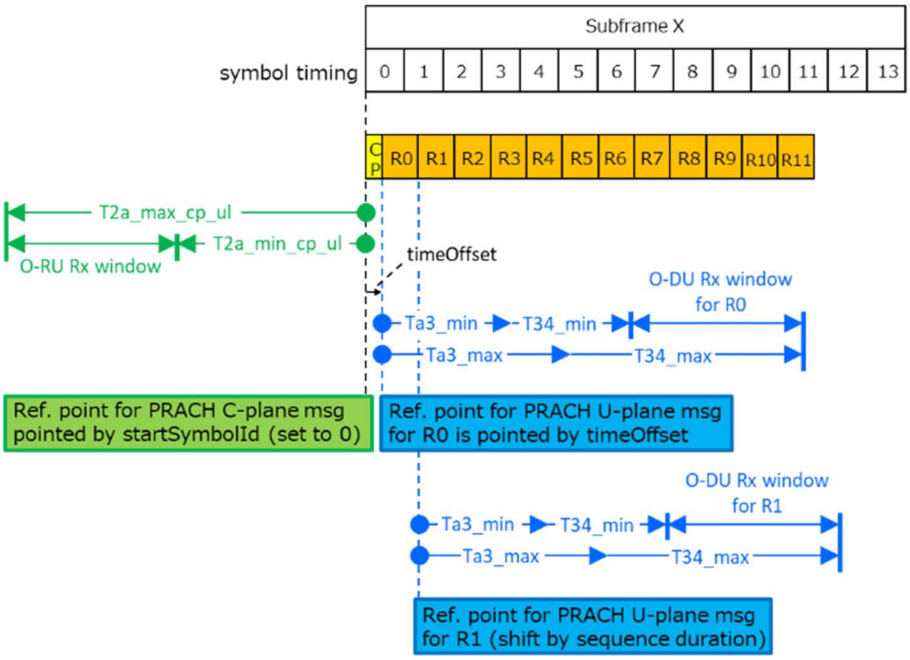

Subframe X
symbol timing
0 1 2 3 4 5 6 7 8 9 10 11 12 13
CP
R0 R1 R2 R3 R4 R5 R6 R7 R8 R9 R10 R11
T2a_max_cp_ul
T2a_min_cp_ul
O-RU Rx window
timeOffset
O-DU Rx window for R0
Ta3_min
T34_min
Ta3_max
T34_max
Ref. point for PRACH C-plane msg pointed by startSymbolId (set to 0)
Ref. point for PRACH U-plane msg for R0 is pointed by timeOffset
O-DU Rx window for R1
Ta3_min
T34_min
Ta3_max
T34_max
Ref. point for PRACH U-plane msg for R1 (shift by sequence duration)

Figure 4-15 : Timing relations for PRACH (Example 3: PRACH Format B4)

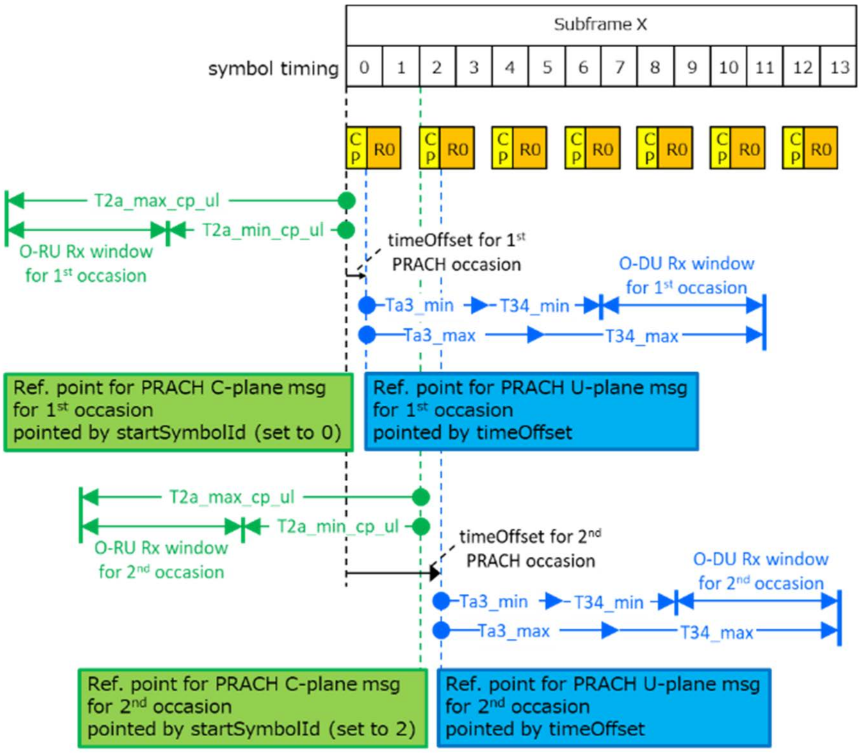

Subframe X
symbol timing
0 1 2 3 4 5 6 7 8 9 10 11 12 13
CP RO
T2a_max_cp_ul
T2a_min_cp_ul
O-RU Rx window for 1st occasion
timeOffset for 1st PRACH occasion
O-DU Rx window for 1st occasion
Ta3_min
T34_min
Ta3_max
T34_max
Ref. point for PRACH C-plane msg for 1st occasion pointed by startSymbolId (set to 0)
Ref. point for PRACH U-plane msg for 1st occasion pointed by timeOffset
T2a_max_cp_ul
T2a_min_cp_ul
O-RU Rx window for 2nd occasion
timeOffset for 2nd PRACH occasion
O-DU Rx window for 2nd occasion
Ta3_min
T34_min
Ta3_max
T34_max
Ref. point for PRACH C-plane msg for 2nd occasion pointed by startSymbolId (set to 2)
Ref. point for PRACH U-plane msg for 2nd occasion pointed by timeOffset

Figure 4-16 : Timing relations for PRACH (Example 4: PRACH Format C0)

The resulting O-RAN delay model parameters are summarized on **Table 4-7**. The respective sections within the table for O-RU and O-DU define the delay profile parameters for the equipment.

Table 4-7 : O-RAN Delay Management Model Parameters

	Model Parameters	C-Plane		U-Plane	
		DL	UL	DL	UL
Network	T12min	T12_min	T12_min	T12_min	NA
	T12max	T12_max	T12_max	T12_max	NA
	T34min	NA	NA	NA	T34_min
	T34max	NA	NA	NA	T34_max
O-RU	T2amin	T2a_min_cp_dl	T2a_min_cp_ul	T2a_min_up	NA
	T2amax	T2a_max_cp_dl	T2a_max_cp_ul	T2a_max_up	NA
		Tcp_adv_dl	NA	NA	NA
	Ta3min	NA	NA	NA	Ta3_min
	Ta3max	NA	NA	NA	Ta3_max
O-DU	T1amin	T1a_min_cp_dl	T1a_min_cp_ul	T1a_min_up	NA
	T1amax	T1a_max_cp_dl	T1a_max_cp_ul	T1a_max_up	NA
	Ta4min	NA	NA	NA	Ta4_min
	Ta4max	NA	NA	NA	Ta4_max

The relationships between the various delay model parameters are defined below:

Table 4-8 : Downlink Delay Relationships

	Earliest transmission from O-DU	Latest Transmission from O-DU
U-Plane	T1a_max_up ≤ T2a_max_up + T12_min	T1a_min_up ≥ T2a_min_up + T12_max
C-Plane	T1a_max_cp_dl ≤ T2a_max_cp_dl + T12_min	T1a_min_cp_dl ≥ T2a_min_cp_dl + T12_max

Table 4-9 : Uplink Delay Relationships

	Earliest Reception at O-DU	Latest Reception at O-DU
U-Plane	Ta4min ≤ Ta3min + T34min	Ta4max ≥ Ta3max + T34max
C-Plane	T1a_min_cp_ul ≥ T2a_min_cp_ul + T12_max	T1a_max_cp_ul ≤ T2a_max_cp_ul + T12_min

For the O-RAN interface to operate properly, the transmit and receive windows at the O-DU shall be properly aligned. (The O-RU window alignment is always based on Ra.) For the O-DU to align transmit and receive windows, it needs the following:

- **O-RU Delay Characteristics**
- **Transport Network Delay Characteristics**

The expected accuracy of the reported O-RU delay characteristics is 200ns. The 200ns value was decided as a compromise between a very high accuracy (tens of nanoseconds) which makes buffering in the O-RU easier and a more modest accuracy (~0.5 microseconds) which is relatively easy for a well-controlled Ethernet network to accomplish. This accuracy applies only to the reported start and end times of the reception and transmission windows relative to Ra. Ra is a fixed reference point as defined by the respective air interface standards and supported features. As an example, if the O-RU reports T2a_min_up of 20 µs, the O-RU may begin processing the data from 19.8 µs to 20.2 µs prior to the symbol over the air transmission time (Ra). However, the O-RU MUST still transmit over the air with the precision as defined by the air interface standards.

The remainder of this clause addresses O-RAN supported methods for providing the O-DU with the required information to properly align transmit and receive windows. The following methods are supported:

Table 4-10 : O-DU transmit/ receive window alignment methods

Delay Characteristics	Computed Methods	
	Defined Transport	Measured Transport
O-RU Delay	O-RU specific values available to O-DU	O-RU specific values available to O-DU
Transport Delay	Pre-defined Min/ Max	Min – Measured; PDVmax - SLA

4.3.3 Computed Latency Methods

As previously noted, the goal of delay management for O-RAN interfaces is to ensure that the transmission/ reception windows as the O-DU are properly aligned to support the O-RU and transport network characteristics. The computed methods require the O-DU to compute the required transmit and receive windows based on O-RU delay and transport network characteristics. The O-DU then determines if the O-DU can accommodate the required transmit and receive windows. If so, it sets its transmit and receive windows (based on O-DU design) to the most appropriate settings. As an example, an O-DU may have a fixed set of transmit and receive windows. The O-DU may then select the fixed settings that best fit the transport latency (T12/ T34) and O-RU delay characteristics for the connected O-RUs.

The O-DU should raise some notification if it is not capable of accommodating the required transmit and receive windows.

Alignment of the transmit window at the O-DU requires that the following conditions be met based on the downlink delay relationships defined in **Table 4-8**:

- The O-DU MUST be able to transmit DL U-Plane **(T1a_min_up/ T1a_max_up)**
  - Early enough **(e.g. before T1a_min_up)** to ensure it is received at O-RU before T2a_min_up
  - Late enough **(e.g. after T1a_max_up)** to ensure it is NOT received before T2a_max_up
- Similarly, the O-DU shall be able to transmit C-Plane messages within their respective windows

Alignment of the receive window at the O-DU requires that the following conditions be met based on the uplink delay relationships defined in **Table 4-9** :

- The O-DU MUST set its receive window for UL U-Plane **(Ta4_min, Ta4_max)**
  - Early enough to ensure UL U-Plane is not received before Ta4_min
  - Late enough to ensure UL U-Plane is not received after Ta4_max

The following O-DU constraints shall be met for delay management to operate properly:

Table 4-11 : O-DU Constraints

O-DU	Constraint		
	Description	**Parameter**	**Relationship**
Transmit Window (U-Plane)	maximum required transmit window	TXmax$_{\text{O-DU}}$	T1a_max_up – T1a_min_up ≥ TXmax$_{\text{O-DU}}$
	Maximum supported T1a_max_up	**T1a_max_up$_{\text{O-DU}}$**	**T1a_max_up$_{\text{O-DU}}$** ≥ T1a_max_up
Receive Window	Maximum supported receive window	**RXmax$_{\text{O-DU}}$**	**RXmax$_{\text{O-DU}}$** ≥ Ta4_max - Ta4_min
	Maximum supported uplink latency relative to Ra	**Ta4_max$_{\text{O-DU}}$**	**Ta4_max$_{\text{O-DU}}$** ≥ Ta4_max
Transmit Window (C-plane)	maximum required transmit window	TXmax$_{\text{O-DU-C-plane}}$	T1a_max_cp_dl – T1a_min_cp_dl ≥ TXmax$_{\text{O-DU-C-plane}}$

If any of the O-DU constraints is violated, then proper delay operation cannot be assured.

4.3.3.1 Fronthaul Timing Domain

When using a computed delay approach, the domain over which the O-DU delay parameters apply shall be considered. A timing domain is defined as the set of O-DU and O-RU ports to which the computed O-DU delay parameters apply. The O-DU timing domain may encompass a single O-DU port, or may encompass multiple O-DU ports, where the O-DU requires the same O-DU timing parameters to apply to all O-DU ports in the domain.

Similarly, each O-DU port in the domain may support O-RAN C/U-Plane traffic to 1 or more O-RU ports. Therefore, a fronthaul timing domain encompasses one or more DU_Port_IDs and one or more RU_port_IDs.

When computing the O-DU transmit/ receive windows, the single set of delay parameters for the O-DU shall encompass the worst case of Transport and O-RU delay characteristics within the domain. The following definitions are used:

- Transport Network Parameters for timing domain

  T12_min$_{ij}$: T12_min between DU_Port_ID(i) and RU_Port_ID(j)

  T34_min$_{ij}$: T34_min between DU_Port_ID(i) and RU_Port_ID(j)

NOTE: it is assumed that if a timing domain has multiple T12_min/ T34_min values, then the transport delay characteristics are measured. For measured transport delay it is not possible to measure the maximum delay. T12_max and T34_max in this case is computed by adding an a pre-defined (e.g. via SLA) worst case variation (PDVmax) to the corresponding transport minimum delay values.

Table 4-12: Radio Delay Parameters for O-RU Port "j" within timing domain

	Downlink U-Plane	Uplink U-Plane	Downlink C-Plane	Uplink C-Plane
Minimum	T2a_min_up$_j$	Ta3_min$_j$	T2a_min_cp_dl$_j$	T2a_min_cp_ul$_j$
Maximum	T2a_max_up$_j$	Ta3_max$_j$	T2a_max_cp_dl$_j$	T2a_max_cp_ul$_j$
NOTE: Tcp_adv_dl may be different across RUs within a domain, therefore T2a_min_cp_dl and T2a_max_cp_dl shall be used to determine downlink C-Plane window.				

The resulting parameters to be used for determining the O-DU window for the timing domain are:

Table 4-13 : Transport Network Parameters for Timing Domain

	Downlink	Uplink
Minimum	T12_min = MIN(T12_min$_{ij}$)	T34_min = MIN(T34_min$_{ij}$)
Maximum	T12_max = MAX(T12_min$_{ij}$) + PDVmax	T34_max = MAX(T34_min$_{ij}$) + PDVmax

Table 4-14 : O-RU Delay Parameters for Timing Domain

		U-Plane	C-Plane
Downlink	Minimum	T2a_min_up = MAX(T2a_min_up$_j$)	T2a_min_cp_dl = MAX(T2a_min_cp_dl$_j$)
	Maximum	T2a_max_up = MIN(T2a_max_up$_j$)	T2a_max_cp_dl = MIN(T2a_max_cp_dl$_j$)
Uplink	Minimum	Ta3_min = MIN(Ta3_min$_j$)	T2a_min_cp_ul = MAX(T2a_min_cp_ul$_j$)
	Maximum	Ta3_max = MAX(Ta3_max$_j$)	T2a_max_cp_ul = MIN(T2a_max_cp_ul$_j$)

The tables defined above are used to derive the corresponding delay parameters to be used in the timing relationships defined in **Table 4-8** and **Table 4-9** to support the computed methods defined below.

When using a computed delay approach, there are two situations that are considered within O-RAN:

- **Defined Transport Method**: the network delay is pre-defined (usually by the network operator) and the delays are computed based on that definition. The definition generally considers the maximum network latency, with an assumption that a smaller delay can more easily be accommodated.
- **Measured Transport method**: the network delay is estimated based on actual measurements of packet latency on the downlink and uplink (sometimes a symmetric delay is assumed but this is not necessary).

Regardless of whether the Defined Transport Method or Measured Transport Method is used, the calculations are mainly the same. The next two clauses describe in more detail these two methods.

4.3.3.2 Defined Transport Method

With the Defined Transport method, the O-DU transmit and receive windows are determined based on pre-defined transport network characteristics, and the delay characteristics of the O-RUs within the timing domain. For this approach to work, the O-RU delay characteristics as defined in **Table 4-14** for each O-RU in the timing domain shall be available to the O-DU. The O-DU can adapt its transmit and receive windows to accommodate the O-RU delay characteristics if O-DU has that capability. The adaptation of the O-DU transmission and reception windows may be limited based on the O-DU design.

Table 4-15 : DL and UL Delay Profiles

	O-RU delay profile	O-DU delay profile
Downlink	T2a_min_up, T2a_max_up, T2a_min_cp_dl, T2a_max_cp_dl, Tcp_adv_dl	T1a_max_up$_{O\text{-}DU}$, TXmax$_{O\text{-}DU}$
Uplink	Ta3_min, Ta3_max, T2a_min_cp_ul, T2a_max_cp_ul	Ta4_max$_{O\text{-}DU}$, RXmax$_{O\text{-}DU}$

Refer to M-Plane specification for details on how O-RU parameters are obtained. Optionally, an O-RU may adapt its delay profile information, especially for uplink, based on O-DU delay profile and transport delay (T12_min/ T34_min). If this option is supported, then the O-DU shall provide its delay profile as well as the T12_min to the O-RU. The exchange of this information should use the same approach as defined for exchange of O-RU delay profile. With this option, the O-RU receives the O-DU delay profile before sending its delay profile to the O-DU. The O-DU is unaware that the O-RU delay profile information has been modified. It simply uses the delay profile provided by the O-RU as it would without this option.

As previously noted, delay characteristics for an O-RU may vary based on air interface properties. To ensure interoperability, O-RAN supported air interface properties which may be used as the basis for supporting different delay characteristics are limited to:

- Channel Bandwidth
- SCS

A set of delay characteristics which applies to a combination of the above properties is referred to as a delay profile. For each supported combination of the above properties that an O-RU supports, a delay profile shall be identified. Note it is possible for multiple combinations of the above properties to utilize the same O-RU delay profile. These delay profiles are O-RU specific, and are not within the scope of the O-RAN specification.

When calculating the O-DU transmit and receive window for a timing domain, the O-DU shall use the delay profile applicable for each O-RU based on the air interface properties used by the O-RU in the specific network configuration. As previously noted, it is possible that the O-DU design is not capable of adjusting its transmit/ delay windows to meet the required transmit or receive windows.

The O-RU and O-DU may have multiple delay profiles depending on the design as shown in **Table 4-14**.Using the delay parameters specified for the corresponding profile for each O-RU in the timing domain, the uplink and downlink O-RU parameters to use for the timing domain can be determined as specified in clause 4.3.3.1. **Table 4-12**. The transport delay parameters are constant for this method and specified as part of the network. The delay relationships as previously defined in **Table 4-8** and **Table 4-9** can then be applied to determine the O-DU transmit and receive delay parameters. Finally, the O-DU uplink and downlink constraints as defined in **Table 4-11** are applied to ensure that the calculated windows can be supported.

4.3.3.3 Measured Transport Method

With the Measured Transport method, the O-DU transmit and receive windows are determined based on the delay characteristics of the O-RUs as defined in clause 4.3.3.2, and measured transport delays between all O-DU ports and O-RU ports in the timing domain. The O-DU shall measure the transport delay(s) for all O-RU/O-DU port pairs in the timing domain and adapt its transmit and receive windows to accommodate the measured transport delay characteristics.

O-RAN specifies the use of the One-Way Delay Measurement messages as defined in the eCPRI specification [2]. Some of the benefits of using the eCPRI approach are:

- Measurement consistency between the C-Plane and U-Plane is achieved by using a single processing element (see clause 3.1 for the definition of a processing element) and associated C/U-Plane endpoint address for the measurement.
- eCPRI One-Way Delay Measurement allows for varying packet sizes to better simulate real traffic.
- eCPRI approach allows for separate T12 and T34 measurements.

Note that for this delay measurement eCPRI One-Way Delay protocol shall be used because no other specified transport mechanism supports one-way delay measurements – this is true even if RoE is used as the C-Plane and U-Plane transport mechanism.

Table 4-16 : eCPRI One-Way Delay Measurement Message

One-Way Delay Measurement (Type 5)									
0 (msb)	1	2	3	4	5	6	7 (lsb)	# of bytes	
ecpriVersion				ecpriReserved			ecpriConcatenation	1	Octet 1
ecpriMessage = 5								1	Octet 2
ecprPayload								2	Octet 3
Measurement ID								1	Octet 5
Action Type								1	Octet 6
TimeStamp (seconds)								6	Octet 7
TimeStamp (nanoseconds)								4	Octet 13
Compensation value (nanoseconds)								8	Octet 17
Dummy bytes								L	Octet 25
									Octet M

The O-DU is the initiator for measuring T12 and optionally T34 according to **Figure 4-17** (T12) and **Figure 4-18**(T34).

eCPRI Specification [3] defines 2 methods for measuring the One-Way-Delay, a 1-Step and a 2-Step method. With the 1-Step method the sender of the Request message will include the values of t1 and tCV1 in the request. With the 2-Step method these values are included in a Follow-Up message. See [3] for detailed information of this procedure.

Usage of either 1-Step or 2-Step method for the T34 measurement is controlled by M-Plane O-RU capability parameters **one-step-t34-supported** and **two-step-t34-supported**. When the O-RU supports Measured Transport Method (eCPRI Msg 5) the O-RU needs to support at least one of the methods for measurement of T34. When the two parameters are reported by the O-RU in the capability report, the O-RU shall support both 1-Step and 2-Step method of the T12 measurement. If the O-DU will initiate a T34 measurement, the O-DU shall support both 1-Step and 2-Step method of the T34 measurement. When the two parameters are not reported by the O-RU in the capability report, but measured transport method is supported by the O-RU, the O-DU may probe the O-RU capabilities by using either 1-step or 2-step method when performing a T12 or T34 measurement.

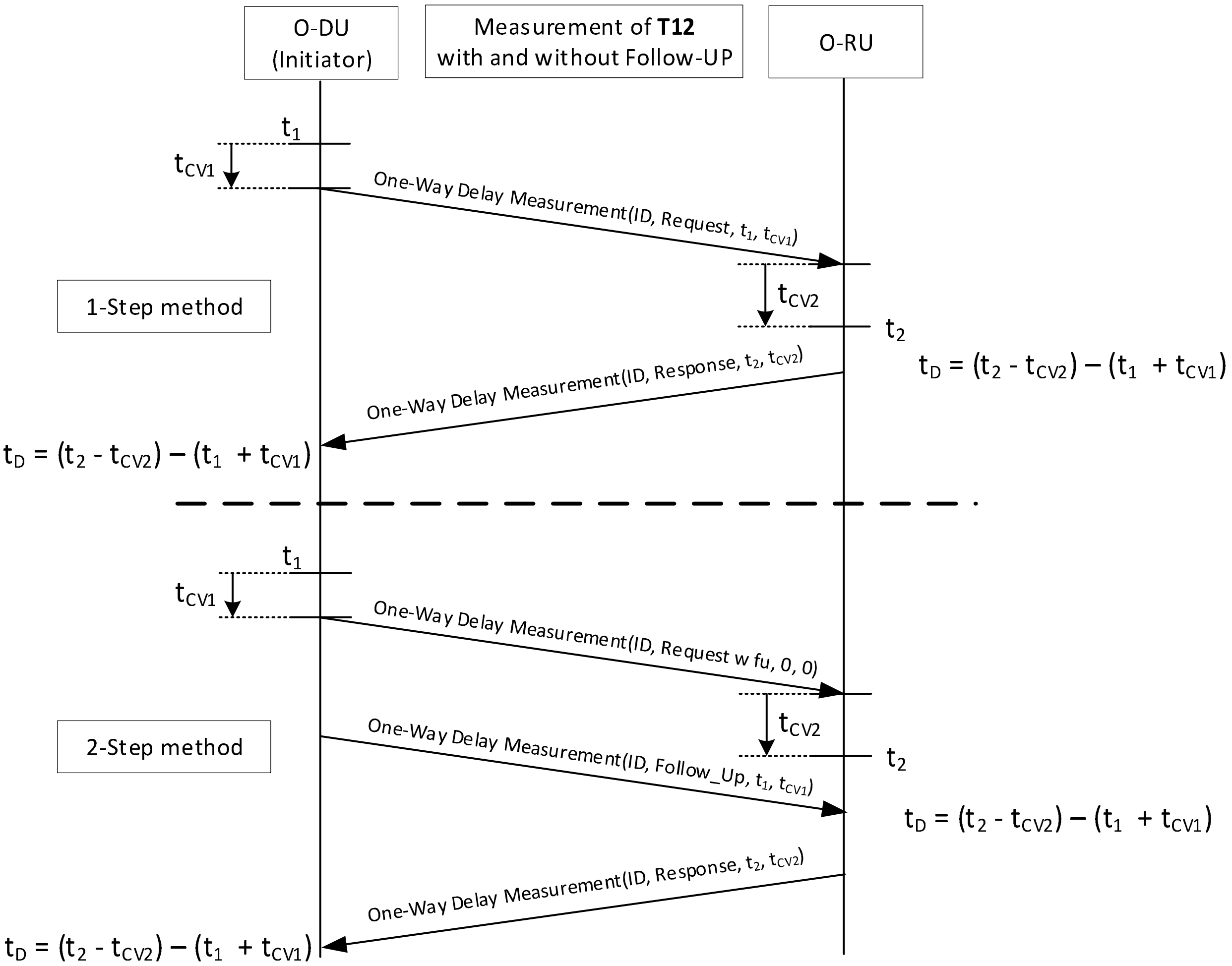

O-DU (Initiator)
Measurement of T12 with and without Follow-UP
O-RU
t1
tCV1
One-Way Delay Measurement(ID, Request, t1, tCV1)
tCV2
t2
1-Step method
tD = (t2 - tCV2) – (t1 + tCV1)
One-Way Delay Measurement(ID, Response, t2, tCV2)
tD = (t2 - tCV2) – (t1 + tCV1)
t1
tCV1
One-Way Delay Measurement(ID, Request w fu, 0, 0)
tCV2
2-Step method
One-Way Delay Measurement(ID, Follow_Up, t1, tCV1)
t2
tD = (t2 - tCV2) – (t1 + tCV1)
One-Way Delay Measurement(ID, Response, t2, tCV2)
tD = (t2 - tCV2) – (t1 + tCV1)

Figure 4-17 : eCPRI One-Way Delay T12 Measurement procedure

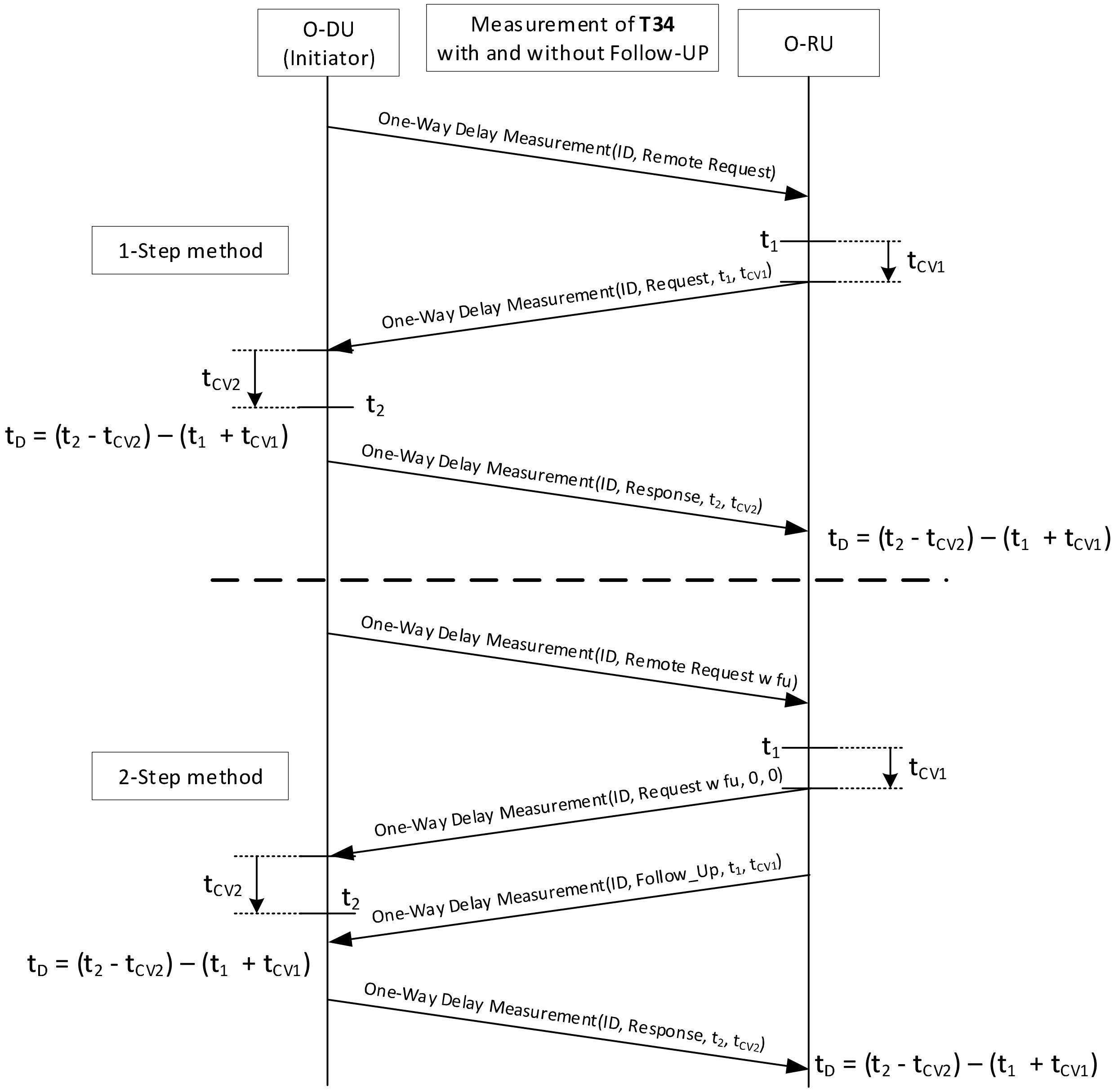

O-DU (Initiator)
Measurement of T34 with and without Follow-UP
O-RU
One-Way Delay Measurement(ID, Remote Request)
1-Step method
t1
tCV1
One-Way Delay Measurement(ID, Request, t1, tCV1)
tCV2
t2
tD = (t2 - tCV2) – (t1 + tCV1)
One-Way Delay Measurement(ID, Response, t2, tCV2)
tD = (t2 - tCV2) – (t1 + tCV1)
One-Way Delay Measurement(ID, Remote Request w fu)
2-Step method
t1
tCV1
One-Way Delay Measurement(ID, Request w fu, 0, 0)
tCV2
t2
One-Way Delay Measurement(ID, Follow_Up, t1, tCV1)
tD = (t2 - tCV2) – (t1 + tCV1)
One-Way Delay Measurement(ID, Response, t2, tCV2)
tD = (t2 - tCV2) – (t1 + tCV1)

Figure 4-18: eCPRI One-Way Delay T32 Measurement procedure

Note that when measuring the one-way delay, the goal is to determine the minimum transport delay. Since any individual packet can experience different delays through the network due to PDV, it is necessary to perform multiple measurements. Since the desired value is the minimum (T12_min or T34_min), the minimum delay measurement among the various measurements is used to estimate the minimum transport delay. The O-DU may use the estimated T12_min value as the T34_min value.

There is no theoretical upper bound for transport delay. The maximum values are determined based on the measured minimum values plus a maximum allowed transport variation (PDVmax) defined as part of the SLA for the transport network.

The O-DU shall measure the delay corresponding to each O-DU/ O-RU port pair within the timing domain. The per port pair minimum delay are then used as per **Table 4-12** to provide the transport parameters to be used in the delay relationships specified in **Table 4-8** and **Table 4-9** . The resulting delay configuration shall meet the O-DU constraints defined in **Table 4-11**.

Since the transport delay is necessary to establish correct timing, this measurement is performed initially before sending of C/U-Plane traffic. The measurement should be performed with packets using the same L2 Cos Priority or L3 DSCP Code as the U-Plane data uses, to ensure the measurement accurately reflects the U-Plane transport latency. This could also be done periodically to verify delay if desired. Note however that clause 4.4 describes traffic counts which can also be used for error detection. If an error is detected, it may be necessary to re-measure the delay and update the O-DU or O-RU timing based on the new measurements.

4.3.4 Latency Categories for O-DU with dynamic timing advance

As an aide to selection of equipment for use in specific transport network use cases, O-RAN O-DU and O-RU are categorized based on delay capabilities. Equipment is categorized with a Category [AAAA-ZZ] and Sub-category [.00-.1000]. The category can be used to determine the maximum T12max/ T34max which the equipment can support. Categories are determined as follows:

- **T1a_max$_{\text{O-DU}}$** ≥ T12max + T2a_min_up
- **Ta4_max$_{\text{O-DU}}$** ≥ T34max + Ta3_max_up

By definition **T1a_max_up$_{\text{O-DU}}$** is the earliest that the O-DU can begin transmission. This results in that the earliest that the O-DU can be guaranteed to complete transmission is **T1a_max_up$_{\text{O-DU}}$ - TXmax$_{\text{O-DU}}$.** The result is that:

- **T1a_max_up$_{\text{O-DU}}$ - TXmax$_{\text{O-DU}}$** - T2a_min_up ≥ T12max

Similarly, for uplink **Ta4_max_up$_{\text{O-DU}}$** is the latest that the O-DU may receive any packet from the O-RU. Therefore:

- **Ta4_max_up$_{\text{O-DU}}$** - Ta3_max_up ≥ T34max

Both of these are illustrated in the **Figure 4-19** below:

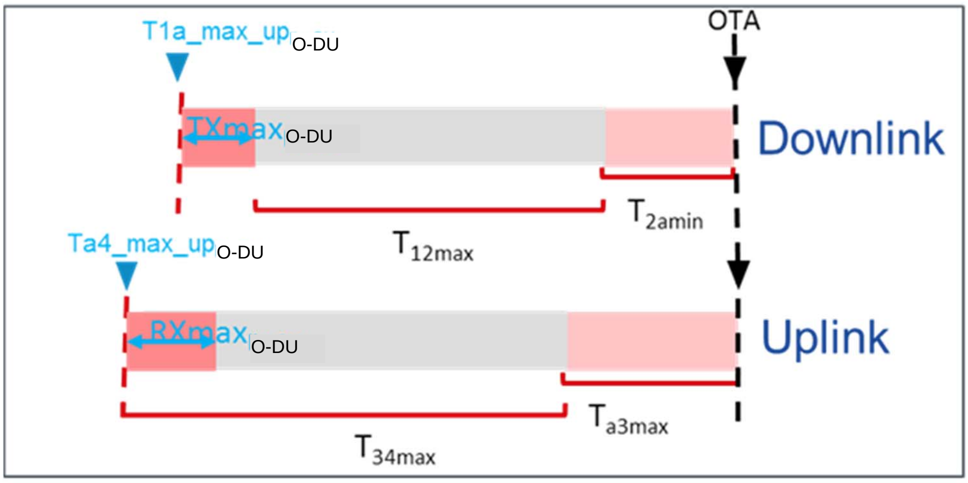


Figure 4-19 : Uplink and Downlink Timing Parameter Relationship

Using these inequalities, O-DU and O-RU can be assigned categories based the following constraints:

Parameters	Constraint			
	Description	O-DU	O-RU Processing	Transport
Downlink	Max Transport	T1a_max_up$_{\text{O-DU}}$-TXmax$_{\text{O-DU}}$	T2a_min_up	**T12_max** = T1a_max_up$_{\text{O-DU}}$ - TXmax$_{\text{O-DU}}$ -T2a_min_up
Uplink	Max Transport	Ta4_max_up$_{\text{O-DU}}$	Ta3_max_up	**T34_max** = Ta4_max_up$_{\text{O-DU}}$ – Ta3_max_up

Note that the maximum range that can be supported is based on the minimum value of T12_max/ T34_max from above. This requires separate categorization for UL and DL for each O-DU and O-RU. Therefore, an uplink category pair (AX) and downlink category pair (BY) shall be evaluated for each pairing of O-DU and O-RU. The minimum distance (T12_max/ T34_max) represented by the corresponding pairs (AX/ BY) then identifies the maximum transport latency which can be supported by the pair.

Each endpoint has a specific value for the characteristics defined in the table immediately above. In order to categorize endpoints, the endpoints are grouped based on the range in which the maximum transport characteristic falls. The category identification then means that the maximum transport characteristic for the endpoint falls somewhere between the maximum and minimum values for the category.

Since the category identifies that the maximum latency supported for the endpoint falls somewhere within the category range, the resulting category pair (AX) defines a range of T12_max/ T34_max for the combination. The combination can be used to identify a maximum T12_max/ T34_max and minimum T12_max/ T34_max that the combination can support. The category pair identifies a best and worst case for the maximum range the combination can support. The actual maximum range supported by a specific O-DU/ O-RU with a given category combination will fall somewhere within the range. That is, for a given O-DU with category A and O-RU with category X, the T12_max/ T34_max supported by the pair is defined as:

- Latency_minAX ≤ (T12_max = T34_max) ≤ Latency_maxAX

Additionally, the paired combination can be mapped to supported use cases. Annex B.2 contains the category definitions, Latency_min, Latency_max, and use case tables.

Note that categories are only useful for identifying the maximum possible range (T12_max/ T34_max) values. To be able to usefully identify equipment for a given use-case, it is also necessary to be able to identify T12_min/ T34_min values. This can be defined in terms of variation, where variation indicates how much lower T12_min/ T34_min can be relative to T12_max/ T34_max respectively. That is:

- DL Variation = T12_max – T12_min
- UL Variation = T34_max – T34_min

An additional level of categorization is required to address the transport delay variation (e.g. T12max – T12min) that a given equipment type pair can support. This is addressed using sub-categories. Note that the total variation that can be supported is a combination of the receive window at the receiver, and the transmit variation. However, the receive window will likely be much larger than the transmit variation. Each endpoint is assigned to both an uplink and downlink category. The O-DU DL sub-category is based on transmit variation, while the O-DU UL sub-category is based on receive window. For UL, the roles are reversed and the O-RU DL sub-category is based on receive window, and the UL sub-category is based on transmit variation.

As a result, sub-categories are defined based on the role for the endpoint in the given transmission direction (UL/ DL).

Downlink variation ($T_{variation\_DL}$) is defined in **Figure 4-20** as follows:

$$T_{variation\_DL} \leq \text{T2a\_max\_up} - \text{T2a\_min\_up} - \text{TXmax}_{O\text{-}DU}$$

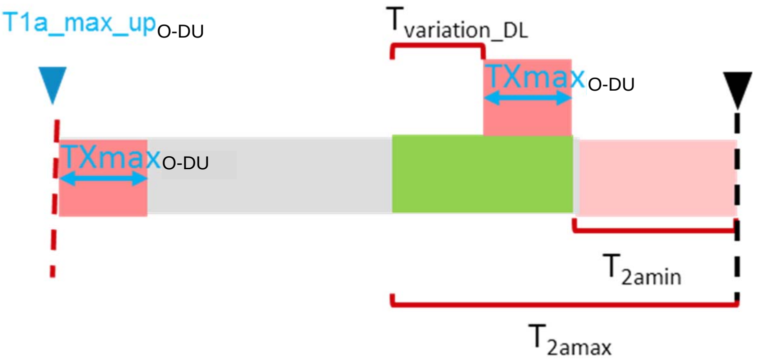


Figure 4-20 : Downlink Transport Variation

Uplink variation ($T_{variation\_UL}$) is defined in **Figure 4-21** as follows:

$$T_{variation\_UL} \leq \text{RXmax}_{O\text{-}DU} - (\text{Ta3\_max} - \text{Ta3\_min})$$

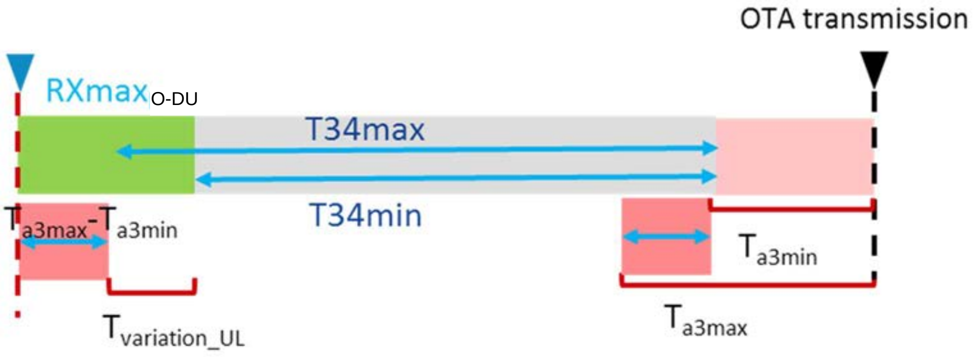


Figure 4-21 : Uplink Transport Variation

Using the respective constraints, the variability can be defined based on O-DU and O-RU delay characteristics as follows:

Parameters	Constraint			
	Description	O-DU	O-RU Processing	Transport
Downlink	Max Variation	$\text{Txmax}_{\text{O-DU}}$	T2a_max_up – T2a_min_up	**T12_max** – **T12_min** = (T2a_max_up – T2a_min_up) – $\text{Txmax}_{\text{O-DU}}$
Uplink	Max Variation	$\text{RXmax}_{\text{O-DU}}$	Ta3_max	**T34_max – T34_min** = $\text{Rxmax}_{\text{O-DU}}$ – (Ta3_max_up – Ta3_min_up)

Sub-categories for each direction are defined using the endpoint characteristics. Similar to categories, sub-categories indicate that the endpoint has a variation characteristic that falls somewhere in the specified sub-category range. Since transmit variability and receive windows are expected to have different ranges, separate tables are created. However, since sub-categories are numeric [.####], to simplify these values are assigned in a similar way. The #### value represents the time in 10's of μsec for the given endpoint property.

For example, if an O-RU with a category X has a T2a_min – T2a_max = 163 μsec, the sub-category for the O-RU would be [.16]. Making the full category for the O-RU X.16 (the "." is not a decimal point but a delimiter only). Note that by using this approach, the receiver sub-category and transmitter sub-category can be directly used to determine the dynamic range in either direction:

TX sub-category: .06

RX sub-category: .16

Resulting $T_{\text{variability}}$ = 16 * 10 - 6 * 10 = 100 μsec. Note that this provides a lower bound on the maximum dynamic range supported in one direction. The minimum of the UL and DL $T_{\text{variability}}$ is used to determine the maximum dynamic range for the combination (e.g. maximum difference between T12_max and T12_min that can be supported).

Annex B.3 contains the sub-category definitions and resulting combination dynamic range use case tables.

With dynamic timing advance, since the O-DU can set its earliest transmission time based on the measured T12_max, the full range of variability is available regardless of the T12_max. For example, if an O-DU/ O-RU combination has 100 usec worth of variability, and a maximum range T12_max of 250 usec, the combination can support the measured T12_max (≤ 250 usec) and T12_min of T12_max – 100 usec. So, if the measured T12_max is 200 usec, the combination can support T12_max of 200 usec and T12_min of 100 usec.

4.3.5 Latency Categories for O-DU with fixed timing advance

The same category concepts apply for O-DU which support fixed transmit/ receive windows. The earliest transmit window and latest receive window timing defines the maximum range that the O-DU can support. $\text{T1a\_max\_up}_{\text{O-DU}}$ for the O-DU is equal to the T1a_max_up of the earliest fixed transmit window. Similarly, $\text{Ta4\_max}_{\text{O-DU}}$ is equal to the Ta4_max of the latest receive window. The O-DU category is then assigned based on this value.

Fixed transmit and receive windows do however impact variability. The fixed transmit time results in sending packets earlier than absolutely needed to achieve the actual T12_max (i.e. the T12_max value that is either measured or configured by the service provider). The following figure **Figure 4-22** illustrates:

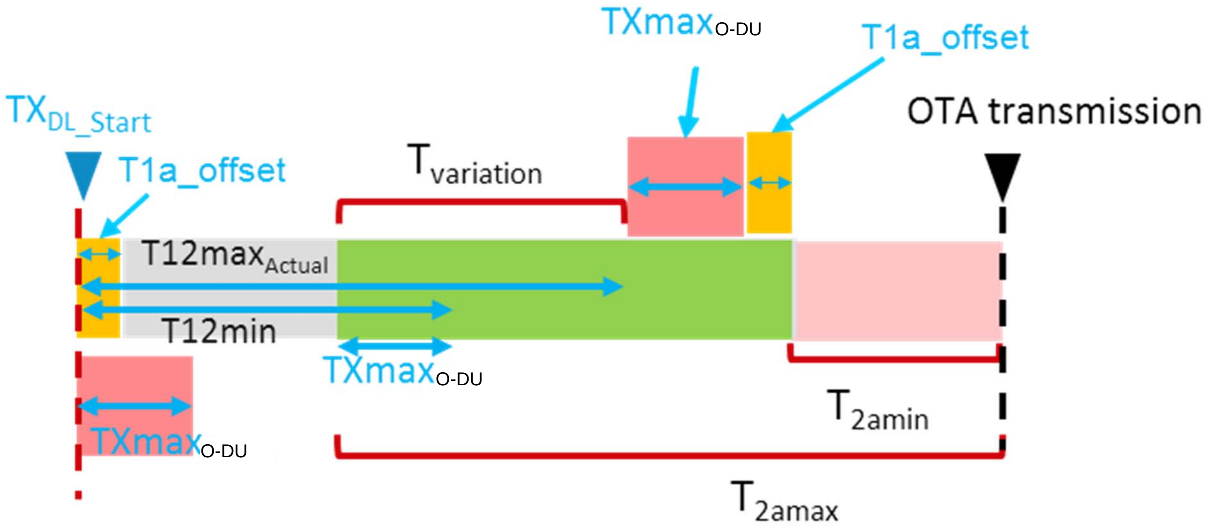


Figure 4-22 : Fixed Transmit Time Illustration

Note that since the TX_{DL_Start} is earlier than required for T12_max by T1a_offset, the latest packets arrive at least T1a_offset before T2a_min_up. This results in T1a_offset worth of O-RU receive buffer being always used, implying that the O-RU needs additional buffer to compensate T1a_offset. Since the TX_{DL_Start} is fixed, the T12min is also fixed at:

$T12min = TX_{DL\_Start} – (T2a\_max\_up – TXmax_{O\text{-}DU})$

Since T12min is unaffected, the resulting $T_{variation}$ is reduced by T1a_offset. To support T12min, O-RU need to provide T1a_offset + T12max – T12min + $TXmax_{\_O\_DU}$

It should also be noted that when T12_max + T2a_min_up = $T1a\_max\_up_{O\text{-}DU}$ (e.g. at maximum range), then T1a_offset = 0. But as the actual T12_max decreases, the amount of receive buffer consumed by T1a_offset increases by the corresponding amount.

Note that the same applies for UL. The result is that the T34_min is defined relative to the T34_max for the combination, not relative to the measured T34_max.

4.3.6 Non-Delay Managed U-Plane Traffic

While most U-Plane data shall meet the delay constraints described above, there are certain types of U-Plane traffic for which the reception windows may not be applicable. Not applying the transmission/ reception window constraints to such data allows the transmission of the data over the fronthaul interface to be spread to reduce the peak bandwidth required by the interface.

Figure 4-23 uses SRS to illustrate an example non-delay managed U-Plane traffic.

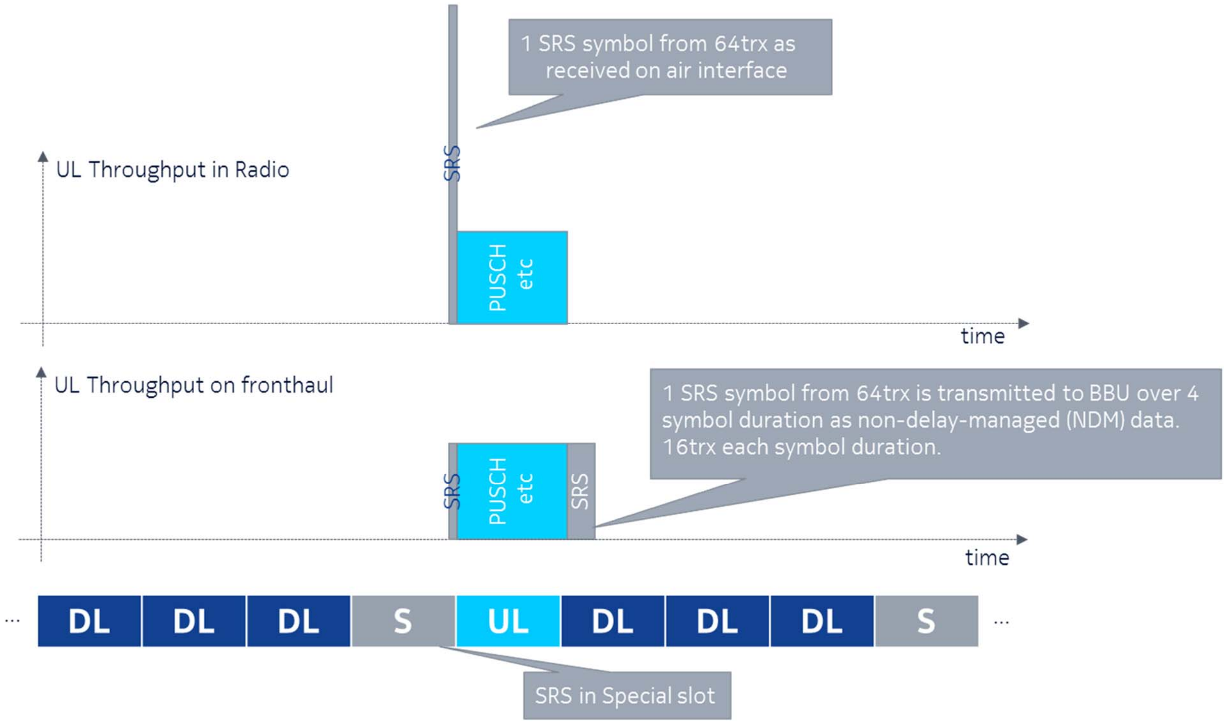

1 SRS symbol from 64trx as received on air interface
UL Throughput in Radio
SRS
PUSCH etc
time
UL Throughput on fronthaul
1 SRS symbol from 64trx is transmitted to BBU over 4 symbol duration as non-delay-managed (NDM) data. 16trx each symbol duration.
SRS
PUSCH etc
SRS
time
...
DL
DL
DL
S
UL
DL
DL
DL
S
...
SRS in Special slot

Figure 4-23 : SRS as non-delay managed U-Plane traffic

In the example, a large amount of SRS data is received during a special slot on the air interface. By allowing the SRS data to be transmitted from the O-RU to the O-DU outside of the normal transmit window, and instead allowing it to use otherwise un-used fronthaul interface bandwidth, the required peak fronthaul bandwidth requirement can be substantially reduced.

Non-delay managed U-Plane traffic flows shall be uniquely identified from delay managed U-Plane traffic flows. The identification of U-Plane traffic flows using eAxC identifiers is described in clause 5.4, and the eAxC shall be used to indicate delay-managed versus non-delay-managed traffic (e.g. some eAxC or Pcid-identified packets may carry delay-managed U-Plane traffic while other eAxC or Pcid-identified packets may carry non-delay-managed U-Plane traffic). Non-delay managed U-Plane traffic flows may not use the same eAxC identifiers as delay managed U-Plane traffic flows. Different RU_Port_IDs are allocated within the eAxC ID for non-delay managed U-Plane flows from those used for delay managed U-Plane traffic flows in the same direction (UL/DL). It is not required that the RU_Port_IDs used in UL for non-delay managed traffic be the same as those used in the DL.

Non-delay managed traffic may be sent by the sender without regard to the transmission window for the corresponding air interface symbol. Similarly, the receiver will not discard received non-delay managed traffic received outside the normal reception window for the corresponding air interface symbol. However, the air interface symbol time is still contained within the U-Plane packets for reference.

Note that the status of traffic flows being delay-managed versus non-delay-managed (which is managed within the application) is independent of any transport prioritization that may be configured.

4.4 Reception Window Monitoring

The reception window can be used for the purpose of error detection. The reception windows at O-RU and O-DU absorb delay variation in the fronthaul. Although its maximum size may vary depending on implementation, the position of windows at O-DU may be adjustable via parametrization as described in clause 4.3.

Data flow error detection during fronthaul operation is monitored and reported via counters based on receipt of packets relative to the reception window. The receiving node monitors the actual reception timing and generates counters regarding the status and whether packets were successfully received in proper timing.

The following reception window performance counters are expected to be supported:

- Data was received on time (within reception window) – no action needed
- Data was received too early – link latency re-measurement may be needed

- Data was received too late – link latency re-measurement may be needed

The above counters and other performance monitoring counters may be implemented as defined in clause 9.1. Based on those counts the system may adjust its operation. The specific reaction will be implementation-specific and depend on such considerations as the number of instances needed before justifying sending an alarm, the ability of the system to measure link latencies, or other design-dependent judgments. Additional details regarding the counts, alarms, and fault handling are outside of the scope of this current document.

Dynamic adaptation of windows at Tx side to handle larger delay values may be employed. For instance, fronthaul distance may be extended by handling a portion (or most) of the fixed delay component at Tx buffer side by adapting the transmission timing based on information provided by Rx side. In other words, rather than accommodating a latency range of zero to “max”, it is possible to instead accommodate a longer latency of “offset” to “max+offset” (longer latency but same range). This is described in detail in clause 4.3.

4.5 Transmission windows

This clause gives a more detailed information on how messages are transmitted within transmission windows.

4.5.1 Normal Transmission

O-RU and O-DU send C- and U-Plane messages at unspecified time within appropriate transmission window (see clause 4.3.2 for general description of transmission windows). This default O-RU transmission behavior in uplink may be altered by configuration over M-plane or with Section Extension 18 in C-plane message as described in the following clauses.

4.5.2 Uniformly Distributed Transmission

O-RU may be configured (over M-plane or with Section Extension 18 in C-plane message) to uniformly distribute transmission of UL U-plane messages over the transmission window (**uniformly-distributed-transmission-supported** in M-Plane).

The definition of “uniformly distributed” is that the total number of packets should as evenly as possible be spread out in time over the transmission window when they are sent on the interface.

The transmission window is divided into N sub-windows where N is the total number of packets to be transmitted during the transmission window. The N packets shall be sent one packet per sub-window. The transmission shall start at a random time within the sub-window.

The above description of when packets are sent can be disturbed by other traffic being transmitted by the O-RU, if that is the case the main object with ‘uniform distribution’ is that the traffic load shall be evenly distributed in time as much as possible.

4.5.2.1 Overlapping transmission windows

In many cases transmission windows will be overlapping between different user data flows (different eAxCs or different section IDs etc.) When this is the case the same “rule” as described above in clause 4.5.2 is still valid. I.e. within the specific sub-window that data packet should be sent at a random time. If several packets (due to different flows) are to be sent within the same sub-window each packet shall be sent at a random time. If the transmission time for the different packets collide then the packets are buffered in any order and sent when link capacity is available.

4.5.3 Ordered Transmission

O-RU may be configured (over M-plane) to order transmission of UL U-plane messages within transmission window (**ordered-transmission-supported** in M-Plane). If O-RU is configured to order transmission of a set of eAxC then O-RU reports how eAxC are grouped into ordering groups (G) and relative order of eAxCs within each group (R) (**transmission-order and transmission-order-group** in M-Plane).

Let

A and B be eAxCs configured to order transmission,
G_A and G_B be ordering groups of eAxC A and B respectively,
R_A and R_B be relative ordering values of eAxC A and B respectively.

M_A and M_B be messages of eAxC A and B respectively that carry data related to the same symbol (the symbol is identified by a the same frame number and the same symbol number within the frame and the same duration as derived from SCS) and transmitted in transmission windows W_A and W_B respectively.

If A and B belong to the same ordering group ($G_A = G_B$) and relative order of A is less then relative order of B ($R_A < R_B$) and start of transmission window W_A is same as start of transmission window W_B and end of transmission window W_A is same as end of transmission window W_B then O-RU transmits message M_A before message M_B. Otherwise O-RU transmits message M_A and M_B in unspecified order. Therefore, order of U-plane messages of eAxCs of different ordering groups and eAxCs of the same relative order (within a group) is unspecified. Also note that messages transmitted in different transmission windows are ordered only as required to transmit them within transmission windows.

4.5.4 Scheduled Transmission

O-RU may be configured (over M-plane or with Section Extension 18 in C-plane message) to shift (delay) and/or resize UL U-plane messages transmission window.

O-RU may be commanded to shift the transmission window by given offset (delay) expressed as number of symbols. Depending on the O-RU capability the eAxC specific window offset (O) and window size (S) may be provided over

1) M-plane (**transmission-window-offset** and **transmission-window-size** in M-Plane) or

2) C-plane (in Section Extension 18).

The window offset (O) and size (S) are specific to each eAxC and depending on O-RU the offset and size may be:

1) specific to all REs in a symbol within eAxC
This is applicable to O and S values provided over M-plane and C-plane.

2) specific to REs selected by section descriptions in a symbol within an eAxC.
This is applicable only to O and S provided in C-plane. In this case, the same values of O and S shall be used for every Section Extension 18 description referring to REs in the same PRB using different values of reMask..

Note that a C-plane message can address multiple eAxCs if Section Extension 7 is used or mapping between of an eAxC_ID to multiple eAxCs is configured with M-plane.

The time at which the O-RU UL U-Plane message is transmitted depends on whether the eAxC is configuration for delay managed or non-delay managed traffic.

If eAxC is configured for delay managed traffic and is also configured (via C-plane or M-plane) to apply transmission window offset (O) and window size (S), then the O-RU transmits the UL U-plane message with data specific to REs in symbol N (N is the number of the symbol within a frame), in a window that starts at the same time as UL U-plane normal transmission window corresponding to symbol N+O and has a duration as configured by window size setting. O-RU reports per endpoint max number of symbols and max number of PRBs that can be buffered (**max-buffered-prbs** and **max-buffered-symbols** in M-Plane). The O-DU shall not request from the O-RU UL a reception and transmission window configuration that exceeds O-RU buffering capacity. For calculation of buffering load, O-DU shall assume that the O-RU allocates its buffer at the time reference for ta3_min (typically start of symbol) and the buffer is released at the end of transmission window.

If eAxC is configured for non-delay managed traffic and is also configured (via C-plane or M-plane) to apply transmission window offset (O) and window size (S), then the O-RU transmits the UL U-plane message with data specific REs in to symbol N (N is the number of the symbol within a frame) in a window that starts after of UL U-plane normal transmission window corresponding to symbol N+O. In this case only the start of transmission window is defined – the end of the transmission window is unspecified. O-RU may report per endpoint max number of symbols and max number of PRBs that can be buffered (**max-buffered-prbs** and **max-buffered-symbols** in M-Plane). O-DU can request from O-RU UL reception and transmission window configuration that exceeds O-RU buffering capacity and O-RU shall handle it on best effort basis. For calculation of buffering load, O-DU shall assume that O-RU allocates buffer at the time reference for ta3_min (typically start of symbol) and the buffer is not released before at the start of transmission window. Note U-plane message transmission of eAxC configured for non-delay managed traffic is always handled by O-RU on best effort basis and transmission is not guaranteed.

If O-RU does not report, over M-plane, support for independent U-plane transmission window control (feature INDEPENDENT-TRANSMISSION-WINDOW-CONTROL in M-Plane), then all endpoints with transmission window control enabled, handling the same carrier type and SCS, shall be configured (via M-plane or C-plane) with parameter values resulting in transmission windows that coincide between the endpoints. That is, for every symbol N, the effective transmission window for symbol N shall start at the same time and end at the same time for all the endpoints handling same carrier type and SCS. This restriction applies only to endpoints that have transmission control enabled over M-plane.

4.6 O-RU External Antenna Delay Handling

Up to and including v06.00 of this specification it is not possible to adjust timing to compensate transmission delays between antenna ports of the O-RU and the external antenna. Figure 2-9 assumes that the antenna delay is negligible compared to the O-RU's internal delay or is known by the O-RU. Figure 2-9 defines the T2a and Ta3 values as the total processing time between Fronthaul interface input/output ports (R2 and R3) and the antenna interface point (Ra).

Figure 4-24 defines the reference points for delay management where the O-RU external antenna delay parameters are defined. Note that T2a and Ta3 will have a new definition when external antenna delays are introduced. T2a is the O-RU internal delay between R2 and Rd and Ta3 is the O-RU internal delay between Ru and R3.

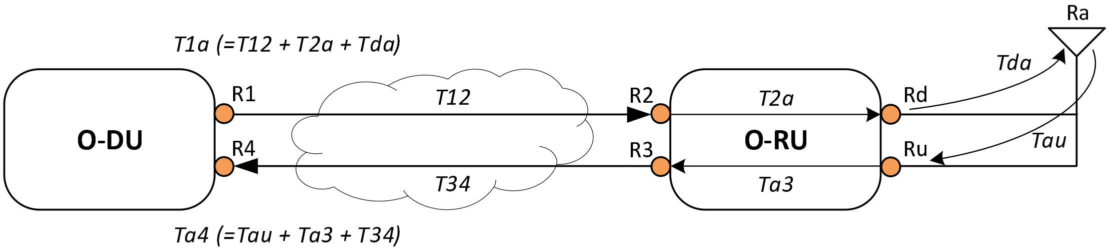


Figure 4-24: Definition of reference points including external antenna delay

Table 4-17: eCPRI O-RU Additional Delay Latency Parameters

	Parameter	Latency	Minimum	Maximum
Downlink	Tda	Timing difference between the output of DL signal at the antenna connector (Rd) of O-RU and the transmission over the air (Ra).	-	-
Uplink	Tau	Timing difference between the reception over the air (Ra) and the input of UL signal at the antenna connector (Ru) of O-RU .	-	-

NOTE :As a first step to introduce the possibility to handle external delays between O-RU and antenna reference point a method is specified in clause 4.6.1 which has minimal impact on the O-DU implementation. FFS: Later releases of this specification may contain other methods including Measured Transport Method.

4.6.1 Minimal O-DU Impact Method (Defined Transport Method)

When using Defined Transport Method as described in clause 4.3.3.2 and non-negligible external delays between the O-RU and antenna are present this method will have a minimal impact on the O-DU implementation.

The presence of the external delays is assumed to be unknown to the part of the O-DU implementation that handles the transmission and reception of messages between the O-DU and O-RU. Since new M-Plane parameters are introduced for Tda and Tau that shall be configured on the O-RU there may be changes to the O-DU's implementation of its O-RU M-Plane configuration flows.

Instead of taking Tda and Tau under consideration when calculating the transmission and reception windows on the O-DU the external delay is taken from the fronthaul delay (T12, T34) timing budget.

When using this method, the values configured to O-RU for **t12-max** and **t34-max** shall maintain the same value as if there is no external delay, i.e. Tau = Tda = 0 (zero). It will be the responsibility of the operator to ensure that the actual fronthaul latency between R1 and R2 respectively R3 and R4 shall not exceed (**t12-max** – Tda) and (**t34-max** – Tau).

This method will keep the transmission and reception windows on the O-DU at the same positions in time with or without the external delays.

The O-RU shall transmit the downlink signal at reference point Rd at (t_{DL}=0 – Tda). The O-RU shall start the processing of the uplink signal received at the reference point Ru at (t_{UL}=0 + Tau).

For downlink C- and U-Plane, the O-RU shall shift its reception windows Tda earlier in time. For uplink C-Plane the reception window shall NOT be shifted. The transmission window for uplink U-Plane data will be shifted by Tau on the O-RU but since the external delay is taken from the fronthaul latency budget the uplink U-Plane messages arrive within the reception window on the O-DU. The sizes for reception and transmission windows remain unchanged on the O-DU and on the O-RU.

4.6.1.1 Minimal O-DU Impact Method - Example

This clause shows an example of the timing relations for downlink and uplink when using the method described in clause 4.1.1.

Table 4-18 contains values for all relevant parameters used in this example.

Table 4-18: Parameters for example

Parameter	Latency (Tda = Tau = 0) [µs]	Latency (Tda = Tau = 25) [µs]
	100	75
	100	
	0	
	100	
	300	
	= T2a_min_up_dl + T12_max = 100 + 100 = 200	
	= T2a_max_up_dl = 300	
	250	
	600	
	= T2a_min_cp_dl + T12_max = 250 + 100 = 350	
	= T2a_max_cp_dl = 600	
	100	
	500	
	= T2a_min_cp_ul + T12_max = 100 + 100 = 200	
	= T2a_max_cp_ul = 500	
	50	
	150	
	= Ta3_min_up = 50	
	= Ta3_max_up + T34_max = 150 + 100 = 250	
µs µs µs µs		

Below are figures similar to figures 2-11 and 2-12, showing the timing relations both with and without the usage of the “Minimal O-DU Impact”-method. **Figure 4-25** and **Figure 4-27** assumes zero external delay both for downlink and uplink, the purpose of these two figures are to show that the relationships stated in Table 2-8 and Table 2-9 are still valid when applying the method in clause 4.6.1., i.e. the transmission and reception windows are unchanged on the O-DU.

The blue lines and arrows of the figures apply to U-Plane messages, and the green lines and arrows apply to C-Plane messages.

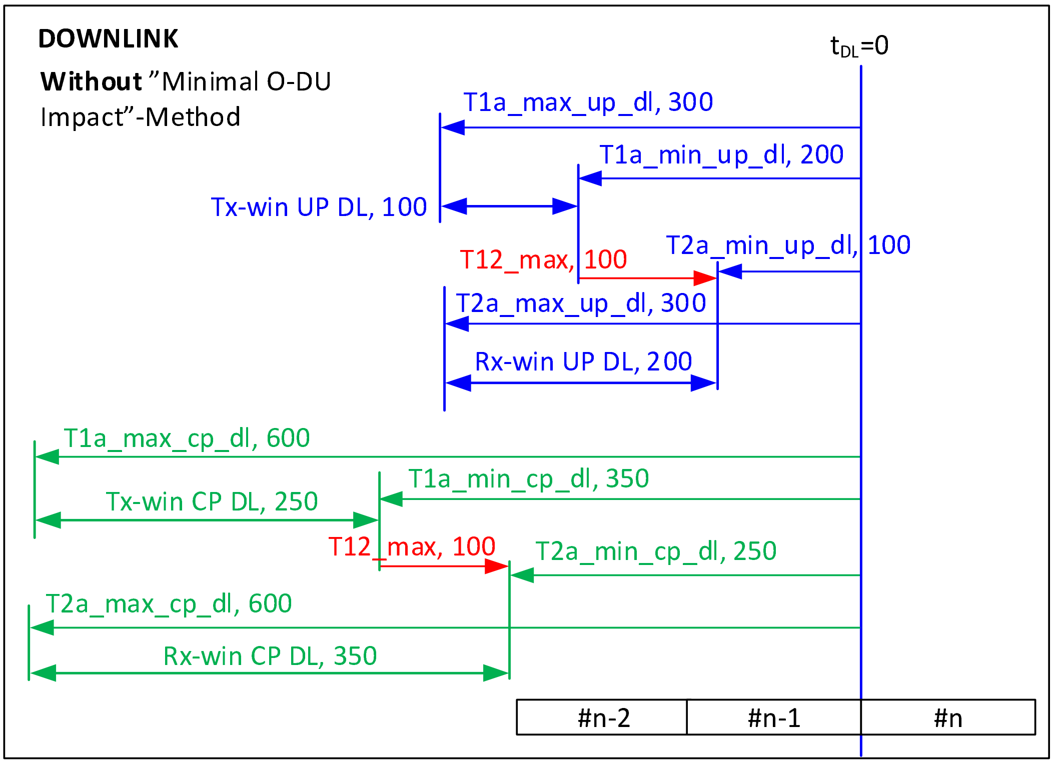


Figure 4-25: Downlink without using 2.6.1

Figure 4-25 shows the timing relations for downlink without any external delays. T12_max in this example is 100 µs, both as configured value and as the actual physical maximum fronthaul latency.

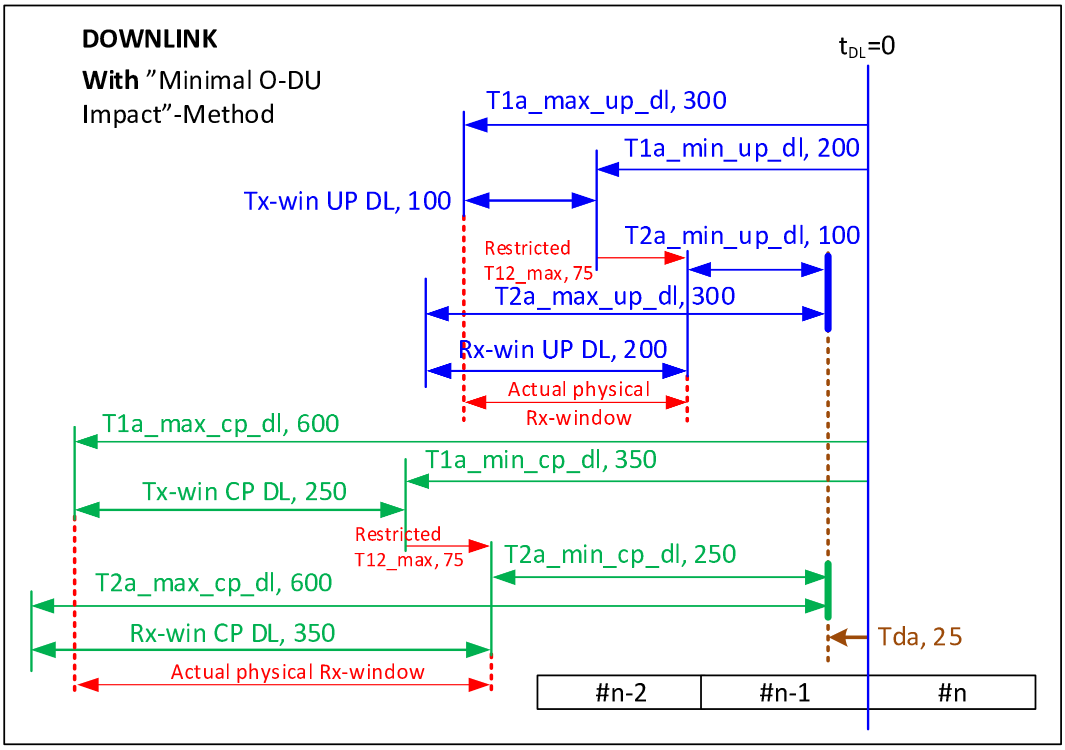


Figure 4-26: Downlink with usage of 2.6.1

Figure 4-26 shows the timing relations for downlink when an external delay of 25 µs is present and the method described in clause 4.6.1 is used. As shown, the transmission windows on the O-DU will be the same as the ones in **Figure 4-25**. The shifting of the reception windows on the O-RU is shown, i.e. these windows will start Tda µs earlier compared to when not using the method in 2.6.1, the sizes of the reception windows will be the same. The red arrows and text show the actual reception windows on the O-RU when restricting the actual physical fronthaul latency from

100 µs to 75 µs but still configuring 100 µs to the system. As can be seen there will be a part at the start of the reception windows that should be "un-used". While it is not expected by the O-RU to receive C-Plane or U-Plane messages during these periods the O-RU shall process these messages.

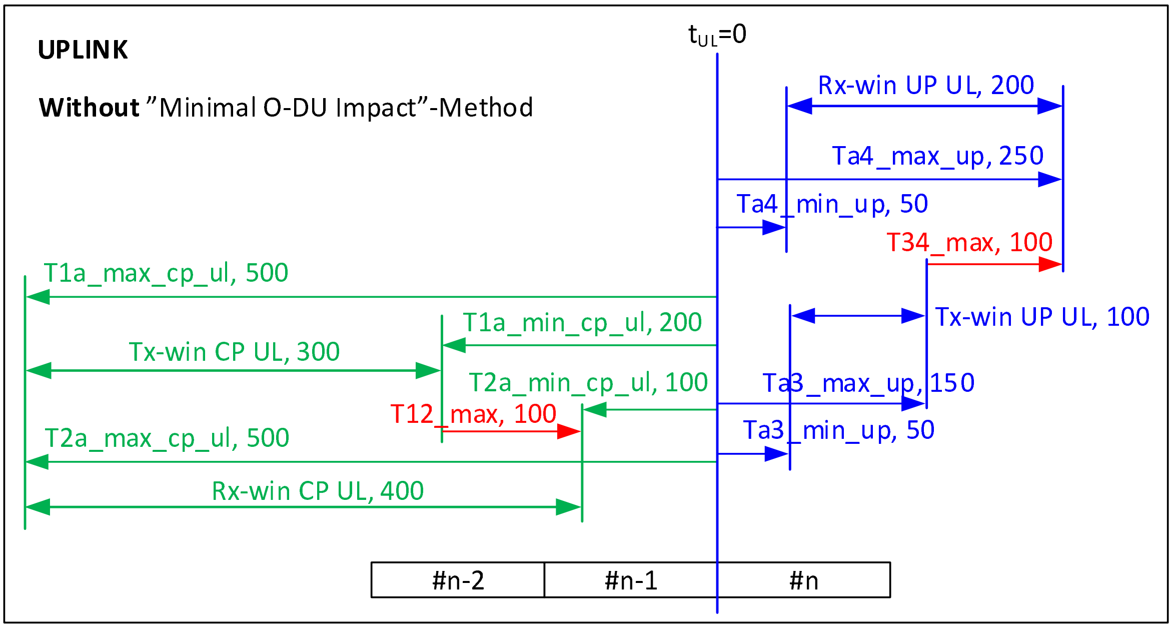


Figure 4-27: Uplink without using 2.6.1

Figure 4-27 shows the timing relations for uplink without any external delays. T12_max and T34_max in this example are 100 µs, both as configured value and as the actual physical maximum fronthaul latency.

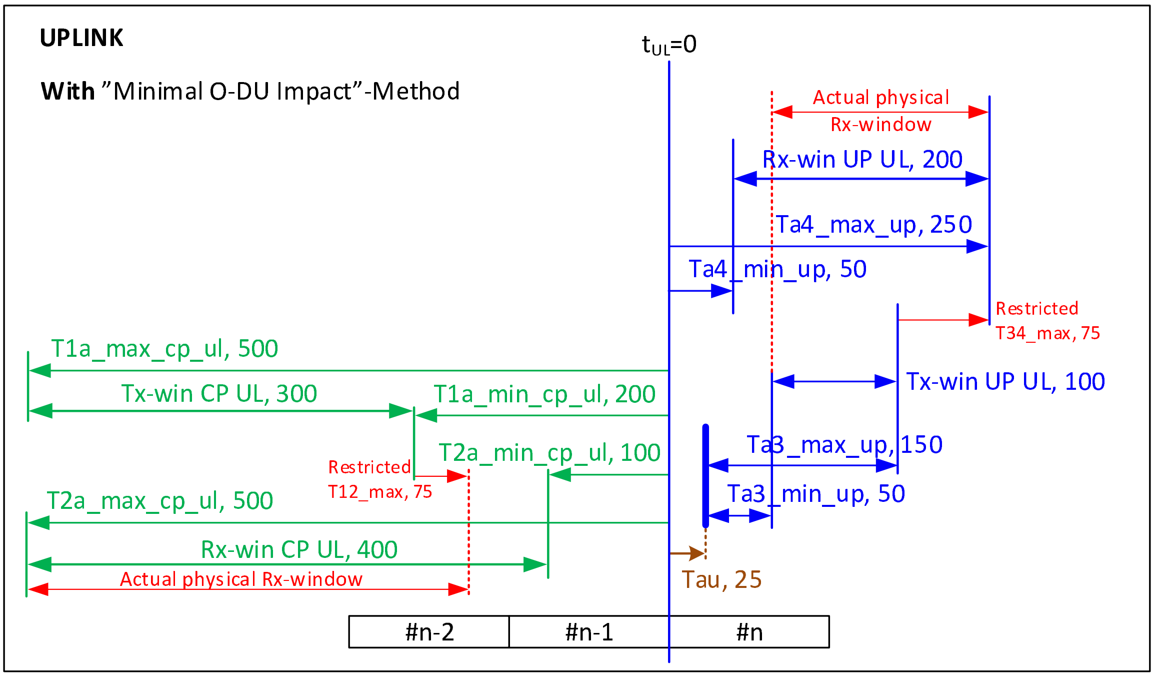


Figure 4-28: Uplink with usage of 2.6.1

Figure 4-28 shows the timing relations for uplink when an external delay of 25 µs is present and the method described in clause 4.6.1 is used. As shown, the transmission window on the O-DU will be the same as the ones in **Figure 4-27**. Note: The reception window on the O-RU for C-Plane messages is NOT shifted for uplink traffic. The size of the reception window will be the same. The red arrows and text show the actual reception windows on the O-RU and on the O-DU when restricting the actual physical fronthaul latency from 100 µs to 75 µs but still configuring 100 µs to the system. As can be seen there will be a part at the end of the reception window on the O-RU that will be "un-used". While it is not expected by the O-RU to receive C-Plane messages during this period the O-RU shall process these messages. On the O-DU there will be a part at the start of the reception window that will be "un-used". While it is not expected by the O-DU to receive U-Plane messages during this period the O-DU shall process these messages. The transmission window on the O-RU for the U-Plane messages will be shifted Tau later in time due to the external latency between reference points Ra and Ru.

5 Transport & Protocol Architecture

5.1 Transport Encapsulation Types

5.1.1 Ethernet Encapsulation

Ethernet can be used as transport mechanism for both U-plane and C-plane. In this case, messages are transmitted over standard Ethernet frames (See Figure 5-1). The supported frame format is Ethernet II/DIX frame with type interpretation of the type length field. The Length-interpretation and multiplexing data with LLC (IEEE 802.2 LLC standard) is not supported. A globally administered unicast MAC address is assigned per Ethernet interface. The Ethernet termination feature supports MAC Client Data field sizes of up to 1500 bytes for basic frames, and up to 9000 bytes for jumbo frames.

To ensure QoS is supported, 3-bit PCP field as defined in IEEE 802.1Q shall be supported meaning the capability for VLAN tagging for the traffic shall be supported and M-Plane configurable (but using VLAN tags may be omitted for certain network configurations e.g. LLS-C1). Priority tagged frames (i.e., frames tagged with VLAN ID=0) are always received, as required by standard. Transmitting priority tagged frames is optional.

The MAC address resolution for U-plane and C-plane is performed via an M-plane procedure specified in [7] regardless of whether the destined node supports IP or not.

The O-DU and O-RU shall support Intermediate L2 switch MAC address learning to avoid flooding U-plane and/or C-plane (not necessary for topology LLS-C1); this is done as a result of test Ethernet frame exchanges during the M-Plane procedure “Ethernet connectivity checking” as specified in [7].

Further, both the eCPRI header and payload are contained within the Ethernet data field. For this encapsulation, either the eCPRI Ethertype or the IEEE 1914.3 Ethertype shall be used.

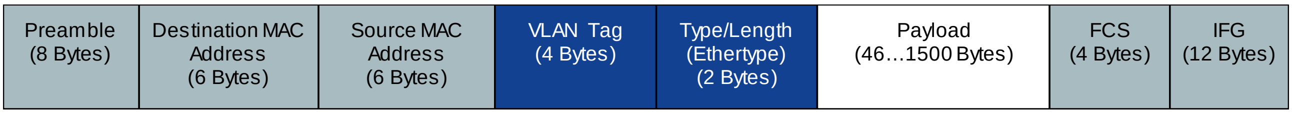

Preamble (8 Bytes)
Destination MAC Address (6 Bytes)
Source MAC Address (6 Bytes)
VLAN Tag (4 Bytes)
Type/Length (Ethertype) (2 Bytes)
Payload (46…1500 Bytes)
FCS (4 Bytes)
IFG (12 Bytes)

Figure 5-1 : Native Ethernet frame with VLAN

5.1.2 IP/UDP Encapsulation

IP/UDP can be used as transport mechanism for both U-plane and C-plane. In this case, IP version 4 shall be supported according to RFC791 (Internet Protocol), and/or IPv6 as per RFC2460 (both IPv4 and IPv6 are optional, see Table 8-2). The encapsulation mechanism is identified by “IPv4” or “IPv6” Ethertype (See Figure 5-2 and Figure 5-3). O-RAN equipment shall not send IPv4 packets with any IPv4 header option field or IPv6 packets with "Next Header" value that is an extension header or "Next Header = 59". If the IPv4 header option field or IPv6 packets with "Next Header" value that is an extension header or "Next Header = 59" is detected in the received packet, the packet is to be silently discarded at the receiving equipment (though the existence of the discarded packet may be logged). For IPv6, if the value in “Next Header” is not an extension header and value is not equal to “59”, it indicates that the next item in the header is upper-layer header and the packets shall be processed by the receiver (Refer RFC2460).

IP based applications can be bound to virtual or physical interface addresses. Each configured physical and logical (i.e. VLAN-) interface shall be configured with a separate subnet. As an option, Alias IP addresses (i.e. more than one IP address on the same interface) can be supported as well. If supported, they can be assigned to logical and physical interfaces as well as to virtual interfaces (loopback interfaces).

ORAN packets shall not use IPv4 or IPv6 fragmentation due to stringent bandwidth and tight latency requirements (such received packets shall be silently discarded). The minimum MTU that is configurable shall be as per the relevant standard, i.e. RFC791 for IPv4 and RFC2460 for IPv6.

The resolution of MAC addresses and the flooding traffic reduction through L2 switch mac address learning shall be done via ARP according to RFC 826 only when IPv4 is supported by the transmitting and receiving nodes.

When using IP/UDP encapsulation, the UDP destination port field shall identify the encapsulated protocol. The UDP destination port field shall be set during initialization via M-Plane configuration. When the eCPRI Specification or IEEE1914.3 define their respective UDP destination ports these will become the default for M-Plane configuration..

On egress, the sending node shall set the checksum for transmitted UDP datagrams as per the relevant standards:

- For IPv4 UDP datagrams the field can be filled with a proper value to mark that the checksum is valid or to 0 to mark that the checksum is not calculated (see RFC768).
- For IPv6 UDP datagrams the field shall be filled with a proper value (see RFC2460).

On ingress, the receiving node may (not shall) validate UDP checksums if the checksum field contains a proper value.

Preamble (8 Bytes)	Destination MAC Address (6 Bytes)	Source MAC Address (6 Bytes)	VLAN Tag (4 Bytes)	Type/Length (Ethertype) (2 Bytes)	IPv4 (20 Bytes)	UDP (8 Bytes)	Payload (1…1472 Bytes)	FCS (4 Bytes)	IFG (12 Bytes)

Figure 5-2 : Native IPv4 packet with VLAN

Preamble (8 Bytes)	Destination MAC Address (6 Bytes)	Source MAC Address (6 Bytes)	VLAN Tag (4 Bytes)	Type/Length (Ethertype) (2 Bytes)	IPv6 (40 Bytes)	UDP (8 Bytes)	Payload (1…1452 Bytes)	FCS (4 Bytes)	IFG (12 Bytes)

Figure 5-3 : Native IPv6 packet with VLAN

Note that while the numbers in **Figure 5-1**, **Figure 5-2** and **Figure 5-3** imply standard Ethernet packets, use of Jumbo frames (up to 9000 bytes in length) is not precluded in this specification.

5.1.3 Transport Headers

O-RAN allows for multiple different transport headers within the Ethernet payload to further describe how the application data is to be handled in the C-Plane and U-Plane. In each case the transport header is 8 bytes in length and provides basic data routing capabilities, including description of the data flow type, sending and reception port identifiers, ability to support concatenation of multiple application messages in a single Ethernet packet, and sequence numbering. The following clauses describe the possible transport headers.

5.1.3.1 eCPRI Transport Header

The definition of the eCPRI transport header is shown in Table 5-1 below.

Table 5-1 : eCPRI Transport Header Field Definitions

<table>
<tr><th colspan="10">Section Type : any</th></tr>
<tr><th>0 (msb)</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7 (lsb)</th><th># of bytes</th><th></th></tr>
<tr><td colspan="4">ecpriVersion</td><td colspan="3">ecpriReserved</td><td>ecpriConcatenation</td><td>1</td><td>Octet 1</td></tr>
<tr><td colspan="8">ecpriMessage</td><td>1</td><td>Octet 2</td></tr>
<tr><td colspan="8">ecpriPayload</td><td>2</td><td>Octet 3</td></tr>
<tr><td colspan="8">ecpriRtcid / ecpriPcid</td><td>2</td><td>Octet 5</td></tr>
<tr><td colspan="8">ecpriSeqid</td><td>2</td><td>Octet 7</td></tr>
</table>

Each field within the eCPRI Transport Header is further described in the following clauses.

5.1.3.1.1 ecpriVersion (eCPRI protocol revision)

Description: This parameter indicates the eCPRI protocol version. NOTE: This parameter is part of the eCPRI common header.

Value range: {0001b=eCPRI version 1.0, 1.1, 1.2 and 2.0, where the interpretation of the eCPRI message shall follow the eCPRI specification versions up to 2.0; 0000b and 0010b-1111b=Reserved for future eCPRI protocol revisions }.

Type: unsigned integer.

Field length: 4 bits

Default Value: 0001b (eCPRI version 1.0, 1.1, 1.2 and 2.0).

5.1.3.1.2 ecpriReserved (eCPRI reserved)

Description: This parameter is reserved for eCPRI future use. NOTE: This parameter is part of the eCPRI common header.

Value range: {001b-111b=Reserved}.

Type: unsigned integer

Field length: 3 bits.

Default Value: 000b (reserved fields should always be set to all zeros).

5.1.3.1.3 ecpriConcatenation (eCPRI concatenation indicator)

Description: This parameter indicates when eCPRI concatenation is in use (allowing multiple eCPRI messages in a single Ethernet payload). NOTE: This parameter is part of the eCPRI common header.

Value range: {0b=No concatenation, 1b=Concatenation}.

Type: binary bit.

Field length: 1 bits.

Default Value: 0b (no concatenation).

5.1.3.1.4 ecpriMessage (eCPRI message type)

Description: This parameter indicates the type of service conveyed by the message type. NOTE: This parameter is part of the eCPRI common header. NOTE: In this version of the specification, only values “0000 0000b” and “0000 0010b” and “0000 0101b” are used.

Value range:
0000 0000b = IQ data message;
0000 0010b = Real-time control data message;
0000 0101b = transport network delay measurement message (see clause 4.3.3.3 for full message format);
other values not recognized within this version of the specification.

Type: unsigned integer.

Field length: 8 bits.

Valid Values: 0x0 (U-Plane data) or 0x2 (C-Plane data) or 0x5 (network delay measurement messages).

5.1.3.1.5 ecpriPayload (eCPRI payload size)

Description: This parameter is the size in bytes of the payload part of the corresponding eCPRI message. It does not include any padding bytes following the eCPRI message. The maximum supported payload size is 2^{16}-1, but the actual size may be further limited by the maximum payload size of the underlying transport network. NOTE: This parameter is part of the eCPRI common header.

Value range: {0000 0000 0000 0000b-1111 1111 1111 1111b}.

Type: unsigned integer.

Field length: 16 bits.

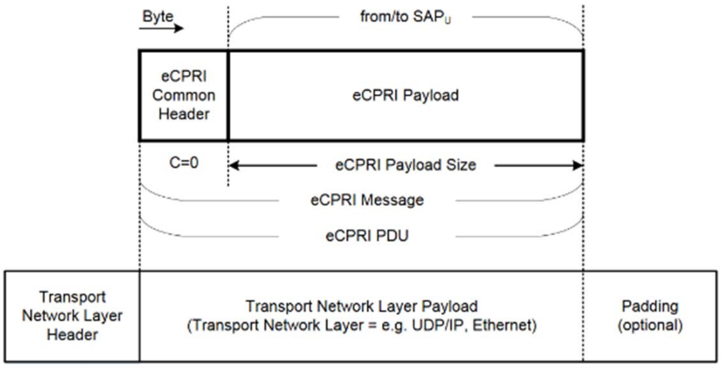


Figure 5-4 : eCPRI payload - no concatenation (source: [2])

5.1.3.1.6 ecpriRtcid / ecpriPcid (real time control data / IQ data transfer message series identifier)

Description: This parameter is an eAxC identifier (eAxC ID) and identifies the specific data flow associated with each C-Plane (ecpriRtcid) or U-Plane (ecpriPcid) message. It is the analog of CPRI's "AxC" (antenna-carrier) value so is designated here as "eAxC" ("e" for "extended" to accommodate multiple bands and multiple component carriers). Multiple O-DU processors may contribute to a single eAxC.

An endpoint may support mixed numerologies by indicating the capability via an M-plane parameter; and a Section Type 3 message can be used to select one of the numerologies (frameStructure) from the capability list of possible numerologies for that endpoint. In this case, a single eAxC id may be used for channels with different characteristics (framestructure, cpLength, timeOffset, freqOffset), e.g. channels with different numerology and PRACH can share same eAxC Id. Alternatively, an endpoint may be simpler, supporting a single numerology by declaring non-support of mixed numerology capability via the M-plane parameter, and a Section Type 3 message can be used to convey different values for parameters other than frameStructure than the M-plane configured value. In this case a unique eAxC id would be used for each mixed numerology channel (frameStructure), i.e. channels with different numerologies will use different eAxC ids. An endpoint may declare non-support for Section Type 3, in which case it is suitable for channels like PDSCH/PUSCH.

The eAxC ID value assigned to an O-RU's endpoint is used to identify the endpoint within O-RU and shall be unique among all endpoints of O-RU of the same direction (Tx or Rx). An O-RU's endpoint could in principle be associated with multiple physical and virtual fronthaul interfaces (Ethernet ports and VLANs), but an eAxC ID value shall be unique and cannot be used to address different endpoints. The association of an O-RU's endpoint with multiple fronthaul interfaces is subject to O-RU design restrictions reported by O-RU via the M-Plane – interfaces not associated with a given endpoint shall not be configured via M-plane for transferring C- and U-plane message to the endpoint. Also, fronthaul interfaces not configured via M-plane shall not be used for transferring C- and U-plane messages to the endpoint. In other words, the O-DU shall obey the O-RU restrictions as conveyed on the M-Plane.

The O-DU may decide to use same eAxC ID value to address endpoints on different O-RUs, but eAxC ID values assigned to one O-RU shall be unique within the O-RU's endpoints of same direction.

It is noted that there are O-RUs which do not support, within a same eAxC ID, independent sequence checkers for C-plane messages describing U-plane DL and C-plane messages describing U-plane UL, which is specified in clause 5.1.3.1.7. This O-RU capability limitation can be signalled by O-RUs and be interpreted by O-DUs from M-plane specification v03.00. If either or both of the O-RU and O-DU only supports previous versions of the M-plane specification, then the network operators and vendors intending to work with such O-RUs will need to ensure that the O-DU can interpret the O-RU capability limitation in other non-standardized ways, i.e. through off-line discussions. To interoperate with such O-RUs, O-DUs will need to avoid requiring at the O-RU, within a same eAxC ID, independent sequence checkers for C-plane messages describing U-plane DL and C-plane messages describing U-plane UL, either by (1) using different eAxC IDs for them; or (2) if the O-DU chooses to use the same eAxC ID, using a shared sequence generator for C-plane messages describing U-plane DL and C-plane messages describing U-plane UL as shown in **Figure 3-6** and described in clause 5.1.3.1.7. However, support of such behavior in order to interoperate with such O-RUs is optional for the O-DUs, and it is noted that incompatible operation is likely to result in data outages and spurious error reports.

This version of specification does not define behavior regarding transferring C- and U-plane messages to or from an endpoint over multiple fronthaul interfaces at the same time. For example, in UL, if an endpoint were to be assigned to multiple fronthaul ports, there is no way to instruct the endpoint which fronthaul port to use for any given message. In addition, in DL there is no accommodation for multiple delay windows per eAxC ID when multiple links are used. Therefore, an endpoint shall be assigned exactly one fronthaul interface via the M-Plane, but different endpoints may use different fronthaul interfaces (ports) so the O-RU as a whole may use multiple fronthaul ports.

eAxC ID subfields

One eAxC identifier (eAxC ID) comprises a band and sector identifier (BandSector_ID), a component-carrier identifier (CC_ID) a spatial stream identifier (RU_Port_ID) and a Distributed Unit identifier (DU_Port_ID).

NOTE: This parameter is specific to eCPRI Message Type = 2 (C-Plane) and Message Type 0 (U-Plane) messages. In this version of the specification, one eAxC contains only one spatial stream (i.e. one beam per subcarrier) at a time. When precoding in the O-RU, then each eAxC contains one layer at a time, except for TxD (LTE TM2) when a single eAxC (single ecpriRtcid and ecpriPcid) represents all TxD layers.

Bit allocation is subdivided as follows:

- DU_Port_ID: Used to differentiate processing units at O-DU (e.g., different baseband cards). It is expected the O-DU will assign these bits, and the O-RU will attach the same value to the UL U-Plane messages carrying the same sectionId data.
- BandSector_ID: Aggregated cell identifier (distinguishes bands and sectors supported by the O-RU).
- CC_ID: distinguishes Carrier Components supported by the O-RU.
- RU_Port_ID: designates logical flows such as data layers or spatial streams, and logical flows such as separate numerologies (e.g. PRACH) or signaling channels requiring special antenna assignments such as SRS.

The assignment of the DU_port_ID, BandSector_ID, CC_ID, and RU_Port_ID as part of the eAxC ID is done solely by the O-DU via the M-plane. Furthermore, the O-RU doesn't need an explicit definition of any bit-level allocation within any of the four fields of the eAxC ID.

The bitwidth of each of the above fields is variable and set via M-Plane messaging. This is to allow flexibility given it is expected that not all fields will simultaneously need their maximum range for any given O-RU. It is expected the M-Plane message will configure the O-RU and O-DU with the appropriate bitwidth of each of the four fields, and the SMO that does the actual assignment will assure all 16 bits are allocated (with or without padding).

Value range: {0000 0000 0000 0000b-1111 1111 1111 1111b = eAxC ID}

Bit allocations:

0 (msb)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15 (lsb)	Number of Octets
DU_Port_ID				BandSector_ID				CC_ID				RU_Port_ID				2

Type: unsigned integer (concatenated bit fields).

Field length: 16 bits.

5.1.3.1.7 ecpriSeqid (message identifier)

Description: This parameter provides unique message identification and ordering on two different levels. The first octet of this parameter is the Sequence ID, which is used to identify ordering of messages within an eAxC message stream. The Sequence ID field increments and wraps independently for each U-Plane eAxC DL, U-Plane eAxC UL, C-Plane eAxC DL, and C-Plane eAxC UL, even if they share the same eAxC ID. The Sequence ID is used to verify that all messages are received and also to reorder messages that are received out of order. The second octet of this parameter is the Subsequence ID. The Subsequence ID is used to verify ordering and implement reordering when radio-transport-level (eCPRI or IEEE-1914.3) fragmentation occurs. Radio-transport (eCPRI or IEEE-1914.3) fragmentation is a method of splitting U-plane messages containing one or more sections whose length exceeds the maximum packet or message length of the underlying protocol. The Subsequence ID field consists of a 7 bit Subsequence counter and a single bit field, called E-bit. The Subsequence number increments starting from zero for each fragment of a U-plane message. The E bit is used to indicate the last message of the radio-transport level fragments. It is always set to zero except for the last message of the U-plane fragment. In the case of C-plane messages radio-transport fragmentation is not allowed, therefore the Subsequence ID shall be set to zero, and the E bit set to one._ See clause 5.5 for a description of the fragmentation process.

NOTE: As an alternative to radio-transport-level fragmentation, application fragmentation can be implemented. In this case the application can take the responsibility to ensure all transport messages are not too long (fit within the necessary transport payload size). When this “application layer fragmentation” is used, the subsequence identifier shall always be set to “0”, and the E-bit set to “1” (See clause 5.5).

Figure 5-5 shows how the sequence numbers are intended to be generated and checked by C-Plane and U-Plane messaging.

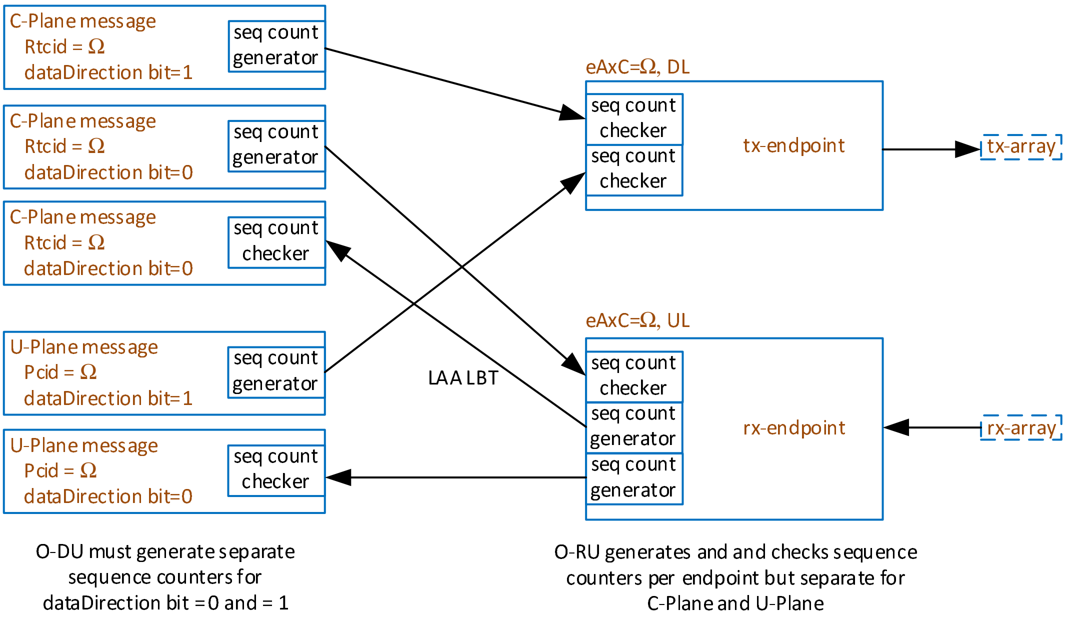


Figure 5-5 : Sequence Count Generation and Checking for C-Plane and U-Plane in O-DU and O-RU

Implications of **Figure 5-5** are as follows:

a) Sequence ID is unique per eAxC (ecpriRtcid or ecpriPcid): different eAxC values have their own Sequence ID generation
b) Sequence ID is generated by the fronthaul interface transmitter in either the O-DU or O-RU
c) The sequence generator and checker for a C-Plane message describing U-Plane DL is independent of the sequence generator and checker for a C-Plane message describing U-Plane UL.

Table 5-2 summarizes this situation consistent with **Figure 5-5**:

Table 5-2 : Description of Shared or Independent Sequence Generators

direction of message flow	**type of flow**	**U-Plane data direction**	**Same/different eAxC between DL & UL**	**shared/independent sequence generator**
O-DU → O-RU	C-plane	DL (dataDirection bit =1)	same or different	independent
O-DU → O-RU	C-plane	UL (dataDirection bit =0)	same or different	independent
O-RU → O-DU	C-plane	UL (dataDirection bit =0)	same or different	independent
O-DU → O-RU	U-plane	DL (dataDirection bit =1)	same or different	independent
O-RU → O-DU	U-plane	UL (dataDirection bit =0)	same or different	independent

As described in clause 5.1.3.1.6, there are O-RUs which do not support, within a same eAxC ID, independent sequence checkers for C-plane messages describing U-plane DL and C-plane messages describing U-plane UL. As also described in clause 5.1.3.1.6, to interoperate with such O-RUs, O-DUs may choose, among other options, to use the same eAxC ID but with a shared sequence generator for C-plane messages describing U-plane DL and C-plane messages describing U-plane UL as described in **Figure 5-6** and **Table 5-3** below.

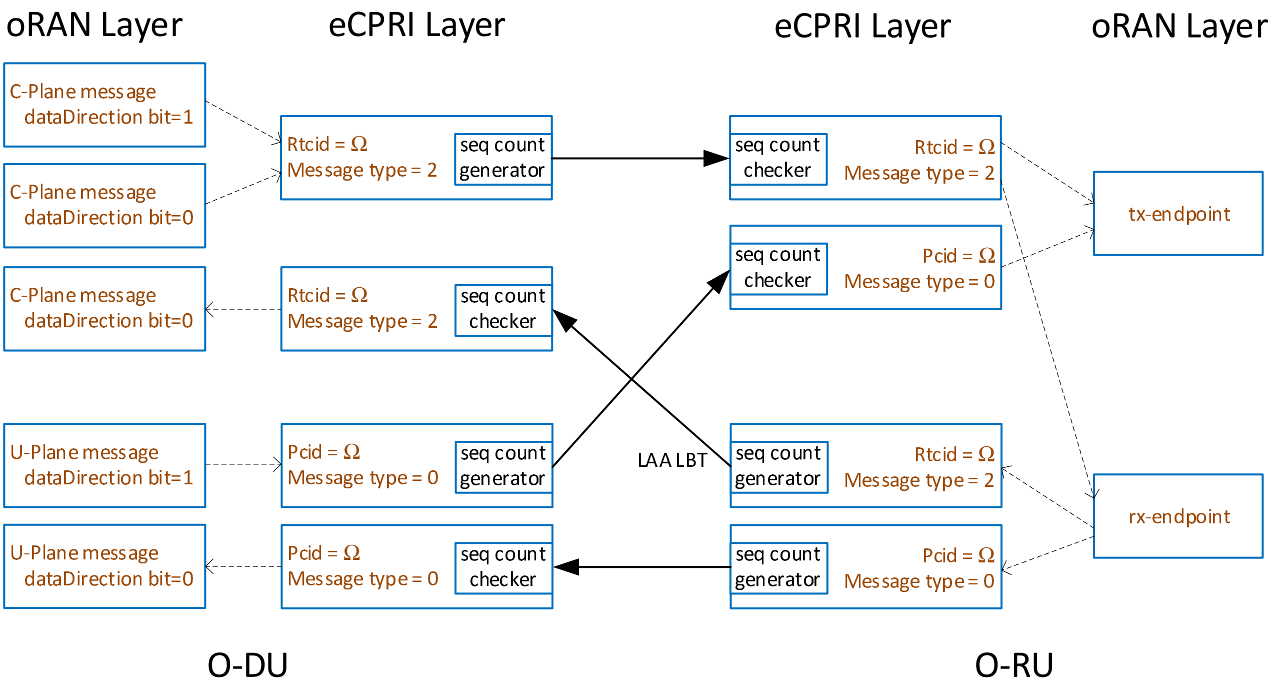

oRAN Layer
eCPRI Layer
eCPRI Layer
oRAN Layer
C-Plane message dataDirection bit=1
C-Plane message dataDirection bit=0
C-Plane message dataDirection bit=0
U-Plane message dataDirection bit=1
U-Plane message dataDirection bit=0
Rtcid = Ω Message type = 2
seq count generator
Rtcid = Ω Message type = 2
seq count checker
Pcid = Ω Message type = 0
seq count generator
Pcid = Ω Message type = 0
seq count checker
seq count checker
Rtcid = Ω Message type = 2
seq count checker
Pcid = Ω Message type = 0
LAA LBT
seq count generator
Rtcid = Ω Message type = 2
seq count generator
Pcid = Ω Message type = 0
tx-endpoint
rx-endpoint
O-DU
O-RU

Figure 5-6 : Optional case of Shared Sequence Generation and Checking for C-Plane and U-Plane in O-DU and O-RU

Table 5-3 : Optional case of Description of Shared Sequence per eAxC id Generators

direction of message flow	**type of flow**	**U-Plane data direction**	**Same/different eAxC between DL & UL**	**shared/independent sequence generator**
O-DU → O-RU	C-plane	DL (dataDirection bit =1)	same	Shared
O-DU → O-RU	C-plane	UL (dataDirection bit =0)	same	Shared
O-RU → O-DU	C-plane	UL (dataDirection bit =0)	same or different	Independent
O-DU → O-RU	U-plane	DL (dataDirection bit =1)	same or different	Independent
O-RU → O-DU	U-plane	UL (dataDirection bit =0)	same or different	Independent

Value range: {0000 0000 0000 0000b-1111 1111 1111 1111b}

Bit allocations:

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
Sequence ID								1	Octet 1
E bit	Subsequence ID							1	Octet 2

Type: unsigned integer (concatenated bit fields)

Field length: Sequence ID: 8 bits; subsequence ID: 7 bits; E-bit: 1 bit.

5.1.3.2 1914.3 Transport Header

As an alternative to eCPRI as a transport header, IEEE 1914.3 may be used. The definition of the 1914.3 transport header is shown in **Table 5-4** below.

Table 5-4 : 1914.3 Transport Header Field Definitions

Section Type : any									
0 (msb)	1	2	3	4	5	6	7 (lsb)	# of bytes	
RoEsubType								1	Octet 1
RoEflowId								1	Octet 2
RoElength								2	Octet 3
RoEorderInfo								4	Octet 5

Each field within the 1914.3 Transport Header is further described in the following clauses.

5.1.3.2.1 RoEsubType (sub type / message type)

Description: This field indicates the payload type within the IEEE 1914.3 Standard for Radio over Ethernet Encapsulations and Mappings (RoE) subType range. RoE allows RoE subTypes in the range 128 to 191 to be mapped to external organizations and companies using a subtype mapping table (below). This table has two fields. The OUI/CID field uses a unique Company ID (CID) value. For O-this version of this specification, the OUI/CID assigned by IEEE to xRAN (0xFAEB6E) shall be used. The payload structure mapping field assigns the RoE subtype to O-RAN message types. When the payload structure mapping field is set to 0xFFFF, this indicates that that RoE subtype is not mapped to an O-RAN message type.

A given O-RAN implementation may choose to have fixed (default) mappings, or alternatively, it may choose to configure the subType mapping table using control packets as described in IEEE 1914.3.

NOTE: The table below indicates the use by O-RAN of the IEEE-defined xRAN subtype mapping; future discussions with IEEE are needed to convert these to O-RAN subtype mapping.

Table 5-5 : RoE Subtype Mapping

RoE subType	OUI/CID subType mapping table (.mapSubtype)	
	OUI/CID Mapping (3 bytes) bit39 <----------------------------------	Payload structure mapping (2 bytes) --> bit 0
128	xRAN=0xFAEB6E	0x0001, IQ (No concatenation)
129	xRAN=0xFAEB6E	0x0002, IQ (With concatenation)
130	xRAN=0xFAEB6E	0x0003, Ctrl (No concatenation)
131	xRAN=0xFAEB6E	0x0004, Ctrl (With concatenation)
132 to 191	xRAN=0xXXXXXX (don’t care)	0xFFFF (IEEE1914.3 default), unused/unmapped by xRAN in this version of the O-RAN specification.

Value range: 128 to 191. Default values are shown below.

Type: unsigned integer.

Field length: 8 bits.

Valid Values: 128 (U-Plane, no concat), 129 (U-Plane, with concat), 130 (C-Plane, no concat), 131 (C0-Plane, with concat).

5.1.3.2.2 RoEflowID (flow identifier)

Description: The RoEflowID is a mechanism which can identify specific flows between end-points. RoEflowID, 0xFF is reserved for RoE control packets. O-RAN has no current use for this field.

Value range: 0 – 0xFE.

Type: unsigned integer.

Field length: 8 bits.

Description: This field is currently unused.

5.1.3.2.3 RoElength (length)

Description: This field is the size in bytes of the payload part of the message. The payload length field value is the total number of octets following the O-RAN common header. It does not include the Ethernet FCS or following bytes.

Value range: 0 – 0xFFFF.

Type: unsigned integer.

Field length: 16 bits.

5.1.3.2.4 RoEorderInfo (order information)

Description: This field is split into seven sub-fields.

Table 5-6 : RoE RoEorderInfo MappingField

Field	Length	Note
DU_Port_ID	16 bits	Used to differentiate processing units at O-DU (e.g., different baseband cards). It is expected the O-DU will assign these bits, and the O-RU will attach the same value to the UL U-Plane messages carrying the same sectionId data. See sub-clause 3.1.3.1.6 for further information.
BandSector_ID		Aggregated cell identifier (distinguishes bands and sectors supported by the O-RU). See sub-clause 3.1.3.1.6 for further information.
CC_ID		Distinguishes Carrier Components supported by the O-RU. See sub-clause 3.1.3.1.6 for further information.
RU_Port_ID		Used to differentiate spatial streams or beams on the O-RU. See sub-clause 3.1.3.1.6 for further information.
Sequence_ID	8 bits	Unique message ordering sequence. See sub-clause 3.1.3.1.7 for further information.
E_Bit	1 bit	Marks the last message pertaining to the section. See sub-clause 3.1.3.1.7 for further information.
Subsequence_ID	7 bits	Unique message ordering sub-sequence. See sub-clause 3.1.3.1.7 for further information.

Value range: 0 to 0xFFFF FFFF.

Type: unsigned integer (concatenated bit fields).

Field length: 32 bits.

5.2 Protocol Architecture

5.2.1 C-plane

Figure 5-7 depicts the protocol stack for C-Plane. Data can be optionally transmitted over IP Layer 3 if supported by the transmitting and receiving nodes.

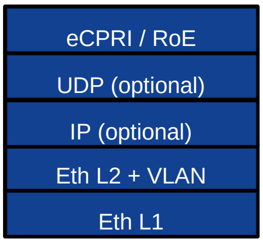


Figure 5-7 : C-plane protocol structure

5.2.2 U-plane

Figure 5-8 depicts the protocol stack for U-Plane. Data can be optionally transmitted over IP Layer 3 if supported by the transmitting and receiving nodes.

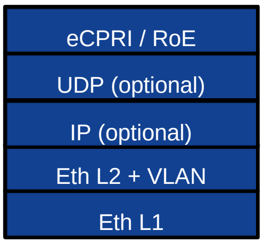


Figure 5-8 : U-plane protocol structure

5.2.3 S-plane

Frequency and time synchronization of O-DUs and O-RUs via Ethernet use Synchronous Ethernet and IEEE 1588 Precision Time Protocol (PTP). Transport of PTP directly over L2 Ethernet (ITU-T G.8275.1 full timing on-path support) is assumed in this version of the specification, whilst transport of PTP over UDP/IP (ITU-T G.8275.2 partial timing support from the network) is also possible albeit with unassured synchronization performance. Security mechanisms will not be used for this plane. The protocol stack for PTP and SyncE over L2 Ethernet is depicted in **Figure 5-9**.

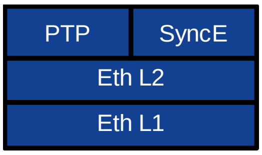


Figure 5-9 : S-plane protocol structure

5.3 Quality of Service

The LLS interface needs to support the ability to distinguish between data flows with different QoS requirements. Configurable priority levels (via the M-Plane) for traffic prioritization of flows shall be supported on each node on the network path. Default values for the respective O-RAN planes are indicated. Priority marking per packet is needed in each protocol layer except for protocols that do not allow such tagging e.g. ITU-T G.8264 for SyncE and ITU-T G.8275.1 for full timing support PTP. For operation at Layer 2, prioritization is performed by specifying a configurable value for the Priority Code Point (PCP) tag in the IEEE 802.1q VLAN header on the outgoing traffic.

Table 5-7 : Quality of service classes

Plane	L2 CoS Priority (range 0-7)	L3 DSCP Code	Preemption(1)
S-Plane	G.8264: N/A G.8275.1: N/A G.8275.2: Default 7	G.8264: N/A G.8275.1: N/A G.8275.2: EF (Expedited Forwarding)	non-preemptable
U-Plane	Default: 7	EF (Expedited Forwarding)	non-preemptable
C-Plane	Default: 7	EF (Expedited Forwarding)	non-preemptable
M-Plane	Default: 2	AF2x (Assured Forwarding)	preemptable
Other traffic	Default: 1	BE (Best Effort)	preemptable
(1) Not all networks will support preemption so this only applies to networks supporting preemption (2) applies also to LBP [7] (3) applies also to ARP, if used, where IPv4 is used			

For U-Plane separate priorities can be supported based on ecpriRtcid / ecpriPcid (or the equivalent fields in RoE packet headers). This allows for different prioritization of some channels (e.g. PRACH/ SRS), or services (URLLC). This requires the ability to configure priority assignments to ecpriRtcid / ecpriPcid via M-Plane messages. The default U-Plane priority can be applied to flows not specifically configured with a different value via the M-Plane. If a node does not support configurable ecpriRtcid / ecpriPcid priorities, then all U-Plane traffic will use the default value.

QoS failure (dropped packets) may be detected using the sequence numbers that make up part of the transport header. Loss of a packet in the U-Plane will generally impact only a specific symbol except in rare cases where a multi-symbol U-Plane message is sent (using the "symInc" field). Loss of a C-Plane packet is more serious, with the possibility of impacting an entire slot's worth of data. A broader impact is also possible with C-Plane packet loss: if downloaded beamforming weights, beamforming attributes, or channel information is lost then a beamId may be incorrectly associated with an "old" set of beamforming weights thereby possibly impacting multiple slots. However, packet loss can be detected via the sequence number so a recovery action may be taken to re-send messages that may have a persistent impact.

5.4 Data Flow Identification

Differentiation between a combined U/C-Plane data flow for a single eAxC_ID and Management Plane data flow traffic can be achieved using the following options:

- Data flow separation based on TCP/UDP (applicable when layer 3 transport is used for the C/U-plane)
- Data flow separation based on VLAN (applicable when layer 2 or layer 3 is used for the C/U-plane transport)
    - NOTE: The mechanism for assigning a VLAN ID to the combined U-Plane and C-Plane data flow for a given eAxC_ID is assumed to be via the M-Plane
- Data flow separation based on usage of different MAC addresses (applicable when Layer 2 is used for the combined C/U plane transport)
    - e.g., one MAC address used for the combined C/U-Plane data flow and a second MAC address used for the M-Plane data flow or additional MAC addresses used for the U-Plane for baseband load sharing purposes
- Data flow separation based on different EtherTypes (applicable when Layer 2 is used for C/U plane transport)

The U-plane application also needs to uniquely associate different data flows (e.g. spatial streams) each with a unique U/C plane endpoint address. This can be achieved in an O-RU using the eAxC identifier, and in the O-DU using the eAxC identifier in combination with transport-based endpoint identifiers to differentiate O-RUs. In addition, O-RU data flows can be switched/routed to different O-DUs (or different O-DU ports or O-DU processors) according to the transport-based identifiers associated with an eAxCid (referred to as processing-elements in the WG4 M-Plane

Specification) to allow frames/packets to be switched/routed by network equipment with no visibility of the eAxC values carried in the eCPRI/1914.3 header.

Different transport identifiers (e.g. UDP/IP, VLAN, MAC) can be used based on whether layer 2 or layer 3 transport is used for the U/C plane:

- Separation of the combined U/C-plane data flow using UDP-port identifiers (applicable when layer 3 transport is used for the C/U-plane)
- Separation of the combined U/C-plane data flow using VLAN identities (applicable when layer 2 or layer 3 is used for the C/U-plane transport)
- Separation of the combined U/C-plane data flow using different MAC addresses (applicable when Layer 2 is used for C/U plane transport)

5.5 Fragmentation

Fragmentation is applied in case data (U-Plane or C-Plane data) with Ethernet transport overheads to be transferred exceed maximum transmission unit (MTU) of the network. This specification allows two methods for fragmentation, application layer fragmentation and transport layer fragmentation:

5.5.1 Application layer fragmentation

- Application creates C-plane or U-Plane messages, which when including overheads can fit to MTU requirements set by network
- Sequence ID: Sequence ID increases for every message, E=1, Subsequence ID=0

An example of application fragmentation is provided in **Figure 5-10**, wherein a long data section is split between multiple U-Plane messages. In the example it is a single data section (0xABC) so described as such in the C-Plane, and in the U-Plane the sectionId is cited multiple times in the multiple U-Plane messages but with differing and consecutive groups of PRBs as indicated by the startPrbu and numPrbu fields. The I and Q word width in the example is 14 bits, which for such a large data section requires multiple U-Plane messages. The ecpriRtcid / ecpriPcid (0xDEF3) is shown with a random example value, as is the sequence ID in the C-Plane (0x99) and independently in the U-Plane (0x40-0x48).

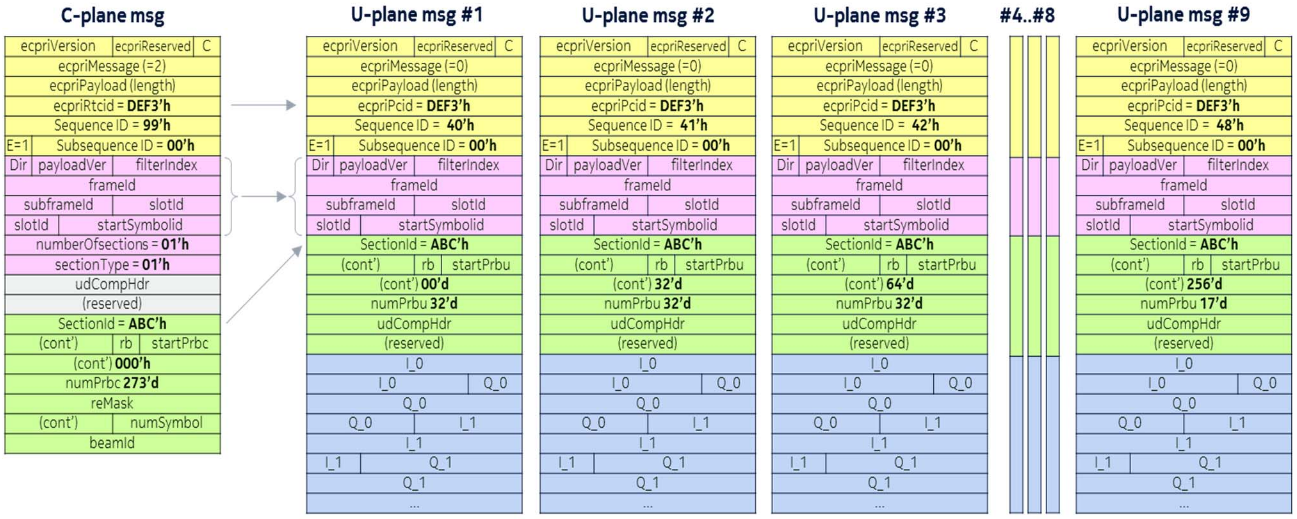


Figure 5-10: Example of Application-Level Fragmentation of U-Plane Messages

5.5.2 Radio Transport layer (eCPRI or IEEE-1914.3) fragmentation

- Application creates U-Plane messages, which when including overhead may exceed MTU requirements set by network
- Radio transport layer splits message which may contain more than one section into pieces such that the fragments with overheads fit to MTU requirements set by network.
- Sequence ID: Sequence ID remains same for all the fragments. Subsequence ID starts from 0 for the first fragment and counts up for each fragment. Last fragment flagged with E=1, others E=0.

Table 5-8 : Example of Sequence Numbers Usage

Application Layer Fragmentation			
eAxC	**SeqId**	**E**	**Sub-SeqId**
Ω	0	1	0
Ω	1	1	0
Ω	2	1	0
Ω	3	1	0
Ω	4	1	0
Ω	5	1	0
Ω	6	1	0
Ω	7	1	0
Ω	8	1	0
Ω	9	1	0

eCPRI Layer Fragmentation			
eAxC	**SeqId**	**E**	**Sub-SeqId**
Ω	0	0	0
Ω	0	1	1
Ω	1	0	0
Ω	1	0	1
Ω	1	0	2
Ω	1	1	3
Ω	2	0	0
Ω	2	0	1
Ω	2	1	2
Ω	3	1	0

5.5.3 Fragmentation Guideline

Application layer fragmentation should be applied so that maximum size standard IEEE 802.3 Ethernet frames can be used. (Jumbo frames, if used, increase the maximum MTU size).

- In case of L2 only solution, application layer maximum transmission unit size is standard IEEE 802.3 Ethernet frame payload size (1500 bytes) – transport overhead (8 bytes) = 1492 bytes (or larger for Jumbo frames)

When a U-plane section payload is fragmented due to large number of PRBs, it will be divided into multiple groups of PRBs such that each group (including the application headers) can fit to the MTU requirements. Each group of PRBs shall have its respective control fields including the same section ID from the C-plane and startPrbu and numPrbu to identify its contents (See **Figure 5-10**). When an application layer message exceeds the MTU size set via the M-Plane configuration, radio-transport layer fragmentation can be applied at the transport layer and has the advantage of allowing application layer messages to fit whatever is the MTU size without the application having to know about transport-level details.

6 Security

6.1 General

Security requirements are depicted in Table 6-1.

Table 6-1 : Security requirements for User-Plane, Control-Plane, and Synchronization-Plane

Plane	Integrity (protection from modifications)	Confidentiality (encryption protection)	Availability (protection from packet insertion)	Remarks
U-Plane	No requirement	No requirement	No requirement	User data protected end to end via PDCP protocol
C-Plane	No requirement	No requirement	No requirement	
S-Plane	No requirement	No requirement	No requirement	Optional in IEEE 1588 (PTP). However, not feasible at a reasonable cost.

NOTE: This note contains clarifying comments which have no normative impact on clause 6 of the present document. The term "no requirement" is intended to be interpreted as meaning that "the present document does not provide security requirements on the fronthaul interface implementation".

7 C-plane Protocol

7.1 General

7.2 Function

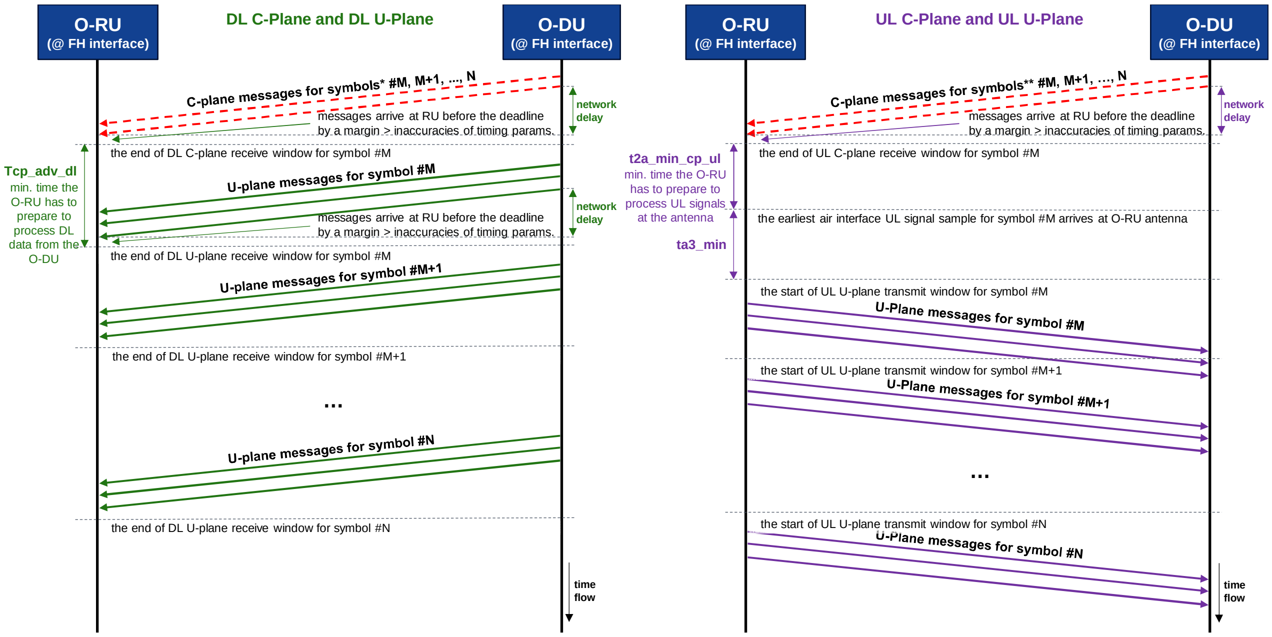
7.2.1 C-Plane Transport

Either eCPRI or IEEE 1914.3 is used as an encapsulation mechanism for the control-plane messages. Due to the nature of these messages (very strict delay constraints), it is assumed that message acknowledgements are not possible. Likewise, it is assumed that a different data flow is used other than the U-Plane channel. Further, C-Plane messages are not concatenated with U-Plane messages within same Ethernet frame.

7.3 Elementary Procedures

7.3.1 Scheduling and Beamforming Commands Transfer procedure

This procedure is used to exchange C-Plane messages between O-DU and O-RU. The main purpose of these messages is to transmit data-associated control information required for processing of user data (e.g., scheduling and beamforming commands) if such information is not provided via M-Plane (see clause 8.2.2 for more details). Messages are sent separately for DL related commands and UL related commands (see **Figure 7-1**). See also **Figure 7-8** for the special case of LTE LAA wherein there are UL as well as DL C-Plane message flows. For purpose of increased flexibility, C-Plane messages may be sent either jointly or separately depending on the channel for which information is conveyed. For example, PUCCH and PUSCH may be bundled or not bundled into a single C-Plane message depending on implementation.


O-RU (@ FH interface)
DL C-Plane and DL U-Plane
O-DU (@ FH interface)
C-plane messages for symbols* #M, M+1, ..., N
network delay
messages arrive at RU before the deadline by a margin > inaccuracies of timing params.
the end of DL C-plane receive window for symbol #M
Tcp_adv_dl
min. time the O-RU has to prepare to process DL data from the O-DU
U-plane messages for symbol #M
network delay
messages arrive at RU before the deadline by a margin > inaccuracies of timing params.
the end of DL U-plane receive window for symbol #M
U-plane messages for symbol #M+1
the end of DL U-plane receive window for symbol #M+1
...
U-plane messages for symbol #N
the end of DL U-plane receive window for symbol #N
time flow
O-RU (@ FH interface)
UL C-Plane and UL U-Plane
O-DU (@ FH interface)
C-plane messages for symbols** #M, M+1, ..., N
network delay
messages arrive at RU before the deadline by a margin > inaccuracies of timing params.
the end of UL C-plane receive window for symbol #M
t2a_min_cp_ul
min. time the O-RU has to prepare to process UL signals at the antenna
the earliest air interface UL signal sample for symbol #M arrives at O-RU antenna
ta3_min
the start of UL U-plane transmit window for symbol #M
U-Plane messages for symbol #M
the start of UL U-plane transmit window for symbol #M+1
U-Plane messages for symbol #M+1
...
the start of UL U-plane transmit window for symbol #N
U-Plane messages for symbol #N
time flow

*) A DL C-plane message describing multiple symbols must arrive at O-RU at least **Tcp_adv_dl** before the end of DL U-plane receive window for symbol **startSymbolId** (the earliest symbol described by the message).

) A UL C-plane message describing multiple symbols must arrive at O-RU at least **t2a_min_cp_ul before the earliest air interface UL signal sample of symbol **startSymbolId** (the earliest symbol described by the message) arrives at O-RU antenna.

Figure 7-1 : Scheduling and beamforming commands transfer procedure

7.3.2 Mixed Numerology and PRACH Handling

5G NR defines Physical Resource Block (PRB) where the number of subcarriers per PRB is the same for all numerologies (twelve). However, subcarrier spacing may differ, resulting in mixed numerology used in time and frequency domains respectively. Likewise, mixed numerologies may be applied to both DL and UL respectively **(Figure 7-2**).

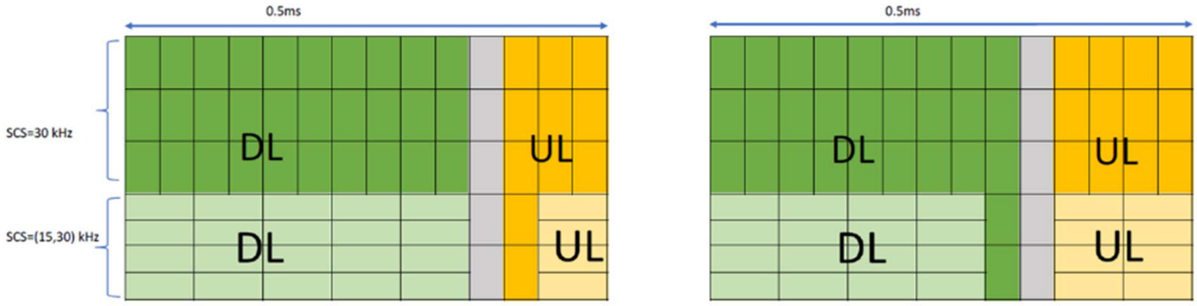


Figure 7-2 : Mixed Numerology in Time and Frequency Domain

Frequency indexing across multiple numerologies

- RB indexing (i.e. the physical resource block) is always dependent on the used numerology. For PRACH and SSB the relation is with the physical channel. In case of PRACH, indexing of RBs inside of each PRACH frequency block follows 3GPP specifications and PRACH subcarrier extraction is performed in the O-DU. In particular, the first RE of the first RB addressed inside section Id shall correspond with the first guard tone used at the lower edge of the PRACH frequency block based on the PRACH SCS. The O-RU, using the filter index and SCS of corresponding PUSCH, knows the PRACH subcarrier position and guard tones and can then adjust its filtering parameters (e.g. filter center, passband). For the SSB case, its own separate frequency offset is assigned.
  Figure 7-3 shows a PRACH example illustrating startPrbc, numPrbc, and freqOffset. Specifically,
  - freqOffset indicate the location of lowest RE’s center in the lowest RB defined by frameStructure, with respect to center-of-channel-bandwidth.
  - startPrbc and numPrbc in the Section Type 3 indicate the PRB in the RB grid defined by corresponding freqOffset and frameStructure.
    - If freqOffset refers to the lowest RE’s center in the lowest RB of the PRACH block (as shown in Figure 5-3), then startPrbc shall be set to 0.
    - If freqOffset refers to lower than the start of the PRACH block, then startPrbc will take a non-zero value, where indexing starts from the PRB referred to by freqOffset.
  - Note that numPrbc may exclude PRBs with all-guard-tones at the upper end of the PRACH frequency block. (Hence, in the example in Figure 5-3, numPrbc could be set to 71 or 72).

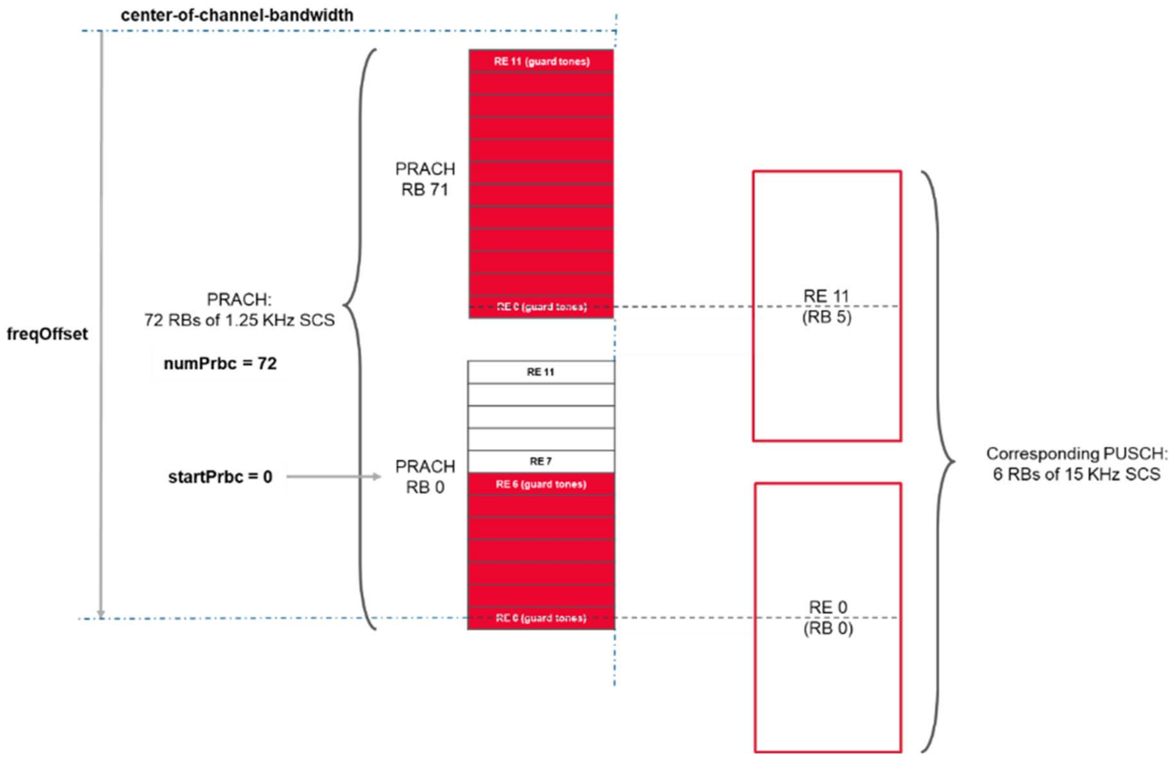
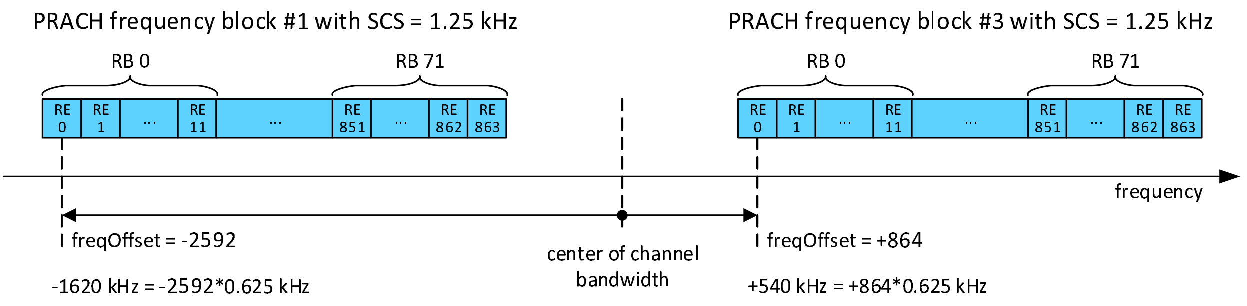


Figure 7-3: startPrbc, numPrbc, and freqOffset Illustration - PRACH example

- For each numerology (and PRACH/SSB), the freqOffset IE determines frequency offset between center of channel bandwidth (configured via M-plane) and center of subcarrier corresponding to RE#0 of RB#0 with resolution of half the SCS of the respective numerology. This concept is depicted in **Figure 7-4**, **Figure 7-5** and **Figure 7-6**.
  - The center of channel bandwidth (component carrier center frequency in Hz) is the common reference to all numerologies and PRACH/SSB channels. The center of channel bandwidth is configured at carrier setup over M-Plane.
  - Frequency offset resolution of 0.5 x SCS allows center of channel bandwidth to be aligned with an RE center, or and RE edge.


PRACH frequency block #1 with SCS = 1.25 kHz
PRACH frequency block #3 with SCS = 1.25 kHz
RB 0
RB 71
RE 0
RE 1
...
RE 11
...
RE 851
...
RE 862
RE 863
frequency
freqOffset = -2592
-1620 kHz = -2592*0.625 kHz
center of channel bandwidth
freqOffset = +864
+540 kHz = +864*0.625 kHz

Note: Each frequency block in sent with separate section ID and control setting in section type 3
Note: In this example, 5G NR 100 MHz carrier PUSCH SCS is 30 kHz, PRACH SCS is 1.25 kHz
Note: In 5G NR PRACH frequency blocks are continuous
Note: In the example guard tones are included.

Figure 7-4 : RB Mapping– PRACH Example

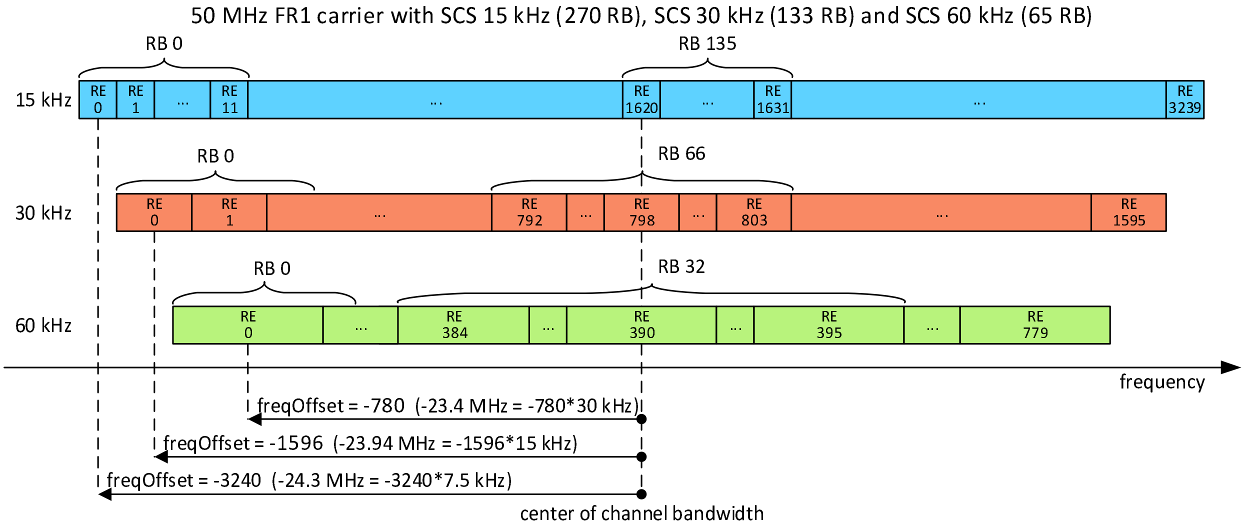

50 MHz FR1 carrier with SCS 15 kHz (270 RB), SCS 30 kHz (133 RB) and SCS 60 kHz (65 RB)
RB 0
RB 135
15 kHz
RE 0
RE 1
RE 11
RE 1620
RE 1631
RE 3239
RB 0
RB 66
30 kHz
RE 0
RE 1
RE 792
RE 798
RE 803
RE 1595
RB 0
RB 32
60 kHz
RE 0
RE 384
RE 390
RE 395
RE 779
frequency
freqOffset = -780 (-23.4 MHz = -780*30 kHz)
freqOffset = -1596 (-23.94 MHz = -1596*15 kHz)
freqOffset = -3240 (-24.3 MHz = -3240*7.5 kHz)
center of channel bandwidth

Figure 7-5 : RB Mapping and Support of Mixed Numerologies - Example with Mixed Numerologies

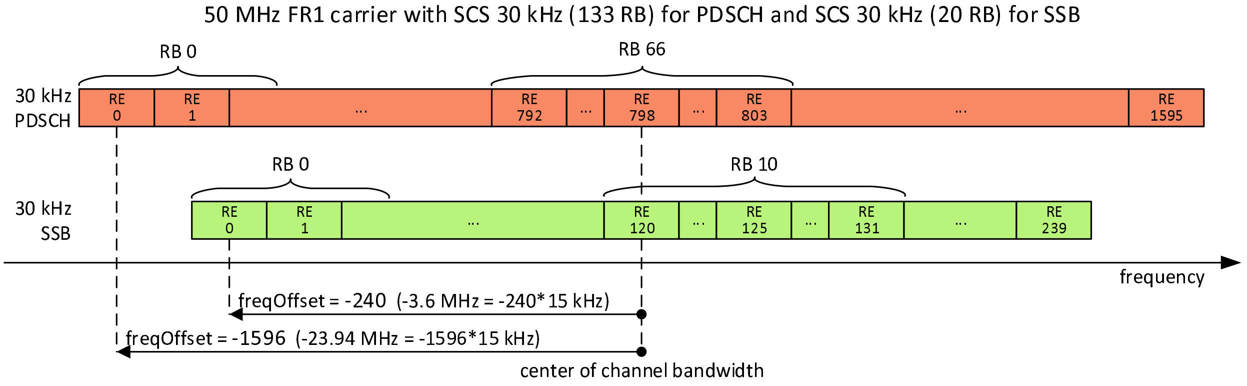

50 MHz FR1 carrier with SCS 30 kHz (133 RB) for PDSCH and SCS 30 kHz (20 RB) for SSB
RB 0
RB 66
30 kHz PDSCH
RE 0
RE 1
RE 792
RE 798
RE 803
RE 1595
RB 0
RB 10
30 kHz SSB
RE 0
RE 1
RE 120
RE 125
RE 131
RE 239
frequency
freqOffset = -240 (-3.6 MHz = -240*15 kHz)
freqOffset = -1596 (-23.94 MHz = -1596*15 kHz)
center of channel bandwidth

Figure 7-6 : RB Mapping and Support of Mixed Numerologies - Example SSB

Slot indexing with mixed numerologies

For a given frequency range FR1 or FR2, the highest possible numerology supported by the O-RU shall be used as the common reference per component carrier for the start of the slot identified by slotId. UL and DL on the same component carrier shall use the same reference numerology for slotId. If the highest numerology supported by the O-RU allows both normal and extended CP then normal CP shall be used as reference. Note that the O-RU shall advance slots with extended CP against the reference. The symbol duration and position in time is calculated from the μ value (SCS from frameStruture in message field or configured via M-plane) and the slotId field in the CU-Plane message. The value of the sectionId field in CU-Plane messages addressed per eAxC shall be unique per slot identified by slotId value.

For example, in FR1 case, the maximum slot numbers per subframe is four, when the maximum supported SCS by the O-RU is 60kHz

- μ= 0 → slot #0; μ=1 → slot #0 & #2; μ=2 → slot #0, #1, #2, #3 (see **Table 7-1**)

Table 7-1 : slotId Indexing (example 1)

μ=2 60 kHz (Highest supported numerology)		μ=1 30 kHz		μ=0 15 kHz	
slotId	symbolId	slotId	symbolId	slotId	symbolId
0	0	0	0	0	0
	1				
	2		1		
	...				
	11		2		1
	12				
	13		...		
1	0				
	1		11		2
	2				
	...		12		
	11				
	12		13		...
	13				
2	0	2	0		
	1				
	2		1		11
	...				
	11		2		
	12				
	13		...		12
3	0				
	1		11		
	2				
	...		12		13
	11				
	12		13		
	13				
0	0	0	0	0	0
	1				
	2		1		
	...				
	11		2		1
	12				
	13				

...

μ=3 120 kHz (Highest supported numerology)		μ=2 60 kHz		μ=1 30 kHz	
slotId	symbolId	slotId	symbolId	slotId	symbolId
0	0	0	0	0	0
	1		1		
	...		2		1
	12				
	13		...		
1	0		11		2
	1				
	...		12		
	12		13		...
	13				
2	0	2	0		
	1		1		11
	...		2		
	12				
	13		...		12
3	0		11		
	1				
	...		12		13
	12		13		
	13				
4	0	4	0	4	0
	1		1		
	...		2		1
	12				
	13		...		
5	0		11		2
	1				
	...		12		
	12		13		...
	13				
6	0	6	0		
	1		1		11
	...		2		
	12				
	13		...		12
7	0		11		
	1				
	...		12		13
	12		13		
	13				
0	0	0	0	0	0
	1		1		
	...		2		1
	12				
	13		...		
1	0		11		2
	1				
	...		12		
	12				

...

Table 7-2: slotId Indexing (example 2)

μ=4 240 kHz (Highest supported numerology)		μ=3 120 kHz	
slotId	symbolId	slotId	symbolId
0	0 1 ... 12 13	0	0 1 2 ... 11 12 13
1	0 1 ... 12 13		
2	0 1 ... 12 13	2	0 1 2 ... 11 12 13
3	0 1 ... 12 13		
4	0 1 ... 12 13	4	0 1 2
...			
	... 12 13		12 13
14	0 1 ... 12 13	14	0 1 2 ... 11 12 13
15	0 1 ... 12 13		
0	0 1 ... 12 13	0	0 1 2 ... 11 12
1	0 1 ... 12		
...			

μ=1 30 kHz (Highest supported numerology)		μ=0 15 kHz	
slotId	symbolId	slotId	symbolId
0	0 1 2 ... 11 12 13	0	0 1 2 ... 11 12 13
1	0 1 2 ... 11 12 13		
0	0 1 2 ... 11 12 13	0	0 1 2 ... 11 12 13
1	0 1 2 ... 11 12 13		
0	0 1 2 ... 11 12	0	0 1 2
...			

Numerology for slotId is based on the highest possible numerology supported by the O-RU. For example:

- If the O-RU supports 15kHz SCS, 30kHz SCS and 60kHz SCS, it will be as per the left hand side table of **Table 7-1**;
- If the O-RU supports 30kHz SCS, 60kHZ and 120kHz SCS, it will be as per the right hand side table of **Table 7-1**;
- If the O-RU supports 120kHz SCS and 240kHz SCS, it will be as per the left hand side table of **Table 7-2**;
- If the O-RU supports 15kHz SCS and 30kHz SCS, it will be as per the right hand side table of **Table 7-2**;

PRACH formats with multiple repetitions of preamble

Certain PRACH formats lead to PRACH symbols to be constructed from multiple repetitions of a preamble sequence, with only the Cyclic Prefix (CP) used with the first sequence. Therefore, the O-RU shall be informed how to correctly execute CP extraction and FFT. This is achieved by sending a single control message spanning over multiple symbols (e.g. example depicted in Figure 5-6: number of symbols = 4, CP length = 0, time offset duration is adjusted by an equivalent time value of 1152 samples to compensate for setting CP length =0), which reduces the number of C-Plane messages and data sections required.

Optionally O-RU can advertise the list of the supported formats for the specific endpoint. If the O-RU chooses not to, it means O-RU supports all PRACH formats and the O-DU may select whatever format that is 3GPP compliant with the selected SCS. If the O-RU implements earlier releases of the M-plane specification which does not support such reporting, then the negotiation of PRACH formats to use shall happen offline, i.e. operators and/or vendors have to ensure that the intended PRACH formats are supported by both O-RU and O-DU.

The example in **Figure 7-7** depicts the scenario with format A2 and 30kHz SCS, in which only CP extraction occurs once, yet there are four associated FFT operations.

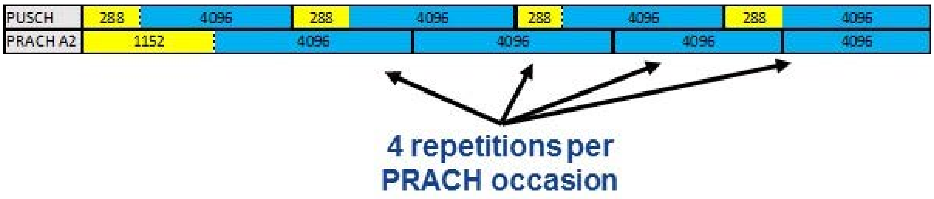


Figure 7-7 : Example of PRACH Format A2

7.3.3 DL Precoding configuration parameters and indications

This clause addresses the method of specifying the precoding operation in the O-RU, which is expected when Category B O-RUs are supported. For Category A O-RUs the precoding is implemented in the O-DU so the descriptions in this clause are not relevant. Annex I contains more information and examples regarding this precoding.

O-DU

12 REs are generally sent on the interface from the O-DU to the O-RU but for certain IQ formats (selective RE sending as described in Annex A.1), fewer REs may be sent. In this case the missing REs would be considered by the O-RU to be equal to zero in both I and Q.

For ‘single Tx’:

- For layer mapping at O-RU, complex-valued modulation symbols mapped in a sequence starting with $d^{(q)}(0),...,d^{(q)}(M^{(q)}_{\text{symb}}-1)$ to frequency REs (k,l) and are packed into a PRB. A single eAxC is used for this purpose.
- For CRS mapping at O-RU, all CRS REs belong to a single Tx antenna and are mapped to frequency REs (k,l) for one layer and are packed into a PRB for transmission and are unpacked at O-RU (see Annex I for details).

For TxD:

- For layer mapping at O-RU, complex-valued modulation symbols mapped in a sequence starting with $d^{(q)}(0),...,d^{(q)}(M^{(q)}_{\text{symb}}-1)$ to frequency REs (k,l) and are packed into a PRB. A single eAxC is used for this purpose.
- For CRS mapping at O-RU, all CRS RE’s are mapped to frequency REs (k,l) for all layers and are packed into a PRB for transmission and are unpacked at O-RU (see Annex I for details).

For TM3/TM4, TM5/6

- At the O-DU, layer mapped symbols for each layer v, mapped in sequence starting with $x(i)=\begin{bmatrix}x^{(0)}(i) & \ldots & x^{(\upsilon-1)}(i)\end{bmatrix}^T$ to frequency REs (k,l) are packed into each PRB (see Annex I).
- For precoding at the O-RU, different eAxCs are used for each layer (See Annex I).

- For CRS mapping at the O-RU, all CRS RE's for each layer are packed into a PRB for transmission and are unpacked at the O-RU (see Annex I for details).
- All C-plane message parameters are kept the same for precoding purposes.

For TM7-10 and NR

- Precoding in the O-RU may be implemented in various ways that are vendor-defined and vendor-specific, because there is no 3GPP-mandated precoding operation for these cases.

O-RU

From the C-plane precoding Section Extension, the O-RU determine the transmission scheme using txScheme field.

For 'single Tx':

- At the O-RU, input modulated symbols $d^{(q)}(0),...,d^{(q)}(M_{\text{symb}}^{(q)}-1)$ are to be unpacked and used to perform layer mapping, precoding and antenna port mapping for single tx.
- For transmission on a single antenna port, a single layer is used, $\upsilon=1$, and the mapping is defined as $x^{(0)}(i)=d^{(0)}(i)$ with $M_{\text{symb}}^{\text{layer}}=M_{\text{symb}}^{(0)}$.
- For single tx transmission on a single antenna port, precoding is defined by $y^{(p)}(i)=x^{(0)}(i)$ where $p\in\{0,4,5,7,8,11,13,107,108,109,110\}$ is the number of the single antenna port used for transmission of the physical channel and $i=0,1,...,M_{\text{symb}}^{\text{ap}}-1$, $M_{\text{symb}}^{\text{ap}}=M_{\text{symb}}^{\text{layer}}$.
- For antenna port mapping p={0}, each y(i)=[$y^{(p)}$ (i) $]^{T}$ RE goes to antenna port y_p(i) after antenna port mapping.
- Since the PRB contains CRS sequences for one antenna port, the RE should extract the CRS RE's using crsSymbolNumber, crsReMask and crsShift (see Annex I for details) and are mapped to the appropriate RE position.

For txScheme 'TxD':

- At the O-RU, input modulated symbols $d^{(q)}(0),...,d^{(q)}(M_{\text{symb}}^{(q)}-1)$ are to be unpacked and used to perform layer mapping, precoding and antenna port mapping.
- The appropriate precoder is selected based on number of layers and antenna ports.
- For antenna port mapping p={0..N}, each y(i)=[$y^{(p)}$ (i) $]^{T}$ RE goes to each antenna port y_p(i) after antenna port mapping.
- Since the PRB contains CRS sequences for N antenna ports, the RE should extract the CRS RE's using crsSymbolNumber, crsReMask and crsShift (see Annex I for details) which are mapped to the appropriate RE position and rest of the REs are populated with zero data.

For TM3/TM4, TM5/6

- Input layer mapped symbols $x(i)=\left[x^{(0)}(i) \quad ... \quad x^{(\upsilon-1)}(i)\right]^{T}$ shall be used to perform precoding at the O-RU based on codeBookIndex, numLayers, layerID.
- In closed loop mode, the appropriate precoder is selected per codebook index, number of layers and antenna ports.
- In open loop mode, the codebook index field is ignored.
- The O-RU changes the precoder per RE automatically based on the number of antenna ports and number of layers.
- After precoding, for antenna port mapping p={0..N}, each y(i)=[y $^{(p)}$ (i) $]^{T}$ RE goes to each antenna port y_p(i) after antenna port mapping.
- Since all PRBs contain CRS sequences for N antenna ports, then based on the layerID (layer 0) extract CRS sequence using crsSymbolNumber, crsReMask and crsShift (see Annex I) for CRS mapping to each of the antenna ports using the reMask bit field; the CRS REs from other layers can be ignored.

For TM7-10 and NR

- One way to implement precoding in the O-RU is via the beamID values, wherein a beamID points to a beamforming vector that also implements the precoding operation.

7.3.4 LAA Commands Transfer procedure

This procedure is used to exchange C-Plane messages between O-DU and O-RU. The main purpose of these messages is to support LAA feature in the O-RU/O-DU. See Annex G for more details on the LAA message flow.

7.3.4.1 LBT procedure overview

The LBT procedure is used to configure the O-RU with the parameters needed to do LBT prior to PDSCH or DRS transmission OTA. The O-RU needs to report the LBT process outcome (either success or failure) in the indication message.

In contrast to licensed spectrum, where the O-RU may continuously send data or reference signals OTA, in unlicensed spectrum, the O-RU can only send discontinuous bursts of data (of length MCOT) or periodic DRS signals. The LBT_DL_CONFIG request message is sent once prior to each OTA transmission on the LAA SCell. This includes both DRS and PDSCH (i.e., MCOT burst) transmissions. For every LBT_DL_CONFIG request message, the O-RU replies with the LBT_DL indication messages which include the LBT outcome and the status of the subframe (transmitted/dropped) (see **Figure 7-8**).

If O-RU can manage a congestion window by itself, the O-DU does not need to send LBT_DL_CONFIG request message to start the LBT procedure at O-RU. Because O-RU can determine the start timing of LBT procedure, the requirements of O-DU for the LBT procedure at the O-RU is a delivery of needed information to adjust a congestion window managed by O-RU. To transmit these information, O-DU sends LBT_CWCONFIG_REQ message, and O-RU notifies to the O-DU by LBT_CWCONFIG_RSP message to indicate that the LBT_CWCONFIG_REQ message is successfully received or not. Also, O-RU can adjust its congestion window based on the information on the LBT_CWCONFIG_REQ message. For every LBT_CWCONFIG_REQ request message, the O-RU replies with the LBT_DL indication message(s) which include the LBT outcome and the status of the subframe (transmitted/dropped).

For the DRS signal, initially, the O-DU shall configure the O-RU with the DRS parameters (DMTC period and DMTC offset) via the M-plane.

Before every DRS period, the O-DU shall send the LBT_DL_CONFIG request message and the DRS signal. O-RU does LBT and transmits the DRS signal OTA in case of LBT success. The O-RU then shall send the LBT_DL indication.

The DRS signal can be transmitted over the fronthaul interface similar to other reference signals such as the PSS/SSS/CRS.

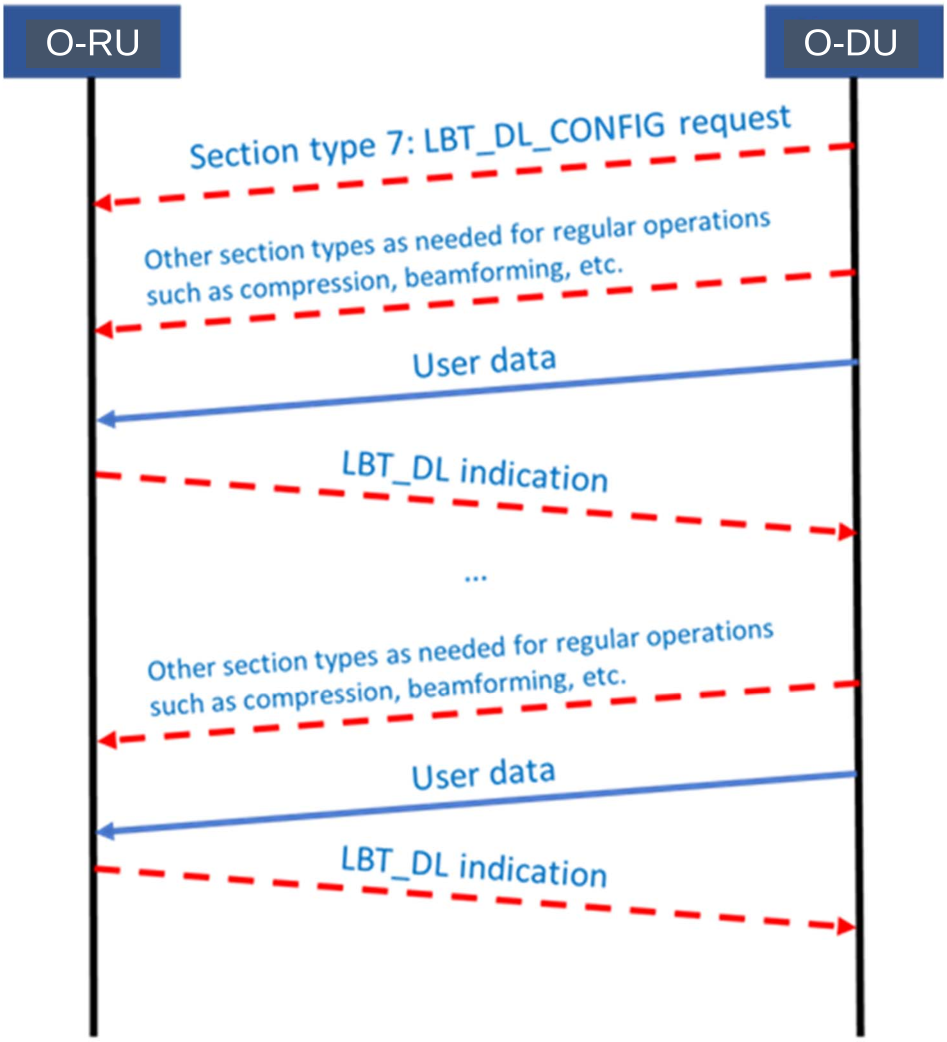

O-RU
O-DU
Section type 7: LBT_DL_CONFIG request
Other section types as needed for regular operations such as compression, beamforming, etc.
User data
LBT_DL indication
...
Other section types as needed for regular operations such as compression, beamforming, etc.
User data
LBT_DL indication

Figure 7-8 : LBT Message Flow

7.3.4.2 Definitions

- D_{OW}: Maximum one-way fronthaul latency
- P_{DU}: Maximum O-DU processing time (reading an upstream O-RU c-plane message, MAC processing, sending the downstream U-plane data)
- P_{RU}: Maximum O-RU processing time (sending an upstream c-plane message, receiving downstream U-plane data, transmitting U-plane data OTA)
- B_{RU}: O-RU Buffer depth (for LAA) ceiled to an integer number of OFDM symbols in microseconds (up to 140 symbols, for example (i.e., MCOT)). This is equivalent to the minimum amount of data that shall be buffered at the O-RU. This parameter could be less than or equal to the actual O-RU buffer size communicated via the M-Plane.
- Expired symbol: Symbol where the current time at the O-RU exceeds its target transmission time (i.e., SFN/SF) OTA.
- Normal (partially-filled) SF assumption: T_{start} is the start of the transmission time, which occurs at the first SF (slot) boundary the O-RU encounters after starting the LBT CAT 4 process and after finishing the minimum sensing duration. Initially known to the O-DU (since it sends the LBT start time and the LBT parameters to the O-RU). It can be updated in real-time based on the LBT outcome and the communication between the O-DU and O-RU. Note that here, a “slot” follows the LTE definition of containing seven symbols.
- LBT_DL indication Threshold (LBT_{th}): The latest time the O-DU is expected to receive the LBT_DL indication message from the O-RU.

$$LBT_{th} = T_{start} + (P_{RU} + D_{OW} + P_{DU})$$

- CU Transmission time threshold (CUT_{th}): The time at which the O-DU is required to send the data to the O-RU to be transmitted OTA at time T_{start}

$$CUT_{th} = T_{start} - (P_{RU} + D_{OW} + P_{DU})$$

- x: Minimum time between any two potential start transmission times

$$x = \begin{cases} 1\, ms, & Normal\, SF \\ 0.5\, ms, & Partially-filled\, SF \end{cases}$$

7.3.4.3 General Guidelines for the LAA-procedure

7.3.4.3.1 PDSCH Transmission

- O-DU should avoid buffer overflow or underflow at the O-RU:
  - O-DU should only send B_{RU} worth of data to the O-RU for every single transmission
  - O-DU should plan for the data to be received at the O-RU only P_{RU} before the actual OTA transmission time at the O-RU
- O-RU should avoid buffer overflow or underflow at the O-RU:
  - O-RU should flush its buffer (by dropping any expired symbols) as soon as any symbol becomes expired (i.e., current time is larger than the symbol's scheduled time)
  - O-RU should immediately send a success LBT indication to the O-DU once the channel is acquired.
  - The O-RU should send an error message to the O-DU if data received is larger than its local buffer
  - The O-RU should send a subframe drop or transmission message to the O-DU when buffered subframe is dropped because scheduled time is passed, or is transmitted after LBT success.
- The O-RU should have a buffer that satisfies the following equation: $B_{RU} \geq 2 \times D_{OW} + P_{DU} + P_{RU}$
- Note on LBT CAT 4 in general:
  - The O-DU can configure the O-RU (via the M-plane) with the threshold on the LBT CAT 4 duration (e.g., 8 ms). Once this threshold has exceeded, the O-RU sends a failure LBT indication to the O-DU, which in return sends back a new LBT config request. The O-RU can then restart the LBT CAT 4 process.
- The data signal may be received at the O-RU before or after the LBT_PDSCH_REQ
- The O-RU should generate the reservation signal locally whenever needed

7.3.4.3.2 DRS Transmission

- Recall:
  1. PSS/SSS being part of a DRS may occur outside subframe 0 and 5
  2. CRS/CSI-RS/PSS/SSS do not vary with subframe number but are kept unchanged across subframes 0–4 (call it DRS_v1) and 5–9 (DRS_v2).
- Assumption: DRS OTA transmission starts at the SF boundaries.
- The DRS signal may be received before or after the LBT_DRS_REQ
- LBT_DRS_REQ is sent once per DRS window (e.g., DMTC offset = 6 ms is conveyed to the O-RU via the M-plane).
- DRS signal is sent every SF in the 6 SFs of the DRS window until LBT succeeds.
- $DRS_{th,1} = SF0 - (P_{RU} + D_{OW} + P_{DU})$
- $DRS_{th,2} = SF5 - (P_{RU} + D_{OW} + P_{DU})$
- $SF0$: The start of SF0, SF1, SF2, SF3, or SF4
- $SF5$: The start of SF5, SF6, SF7, SF8, or SF9
- At $DRS_{th,1}$, O-DU sends DRS_v1 (i.e., to be transmitted within SFs 0-4)

- At $DRS_{th,2}$, O-DU sends DRS_v2 (i.e., to be transmitted within SFs 5-9)
- Notes:
    1. LBT indication with failure outcome shall be sent for every sensing period.

7.3.4.3.3 Congestion Window Information Transmission

- O-DU should send information regarding a congestion window adjustment to O-RU
    - HARQ feedback information for the reference subframe and number of TB are included
- O-RU should adjust its managed congestion window value and notify to O-DU the packet reception status
    - O-RU should adjust its congestion window value based on received information
    - O-RU should immediately send a success LBT_CWCONFIG_RSP to the O-DU when the message is successfully received. If not, O-RU should send a fail LBT_CWCONFIG_RSP.

7.3.5 Symbol Numbering and Duration

For the NR case, data symbol numbering in a slot shall be from zero to thirteen (for extended CP, eleven). For LTE the symbol numbering shall be mapped to the NR numbering as follows:

LTE with normal prefix maps to NR with same μ and normal prefix:

LTE symbol l_{LTE} of slot $2i$ is mapped to NR slot i, symbol $l' = l_{LTE}$
LTE symbol l_{LTE} of slot $2i + 1$ is mapped to NR slot i, symbol $l' = 7 + l_{LTE}$

LTE with extended prefix the following applies:

LTE symbol l_{LTE} of slot $2i$ is mapped to NR slot i, symbol $l' = l_{LTE}$
LTE symbol l_{LTE} of slot $2i + 1$ is mapped to NR slot i, symbol $l' = 6 + l_{LTE}$

The symbol duration can be determined from the SCS provided by “frameStructure” and knowledge of whether LTE versus NR is supported; 3GPP specifications 36.211 and 38.211 provide the necessary formulas.

When the SCS value provided by “frameStructure” is less than 15kHz (e.g. for long preamble PRACH formats), numerology for symbolId is based on 15kHz SCS.

7.3.6 Dynamic Spectrum Sharing (DSS)

Dynamic Spectrum sharing allows different technologies – 4G LTE and 5G NR in this version of the specification, to share same frequency carrier dynamically, i.e. the O-DU may allocate one or more set(s) of frequency-time resources (or PRBs) to LTE or NR dynamically. The O-RU also shall support this feature for the O-DU to take advantage of such an allocation. This version of the specification supports DSS via using dedicated eAxC ids (i.e. endpoints) for LTE and NR or via using Section Extension =9 for DSS.

7.3.6.1 Dynamic Spectrum Sharing (DSS) via Dedicated Endpoints

Dedicated endpoints are configured in the O-RU for LTE and NR (hence dedicated eAxC ids) to implement DSS. For e.g. when an O-DU allocates a set of PRBs for LTE, O-DU uses an eAxC id dedicated for LTE (assigned to a carrier identified as an LTE carrier) to address the LTE endpoint; at a different time when the O-DU allocates the same set of PRBs for NR, the O-DU uses eAxC id dedicated for NR (assigned to a carrier identified as an NR carrier) to address the NR endpoint in the O-RU.

When using this approach, a carrier will be assigned to LTE or NR, so when both LTE and NR are to be supported at the same time (“DSS”), two carriers with overlapping frequency ranges are to be identified, one being LTE and the other being NR. It will be up to the O-RU to overlap the two carriers such that the same frequency range can support both NR and LTE, and it will be up to the O-DU to assure that no RE is assigned to both LTE and NR at the same time.

7.3.6.2 Dynamic Spectrum Sharing (DSS) via Section Extension =9 for DSS

Using Section Extension =9 for DSS, the O-DU can use a single eAxC id (and hence a single endpoint) for conveying LTE and NR PRB allocations if the O-RU is capable of handling both LTE and NR processing using one endpoint (i.e.

the O-RU indicates support for Section Extension =9). Note that a single carrier may be allocated to either LTE, NR or both, and supporting DSS in this manner means the carrier (to which the endpoint is assigned) would be designated as supporting both. The O-DU indicates whether the PRB allocation information is applicable to LTE or to NR via Section Extension =9.

7.3.7 Channel Information based Beamforming

In channel information based beamforming method, the O-DU provides channel information per UE periodically (generally less often than every slot) using Section Type 6 C-plane message and then on a slot-by-slot basis the O-DU provides scheduling information using Section Type 5 C-plane message which the O-RU uses along with the channel information to calculate the proper beamforming weights for the specific slot with its co-scheduled UEs. To reduce peak bandwidth, the transmission/ reception window constraints shall not apply to Section Type 6 message. O-RU is expected to use the latest possible available channel information for a ueId for channel information based beamforming operation i.e., O-RU uses channel information for a ueId available in O-RU at the end of receive window of Section Type 5 message. O-RU is expected to store updated channel information for a ueId when it is received, to be used when that ueId is scheduled in future.

7.4 Elements for the C-plane Protocol

7.4.1 General

C-Plane messages are encapsulated using a two-layered header approach. The first layer consists of a eCPRI common header or IEEE 1914.3 common header, including corresponding fields used to indicate the message type, while the second layer is an application layer including necessary fields for control and synchronization. Within the application layer, a “section” defines the characteristics of U-plane data to be transferred or received from a beam with one pattern id. In general, the transmission header, application header, and sections are all intended to be aligned on 4-byte boundaries and are transmitted in “network byte order” meaning the most significant byte of a multi-byte parameter is transmitted first.

When the optional “little endian byte order” is chosen via M-plane, the beamforming weights (Section Type 1 or 3, and extType = 1) and ciIsample/ciQsample (Section Type 6) fields shall use little endian byte order to transmit the complex numbers. Annex D.2 shows little endian byte order format for various I/Q data length.

Table 7-3 describes the Section Types that are supported within the C-Plane.

Table 7-3 : Section Types

Section Type	Target Scenario	Remarks
0	Unused Resource Blocks or symbols in Downlink or Uplink	Indicates to O-RU that certain Resource Blocks or symbols will not be used (idle periods, guard periods). Likewise, there are no associated U-Plane messages containing IQ data for this Section Type. The purpose is to inform the O-RU that transmissions may be halted during the specified idle interval for e.g. power-savings or to provide an interval for calibration.
1	Most DL/UL radio channels*	Here “most” refers to channels not requiring time or frequency offsets such as are needed for mixed-numerology channels
2	reserved for future use	
3	PRACH and mixed-numerology channels*	Channels requiring time or frequency offsets or different-than-nominal SCS values
4	Reserved for future use	
5	UE scheduling information (UE-ID assignment to section)	Provides scheduling information for UE-IDs
6	Channel information	Sends UE-specific channel information from the O-DU to the O-RU
7	LAA	Messages communicated between O-DU and the O-RU in both directions to configure LBT for PDSCH/DRS transmission and to report the LBT outcome.
8-255	Reserved for future use	

*Note : When PRACH having same numerology as other UL channel, Section Type 1 can alternatively be used by O-DU for PRACH signaling. In this case, O-RU is not expected to perform any PRACH specific processing.

7.4.1.1 Section Extensions

Within a data section description, the section header in most cases includes an “extension flag” which indicates parameters that apply to the section beyond those within the section header (the extension flag is most often adjacent to the beamId or ueId fields). The presence of this extension flag indicates that following the header, a Section Extension is present. Within the Section Extension is another extension flag, adjacent to the “extType” field, which indicates that a second (or third, etc.) extension is present. In this way any number of Section Extensions may be included within a data section. This provides an extensibility for section parameters without the need to continually redefine the section header or create new Section Types to accommodate future fronthaul specification needs.

The Section Extension takes the form reminiscent of “TLV”: there is a type field specifically “extType” which is a 7-bit field describing the specific extension type, then a one-byte length field specifically “extLen” field detailing how many 4-byte words are contained in the extension (minimum of 1 including the “extType” and “extLen” fields), and then some number of parameters that depend on the “extType” value for their definition. In all cases the Section Extension will be an integer number of (4-byte) words in length.

Specific valid values of “extType”, their meanings and their associated parameters are detailed in subsequent sub-clauses of this specification.

7.4.1.2 Coupling of C-Plane and U-Plane

Selected types of C-Plane messages (Section Types 1, 3 and 5) carry data section descriptions that convey information applicable to data sections carried in U-Plane messages. This clause describes a method of coupling data section descriptions from C-Plane and data sections in U-Plane.

7.4.1.2.1 Coupling via sectionId Value

This method of C-Plane and U-Plane coupling is used by default. O-DU may configure O-RU to use the other method via M-Plane. In this method, information from section description D in C-Plane message CM is applied to resource element R in PRB P (PRB understood as 12 resource elements of one OFDM symbol consecutive in frequency) in data section S in U-Plane message UM if:

1. CM corresponds to UM i.e. both have the same eAxC_id and CM.**dataDirection** value matches direction of UM
2. CM.**frameId** = UM.**frameId** and CM.**subframeId** = UM.**subframeId** and CM.**slotId** = UM.**slotId**
3. symbol identified by UM.**symbolId** is described by the section description D.
   This requires evaluation of CM.**startSymbolId**, **symInc** and **numSymbol** in CM or checking D.**symbolMask** in Section Extension #6 , #12 or #19 if any of the extensions is present.
4. PRB P is described by the section description D. This requires checking that D.**startPrbc** ≤ P < D.**startPrbc** + D.**numPrbc** · (D.**rb** + 1), if D.**rb** = 1 checking that D.**startPrbc** modulo 2 = P modulo 2; it requires evaluating D.**rbgMask** in Section Extension #6 if this extension is present in D. Checking Section Extension #12, #13, #19 and #20 is required if any of them is present in D.
5. D.**reMask** has value 1 in a bit corresponding to RE R. Checking effect of Section Extension #20 is required if this extension is present in D.
6. D.**sectionId** = S.**sectionId**

In addition, the following restrictions apply (in context of one eAxC):

1. A PRB may occur only once in all U-Plane messages related to the same OFDM symbol.
2. A resource element may be “referenced” (see conditions 3, 4 and 5 above) by only one section description.
3. All section descriptions that are applicable to a PRB (see conditions 3 and 4 above) shall have same value of sectionId.
4. Section descriptions that are not applicable to same PRB (see conditions 3 and 4 above) shall have different value of sectionId.
5. Sender of C-Plane shall send in sectionId a value that is unique within a slot.
6. Section descriptions that have the same value of sectionId shall have same value of **rb**, **startPrbc**, **numPrbc** and **numSymbol**.

7.4.1.2.2 Coupling via Frequency and Time

This coupling method is used if a corresponding capability is reported by O-RU via M-Plane and use for an eAxC is configured via M-Plane. In this method, information from section description D in C-Plane message CM is applied to resource element R in PRB P (PRB understood as 12 resource elements of one OFDM symbol consecutive in frequency) in data section S in U-Plane message UM if:

1. CM corresponds to UM i.e. both have the same eAxC_id and CM.**dataDirection** value matches direction of UM
2. CM.**frameId** = UM.**frameId** and CM.**subframeId** = UM.**subframeId** and CM.**slotId** = UM.**slotId**
3. symbol identified by UM.**symbolId** is described by the section description D.
   This requires evaluation of CM.startSymbolId, symInc and numSymbol in CM or checking D.symbolMask in Section Extension #6, #12 or #19 if any of the extensions is present.
4. PRB P is described by the section description D. This requires checking that D.**startPrbc** ≤ P < D.**startPrbc** + D.**numPrbc** · (D.**rb** + 1), if D.**rb** = 1 checking that D.**startPrbc** modulo 2 = P modulo 2; it requires evaluating D.**rbgMask** in Section Extension #6 if this extension is present in D. Checking Section Extension #12, #13, #19 and #20 is required if any of them is present in D.
5. D.**reMask** has value 1 in a bit corresponding to RE R. Checking effect of Section Extension #20 is required if this extension is present in D.

In addition, the following restrictions apply (in context of one eAxC):

1. A PRB may occur only once in all U-Plane messages related to the same OFDM symbol.
2. A resource element may be “referenced” (see conditions 3, 4 and 5 above) by only one section description.
3. Sender of C-Plane shall send in sectionId a value 4095.
4. Sender of C-Plane shall order section descriptions such that any section descriptions referencing to any RE (non-zero value of reMask) are present before section descriptions that do not reference any RE (i.e. zero value in reMask). This allows O-RU to more quickly identify section descriptions that carry beamforming configuration for any RE and also makes them localized in memory which can improve message processing performance.

7.4.1.2.3 Coupling via Frequency and Time with Priorities

This optional coupling method is used if a corresponding capability is reported by O-RU via M-Plane and use for an eAxC is configured via M-plane. In this method information from section description D in C-Plane message CM is applied to resource element R in PRB P (PRB understood as 12 resource elements of one OFDM symbol consecutive in frequency) in data section S in U-Plane message UM if:

1. CM corresponds to UM i.e. both have the same eAxC_id and CM.**dataDirection** value matches direction of UM
2. CM.**frameId** = UM.**frameId** and CM.**subframeId** = UM.**subframeId** and CM.**slotId** = UM.**slotId**
3. symbol identified by UM.**symbolId** is described by the section description D.
   This requires evaluation of CM.startSymbolId, symInc and numSymbol in CM or checking D.symbolMask in Section Extension #6, #12 or #19 if any of the extensions is present.
4. PRB P is described by the section description D. This requires checking that D.**startPrbc** ≤ P < D.**startPrbc** + D.**numPrbc** · (D.**rb** + 1), if D.**rb** = 1 checking that D.**startPrbc** modulo 2 = P modulo 2; it requires evaluating D.**rbgMask** in Section Extension #6 if this extension is present in D. Checking Section Extension #12, #13, #19 and #20 is required if any of them is present in D.
5. D.**reMask** has value 1 in a bit corresponding to RE R and D has highest priority among data section descriptions referencing the RE R in the message CM. Checking effect of Section Extension #20 is required if this extension is present in D. Priority of data section description is the value of priority field in Section Extension #6 (see 5.4.7.6.4), #12 (see 5.4.7.6.4) or #19 (see 5.4.7.6.4) if any of the extensions is present in D and zero if the extensions are absent. NOTE: scope of search for highest priority description is restricted to one C-Plane message to avoid beamforming configuration errors that would be unavoidable if scope would be covering multiple messages and one of them would be lost. See also note in restriction #3 below.

In addition, the following restrictions apply (in context of one eAxC):

1. A PRB may occur only once in all U-Plane messages related to the same OFDM symbol.
2. Data section descriptions that refer to the same RE and are conveyed in the same C-Plane message shall have different priority in order to avoid ambiguity.
3. If a complete beamforming configuration does not fit into one C-Plane message, then sender shall duplicate highest priority data section descriptions, specifically each C-Plane message shall contain the highest priority section description referring to any RE that is referred in a message. This is required to ensure that O-RU will interpret configuration received in each message correctly. If such duplication is needed to avoid misinterpretation it does not violate restriction #4 below. Alternatively, Section Extension #20 can be used instead of repeating high priority sections (refer to clause 7.4.7.20, and 5.5.10 for more details)
4. Sender shall avoid sending unnecessary data section descriptions e.g. descriptions that carry the same configuration (but refer to note in restriction #3 above).
5. Sender of C-Plane shall send in sectionId a value 4095.
6. Sender of C-Plane shall order section descriptions such that any section descriptions referring to any RE (non-zero value of reMask) are present before section descriptions that do not reference any RE (i.e. zero value in reMask); section descriptions referring to any RE shall be ordered by effective priority (highest priority first). Ordering section descriptions with non-zero reMask first allows O-RU to more quickly identify section descriptions that carry beamforming configuration for any RE and also makes them localized in memory which can improve message processing performance. Ordering section descriptions by highest priority first is intended to optimize O-RU C-plane message processing i.e. RE beamforming configuration can be determined from the earliest section description occurring in the message.

7.4.1.2.4 Coupling via Frequency and Time with Priorities (Optimized)

All the restrictions remain same as coupling via frequency and time with priorities except restriction 5:

5. Sender of C-Plane shall send in sectionId value 4095 for section description with lower priority. For highest priority sections unique sectionIds shall be set in a defined range starting from “0” to “max-highest-priority-sections-per-slot” specified through M-Plane, across eAxC IDs (limiting this range helps O-RU reduce the search space for repeated highest priority section descriptions detection). While duplicating highest priority sections, all fields including sectionId shall be duplicated. In addition to this, for C-Plane message processing O-RUs, additional limits to restrict the number of highest priority sections per C-Plane message on top of eAxC limits shall be applied (clause 7.6.2).

In addition to associating unique sectionId to highest priority section as described above, O-DU shall set the “repetition” flag to “1” for every repeated C-Plane message. O-RU can choose to process or ignore the flag based on its implementation (clause 7.4.7.6.5).

7.4.2 Scheduling and Beamforming Commands

A common frame format is used for C-Plane messages, consisting of a transport layer and an application layer. The application layer is within the transport layer payload and consists of a common header for time reference, followed by information and parameters dependent and specific to the Section Type in use. Multiple sets of section data of the same Section Type value can be lined up one after another within the payload. To minimize packet rate over the interface, transmitter should fill messages with as many subsequent sections (with or without sequential section IDs) as possible. However, sets of section data of different Section Type values are to be sent via separate messages (i.e. different values of Section Type shall not be mixed within a single C-Plane message payload).

- Transport Layer (see clause 5.1.3)
- Application Layer
    - Section Type “0” Fields (used for indicating idle or guard periods from O-DU to O-RU)
        - <u>Common Header Fields</u>
            - **dataDirection** (data direction (gNB Tx/Rx)) field: 1 bit

- **payloadVersion** (payload version) field: 3 bits
  - value = “1” shall be set (1st protocol version for payload and time reference format)
- **filterIndex** (filter index) field: 4 bits
- **frameId** (frame identifier) field: 8 bits
- **subframeId** (subframe identifier) field: 4 bits
- **slotID** (slot identifier) field: 6 bits
- **startSymbolid** (start symbol id) field: 6 bits
- **numberOfsections** (number of sections) field: 8 bits
- **sectionType** (Section Type) field: 8 bits
  - value = “0” shall be set
- **timeOffset** (time offset) field: 16 bits
- **frameStructure** (frame structure) field: 8 bits
- **cpLength** (cyclic prefix length) field: 16 bits
- **reserved** (reserved for future use) field: 8 bits

- Section Fields
  - **sectionID** (section identifier) field: 12 bits
  - **rb** (resource block indicator) field: 1 bit
  - **symInc** (symbol number increment command) field: 1 bit
  - **startPrbc** (starting PRB of data section description) field: 10 bits
  - **numPrbc** (number of contiguous PRBs per data section description) field: 8 bits
  - **reMask** (resource element mask) field: 12 bits
  - **numSymbol** (number of symbols) field: 4 bits
  - **ef** (extension flag) field: 1 bit
  - **reserved** (reserved for future use) field: 15 bits

- Section Type “1” Fields (used for most Downlink and Uplink radio channels – some channels especially PRACH and mixed-numerology channels may need more information elements contained in other Section Types. However, if Section Type “1” is used for PRACH channel, it should be processed in the same manner as a non-PRACH channel is processed in O-RU.)
  - Common Header Fields
    - **dataDirection** (data direction (gNB Tx/Rx)) field: 1 bit
    - **payloadVersion** (payload version) field: 3 bits
      - value = “1” shall be set (1st protocol version for payload and time reference format)
    - **filterIndex** (filter index) field: 4 bits
    - **frameId** (frame identifier) field: 8 bits
    - **subframeId** (subframe identifier) field: 4 bits
    - **slotID** (slot identifier) field: 6 bits
    - **startSymbolid** (start symbol id) field: 6 bits
    - **numberOfsections** (number of sections) field: 8 bits
    - **sectionType** (Section Type) field: 8 bits
      - value = “1” shall be set
    - **udCompHdr** (user data compression header) field: 8 bits

        - **reserved** (reserved for future use) field: 8 bits
    - Section Fields
        - **sectionId** (section identifier) field: 12 bits
        - **rb** (resource block identifier) field: 1 bit
        - **symInc** (symbol number increment command) field: 1 bit
        - **startPrbc** (starting PRB of data section description) field: 10 bits
        - **numPrbc** (number of contiguous PRBs per data section description) field: 8 bits
        - **reMask** (resource element mask) field: 12 bits
        - **numSymbol** (number of symbols) field: 4 bits
        - **ef** (extension flag) field: 1 bit
        - **beamId** (beam identifier) field: 15 bits

- Section Type “3” Fields (used for PRACH and mixed-numerology channels):
    - Common Header Fields
        - **dataDirection** (data direction (gNB Tx/Rx)) field: 1 bit
        - **payloadVersion** (payload version) field: 3 bits
            - value = “1” shall be set (1st protocol version for payload and time reference format)
        - **filterIndex** (filter index) field: 4 bits
        - **frameId** (frame identifier) field: 8 bits
        - **subframeId** (subframe identifier) field: 4 bits
        - **slotID** (slot identifier) field: 6 bits
        - **startSymbolid** (start symbol identifier) field: 6 bits
        - **numberOfsections** (number of sections) field: 8 bits
        - **sectionType** (Section Type) field: 8 bits
            - value = “3” shall be set
        - **timeOffset** (time offset) field: 16 bits
        - **frameStructure** (frame structure) field: 8 bits
        - **cpLength** (cyclic prefix length) field: 16 bits
        - **udCompHdr** (user data compression header) field: 8 bits
    - Section Fields
        - **sectionID** (section identifier) field: 12 bits
        - **rb** (resource block identifier) field: 1 bit
        - **symInc** (symbol number increment command) field: 1 bit
        - **startPrbc** (starting PRB of data section description) field: 10 bits
        - **numPrbc** (number of contiguous PRBs per data section description) field: 8 bits
        - **reMask** (resource element mask) field: 12 bits
        - **numSymbol** (number of symbols) field: 4 bits
        - **ef** (extension flag) field: 1 bit
        - **beamId** (beam identifier) field: 15 bits
        - **freqOffset** (frequency offset) field: 24 bits
        - **reserved** (reserved for future use) field: 8 bits

- Section Type “5” Fields (used for UE scheduling information):
    - Common Header Fields
        - **dataDirection** (data direction (gNB Tx/Rx)) field: 1 bit
        - **payloadVersion** (payload version) field: 3 bits
            - value = “1” shall be set (1st protocol version for payload and time reference format)
        - **filterIndex** (filter index) field: 4 bits
        - **frameId** (frame identifier) field: 8 bits
        - **subframeId** (subframe identifier) field: 4 bits
        - **slotID** (slot identifier) field: 6 bits
        - **startSymbolid** (start symbol identifier) field: 6 bits
        - **numberOfsections** (number of sections) field: 8 bits
        - **sectionType** (Section Type) field: 8 bits
            - value = “5” shall be set
        - **udCompHdr** (user data compression header) field: 8 bits
        - **reserved** (reserved for future use) field: 8 bits
    - Section Fields
        - **sectionID** (section identifier) field: 12 bits
        - **rb** (resource block identifier) field: 1 bit
        - **symInc** (symbol number increment command) field: 1 bit
        - **startPrbc** (starting PRB of data section description) field: 10 bits
        - **numPrbc** (number of contiguous PRBs per data section description) field: 8 bits
        - **reMask** (resource element mask) field: 12 bits
        - **numSymbol** (number of symbols) field: 4 bits
        - **ef** (extension flag) field: 1 bit
        - **ueId** (UE identifier) field: 15 bits
- Section Type “6” Fields (used for sending channel information for a specific UE ID):
    - Common Header Fields
        - **dataDirection** (data direction (gNB Tx/Rx)) field: 1 bit
        - **payloadVersion** (payload version) field: 3 bits
            - value = “1” shall be set (1st protocol version for payload and time reference format)
        - **filterIndex** (filter index) field: 4 bits
        - **frameId** (frame identifier) field: 8 bits
        - **subframeId** (subframe identifier) field: 4 bits
        - **slotID** (slot identifier) field: 6 bits
        - **startSymbolid** (start symbol identifier) field: 6 bits
        - **numberOfsections** (number of sections) field: 8 bits
        - **sectionType** (Section Type) field: 8 bits
            - value = “6” shall be set
        - **numberOfUEs** (number of UE-specific channel information data sets) field: 8 bits
        - **reserved** (reserved for future use) field: 8 bits

  - Section Fields
    - **ef** (extension flag) field: 1 bit
    - **ueId** (UE identifier) field: 15 bits
    - **regularizationFactor** (regularization factor used for MMSE reception) field: 16 bits
    - **reserved** (reserved for future use) field: 4 bits
    - **rb** (resource block identifier) field: 1 bit
    - **symInc** (symbol number increment command) field: 1 bit
    - **startPrbc** (starting PRB of data section description) field: 10 bits
    - **numPrbc** (number of contiguous PRBs per data section description) field: 8 bits
    - **ciIsample** (channel information value, in-phase sample) field: 16 bits
    - **ciQsample** (channel information value, quadrature sample) field: 16 bits

- Section Type “7” Fields (used to support LAA):
  - Common Header Fields
    - **reserved** (reserved for future use) field: 1 bit
    - **payloadVersion** (payload version) field: 3 bits
      - value = “1” shall be set (1st protocol version for payload and time reference format)
    - **reserved** (reserved for future use) field: 4 bits
    - **frameId** (frame identifier) field: 8 bits
    - **subframeId** (subframe identifier) field: 4 bits
    - **slotID** (slot identifier) field: 6 bits
    - **reserved** (reserved for future use) field: 14 bits
    - **sectionType** (Section Type) field: 8 bits
      - value = “7” shall be set
    - **reserved** (reserved for future use) field: 16 bits
    - Section Fields
    - **laaMsgType** (LAA message type) field: 4 bits
    - **laaMsgLen** field: 4 bits

- laaMsgType = “0” shall be set for LBT_DL_CONFIG.request: LBT_PDSCH_REQ
    - **lbtHandle** (An opaque handling returned in LBT_PDSCH_RSP) field: 16 bits
    - **lbtOffset** (LBT start time in microseconds from the beginning of the subframe scheduled by this message) field: 10 bits
    - **lbtMode** (LBT process type) field: 2 bits
    - **reserved** (reserved for future use) field: 1 bit
    - **lbtDeferFactor** (Defer factor in sensing slots as described in 3GPP TS 36.213 [14] clause 15.1.1) field: 3 bits
    - **lbtBackoffCounter** (LBT backoff counter in sensing slots as described in 3GPP TS 36.213 [14] clause 15.1.1) field: 10 bits
    - **MCOT** (LTE TXOP duration in subframes) field: 4 bits
    - reserved (reserved for future use) field: 10 bits

- laaMsgType = “1” shall be set for LBT_DL_CONFIG.request: LBT_DRS_REQ

    - **lbtHandle** (An opaque handling returned in LBT_DRS_RSP) field: 16 bits
    - **lbtOffset** (LBT start time in microseconds from the beginning of the subframe scheduled by this message) field: 10 bits
    - **lbtMode** (LBT process type) field: 2 bits:
    - **reserved** (reserved for future use) field: 28 bit

- laaMsgType = "2" shall be set for LBT_DL.indication: LBT_PDSCH_RSP
    - **lbtHandle** (An opaque handling returned in LBT_PDSCH_RSP) field: 16 bits
    - **lbtPdschRes** (LBT result of SFN/SF) field: 2 bits
    - **inParSF** (Indicates whether the initial SF in the LBT process is full or partial) field: 1 bit
    - **sfStatus** (subfrme status) field: 1 bit
    - **sfnSf** (SFN/SF of subframe which is dropped or successfully transmitted at O-RU) field: 12 bits
    - **reserved** (reserved for future use) field: 24 bits

- laaMsgType = "3" shall be set for LBT_DL.indication: LBT_DRS_RSP
    - **lbtHandle** (An opaque handling returned in LBT_DRS_RSP) field: 16 bits
    - **lbtDrsRes** (LBT result of SFN/SF) field: 1 bit
    - **reserved** (reserved for future use) field: 7 bits

- laaMsgType = "4" shall be set for LBT_buffer error: LBT_Buffer_Error
    - **lbtHandle** (An opaque handling returned in LBT_DRS_RSP) field: 16 bits
    - **lbtBufErr** (LBT buffer error type) field: 1 bit
    - **reserved** (reserved for future use) field: 7 bits

- laaMsgType = "5" shall be set for LBT_DL_CONFIG.request: LBT_CWCONFIG_REQ
    - **lbtHandle** (An opaque handle returned in LBT_CWCONFIG_RSP) field: 16 bits
    - **lbtCWConfig_H** (HARQ feedback information regarding number of NACK about reference subframe as described in 3GPP TS 36.213 [14] clause 15.1.3) field: 8 bits
    - **lbtCWConfig_T** (number of TB to manage Congestion Window as described in 3GPP TS 36.213 [14] clause 15.1.3) field: 8 bits
    - **lbtMode** (LBT process type) field: 2 bits
    - **lbtTrafficClass** (LBT traffic class) field: 3 bits
    - **reserved** (reserved for future use) field: 19 bits

- laaMsgType = "6" shall be set for LBT_DL.indication: LBT_CWCONFIG_RSP
    - **lbtHandle** (An opaque handling returned in LBT_CWCONFIG_RSP) field: 16 bits
    - **lbtCWR_Rst** (Notification LBT_CWCONFIG_REQ message successful or not) field: 1 bit
    - **reserved** (reserved for future use) field: 7 bits

Table 7-4 : Scheduling and beamforming commands frame format (Section Type "0")

Section Type 0 : idle / guard periods									
0 (msb)	1	2	3	4	5	6	7 (lsb)	# of bytes	

<table>
<tr><td colspan="8">transport header, see clause 5.1.3</td><td>8</td><td>Octet 1</td></tr>
<tr><td>dataDirection</td><td colspan="3">payloadVersion</td><td colspan="4">filterIndex</td><td>1</td><td>Octet 9</td></tr>
<tr><td colspan="8">frameId</td><td>1</td><td>Octet 10</td></tr>
<tr><td colspan="4">subframeId</td><td colspan="4">slotId</td><td>1</td><td>Octet 11</td></tr>
<tr><td colspan="2">slotId</td><td colspan="6">startSymbolid</td><td>1</td><td>Octet 12</td></tr>
<tr><td colspan="8">numberOfsections</td><td>1</td><td>Octet 13</td></tr>
<tr><td colspan="8">sectionType = 0</td><td>1</td><td>Octet 14</td></tr>
<tr><td colspan="8">timeOffset</td><td>2</td><td>Octet 15</td></tr>
<tr><td colspan="8">frameStructure</td><td>1</td><td>Octet 17</td></tr>
<tr><td colspan="8">cpLength</td><td>2</td><td>Octet 18</td></tr>
<tr><td colspan="8">Reserved</td><td>1</td><td>Octet 20</td></tr>
<tr><td colspan="8">sectionId</td><td>1</td><td>Octet 21</td></tr>
<tr><td colspan="4">sectionId</td><td>rb</td><td>symInc</td><td colspan="2">startPrbc</td><td>1</td><td>Octet 22</td></tr>
<tr><td colspan="8">startPrbc</td><td>1</td><td>Octet 23</td></tr>
<tr><td colspan="8">numPrbc</td><td>1</td><td>Octet 24</td></tr>
<tr><td colspan="8">reMask[11:4]</td><td>1</td><td>Octet 25</td></tr>
<tr><td colspan="4">reMask[3:0]</td><td colspan="4">numSymbol</td><td>1</td><td>Octet 26</td></tr>
<tr><td>ef</td><td colspan="7">reserved (7 bits)</td><td>1</td><td>Octet 27</td></tr>
<tr><td colspan="8">reserved (8 bits)</td><td>1</td><td>Octet 28</td></tr>
<tr><td colspan="8">Section Extensions as indicated by”ef” if any</td><td>var</td><td>Octet 29</td></tr>
<tr><td colspan="8">…</td><td></td><td></td></tr>
<tr><td colspan="8">sectionId</td><td>1</td><td>Octet N</td></tr>
<tr><td colspan="4">sectionId</td><td>rb</td><td>symInc</td><td colspan="2">startPrbc</td><td>1</td><td>N+1</td></tr>
<tr><td colspan="8">startPrbc</td><td>1</td><td>N+2</td></tr>
<tr><td colspan="8">numPrbc</td><td>1</td><td>N+3</td></tr>
<tr><td colspan="8">reMask[11:4]</td><td>1</td><td>N+4</td></tr>
<tr><td colspan="4">reMask[3:0]</td><td colspan="4">numSymbol</td><td>1</td><td>N+5</td></tr>
<tr><td>ef</td><td colspan="7">reserved (7 bits)</td><td>1</td><td>N+6</td></tr>
<tr><td colspan="8">reserved (8 bits)</td><td>1</td><td>N+7</td></tr>
<tr><td colspan="8">Section Extensions as indicated by”ef” if any</td><td>var</td><td>N+8</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>Octet M</td></tr>
</table>

shading: yellow is transport header, pink is radio application header, others are repeated sections

Table 7-5 : Scheduling and beamforming commands frame format (Section Type “1”)

<table>
<tr><th colspan="10">Section Type 1 : DL/UL control msgs</th></tr>
<tr><th>0 (msb)</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7 (lsb)</th><th># of bytes</th><th></th></tr>
<tr><td colspan="8">transport header, see clause 5.1.3</td><td>8</td><td>Octet 1</td></tr>
<tr><td>dataDirection</td><td colspan="3">payloadVersion</td><td colspan="4">filterIndex</td><td>1</td><td>Octet 9</td></tr>
<tr><td colspan="8">frameId</td><td>1</td><td>Octet 10</td></tr>
<tr><td colspan="4">subframeId</td><td colspan="4">slotId</td><td>1</td><td>Octet 11</td></tr>
<tr><td colspan="2">slotId</td><td colspan="6">startSymbolid</td><td>1</td><td>Octet 12</td></tr>
<tr><td colspan="8">numberOfsections</td><td>1</td><td>Octet 13</td></tr>
<tr><td colspan="8">sectionType = 1</td><td>1</td><td>Octet 14</td></tr>
<tr><td colspan="8">udCompHdr</td><td>1</td><td>Octet 15</td></tr>
<tr><td colspan="8">reserved</td><td>1</td><td>Octet 16</td></tr>
<tr><td colspan="8">sectionId</td><td>1</td><td>Octet 17</td></tr>
<tr><td colspan="4">sectionId</td><td>rb</td><td>symInc</td><td colspan="2">startPrbc</td><td>1</td><td>Octet 18</td></tr>
<tr><td colspan="8">startPrbc</td><td>1</td><td>Octet 19</td></tr>
<tr><td colspan="8">numPrbc</td><td>1</td><td>Octet 20</td></tr>
<tr><td colspan="8">reMask[11:4]</td><td>1</td><td>Octet 21</td></tr>
</table>

reMask[3:0]				numSymbol				1	Octet 22
ef = 1	beamId[14:8]							1	Octet 23
beamId[7:0]								1	Octet 24
Section Extensions as indicated by "ef"								var	Octet 25
…									
sectionId								1	Octet N
sectionId				rb	symInc	startPrbc		1	N+1
startPrbc								1	N+2
numPrbc								1	N+3
reMask[11:4]								1	N+4
reMask[3:0]				numSymbol				1	N+5
ef = 0	beamId[14:8]							1	N+6
beamId[7:0]								1	N+7
Section Extensions as indicated by "ef"								var	N+8
									Octet M

shading: yellow is transport header, pink is radio application header, others are repeated sections

Table 7-6 : Scheduling and beamforming commands frame format (Section Type "3")

Section Type 3 : PRACH & mixed-numerology									
0 (msb)	1	2	3	4	5	6	7 (lsb)	# of bytes	
transport header, see clause 5.1.3								8	Octet 1
dataDirection	payloadVersion			filterIndex				1	Octet 9
frameId								1	Octet 10
subframeId				slotId				1	Octet 11
slotId		startSymbolid						1	Octet 12
numberOfsections								1	Octet 13
sectionType = 3								1	Octet 14
timeOffset								2	Octet 15
frameStructure								1	Octet 17
cpLength								2	Octet 18
udCompHdr								1	Octet 20
sectionId								1	Octet 21
sectionId				rb	symInc	startPrbc		1	Octet 22
startPrbc								1	Octet 23
numPrbc								1	Octet 24
reMask[11:4]								1	Octet 25
reMask[3:0]				numSymbol				1	Octet 26
ef	beamId[14:8]							1	Octet 27
beamId[7:0]								1	Octet 28
freqOffset								3	Octet 29
reserved (8 bits)								1	Octet 32
Section Extensions as indicated by "ef"								var	Octet 33
…									
sectionId								1	Octet N
sectionId				rb	symInc	startPrbc		1	N+1
startPrbc								1	N+2
numPrbc								1	N+3
reMask[11:4]								1	N+4
reMask[3:0]				numSymbol				1	N+5
ef	beamId[14:8]							1	N+6

beamId[7:0]								1	N+7
freqOffset								3	N+8
reserved (8 bits)								1	N+11
Section Extensions as indicated by “ef”								var	N+12
									Octet M

shading: yellow is transport header, pink is radio application header, others are repeated sections

Table 7-7 : UE scheduling information frame format (Section Type “5”)

Section Type 5 : UE scheduling information conveyance									
0 (msb)	1	2	3	4	5	6	7 (lsb)	# of bytes	
transport header, see clause 5.1.3								8	Octet 1
dataDirection	payloadVersion			filterIndex				1	Octet 9
frameId								1	Octet 10
subframeId				slotId				1	Octet 11
slotId		startSymbolid						1	Octet 12
numberOfsections								1	Octet 13
sectionType = 5								1	Octet 14
udCompHdr								1	Octet 15
reserved								1	Octet 16
sectionId								1	Octet 17
sectionId				rb	symInc	startPrbc		1	Octet 18
startPrbc								1	Octet 19
numPrbc								1	Octet 20
reMask[11:4]								1	Octet 21
reMask[3:0]				numSymbol				1	Octet 22
ef	ueId[14:8]							1	Octet 23
ueId[7:0]								1	Octet 24
Section Extensions as indicated by “ef”								var	Octet 25
…									
sectionId								1	Octet N
sectionId				rb	symInc	startPrbc		1	N+1
startPrbc								1	N+2
numPrbc								1	N+3
reMask[11:4]								1	N+4
reMask[3:0]				numSymbol				1	N+5
ef	ueId[14:8]							1	N+6
ueId[7:0]								1	N+7
Section Extensions as indicated by “ef”								var	N+8
									Octet M

shading: yellow is transport header, pink is radio application header, others are repeated sections

Table 7-8 : UE channel information frame format (Section Type “6”)

Section Type 6 : channel information conveyance									
0 (msb)	1	2	3	4	5	6	7 (lsb)	# of bytes	
transport header, see clause 5.1.3								8	Octet 1
dataDirection	payloadVersion			filterIndex				1	Octet 9
frameId								1	Octet 10
subframeId				slotId				1	Octet 11

slotId		startSymbolid						1	Octet 12
numberOfsections								1	Octet 13
sectionType = 6								1	Octet 14
numberOfUEs								1	Octet 15
reserved								1	Octet 16
ef	ueId[14:8]							1	Octet 17
ueId[7:0]								1	Octet 18
regularizationFactor								2	Octet 19
reserved				rb	symInc	startPrbc		1	Octet 21
startPrbc								1	Octet 22
numPrbc								1	Octet 23
ciIsample (first PRB, first antenna)								var	Octet 24
ciQsample (first PRB, first antenna)								var	
ciIsample (first PRB, second antenna)								var	
ciQsample (first PRB, second antenna)								var	
…									
ciIsample (first PRB, last antenna)								var	
ciQsample (first PRB, last antenna)								var	
…									
ciIsample (last PRB, last antenna)								var	
ciQsample (last PRB, last antenna)								var	
Section Extensions as indicated by “ef”								var	
…									
ef	ueId[14:8]							1	Octet N
ueId[7:0]								1	N+1
regularizationFactor								2	N+2
Reserved				rb	symInc	startPrbc		1	N+4
startPrbc								1	N+5
numPrbc								1	N+6
ciIsample (first PRB, first antenna)								var	N+7
ciQsample (first PRB, first antenna)								var	
ciIsample (first PRB, second antenna)								var	
ciQsample (first PRB, second antenna)								var	
…									
ciIsample (first PRB, last antenna)								var	
ciQsample (first PRB, last antenna)								var	
…									
ciIsample (last PRB, last antenna)								var	
ciQsample (last PRB, last antenna)								var	
Section Extensions as indicated by “ef”								var	
									Octet M

shading: yellow is transport header, pink is radio application header, others are repeated sections

Table 7-9 : LAA Message, O-DU to O-RU -OR- O-RU to O-DU (Section Type “7”)

Section Type 7 : LAA Message,O-DU to O-RU or O-RU toO-DU									
0 (msb)	1	2	3	4	5	6	7 (lsb)	# of bytes	
transport header, see clause 5.1.3								8	Octet 1
reserved	payloadVersion			reserved				1	Octet 9

frameId								1	Octet 10
subframeId				slotId				1	Octet 11
slotId		reserved						1	Octet 12
reserved								1	Octet 13
sectionType = 7								1	Octet 14
reserved								1	Octet 15
reserved								1	Octet 16
laaMsgType				laaMsgLen				1	Octet 17
Payload (see below) plus padding to 32-bit boundary								3 or 7	Octet 18+

Note that here, frameId, subframeId and slotId serve to provide a time stamp on the LAA LBT C-Plane message and do not describe any U-Plane data associated with the Section Type 7 C-Plane message.

Section Type 7 : LAA Request Message, O-DU to O-RU **LBT_PDSCH_REQ**									
0 (msb)	1	2	3	4	5	6	7 (lsb)	# of bytes	
laaMsgType = 0000b (LBT_PDSCH_REQ)				laaMsgLen = 2 (2 words)				1	Octet 17
lbtHandle								2	Octet 18
lbtOffset[9:2]								1	Octet 20
lbtOffset[1:0]		lbtMode		reserved	lbtDeferFactor			1	Octet 21
lbtBckoffCounter[9:2]								1	Octet 22
lbtBckoffCounter[1:0]		MCOT				reserved		1	Octet 23
reserved								1	Octet 24

Section Type 7 : LAA Request Message, O-DU to O-RU **LBT_DRS_REQ**									
0 (msb)	1	2	3	4	5	6	7 (lsb)	# of bytes	
laaMsgType = 0001b (LBT_DRS_REQ)				laaMsgLen = 2 (2 words)				1	Octet 17
lbtHandle								2	Octet 18
lbtOffset[9:2]								1	Octet 20
lbtOffset[1:0]		lbtMode		reserved				1	Octet 21
reserved								1	Octet 22
reserved								1	Octet 23
reserved								1	Octet 24

Section Type 7 : LAA Response Message, O-RU to O-DU **LBT_PDSCH_RSP**									
0 (msb)	1	2	3	4	5	6	7 (lsb)	# of bytes	
laaMsgType = 0010b (LBT_PDSCH_RSP)				laaMsgLen = 2 (2 words)				1	Octet 17
lbtHandle								2	Octet 18
lbtPdschRes		inParSF	sfStatus	sfnSf[11:8]				1	Octet 20
sfnSf[7:0]								1	Octet 21
reserved								1	Octet 22
reserved								1	Octet 23
reserved								1	Octet 24

Section Type 7 : LAA Response Message, O-RU to O-DU **LBT_DRS_RSP**									
0 (msb)	1	2	3	4	5	6	7 (lsb)	# of bytes	
laaMsgType = 0011b (LBT_DRS_RSP)				laaMsgLen = 1 (1 word)				1	Octet N
lbtHandle								2	N+1
lbtDrsRes	reserved							1	N+3

Section Type 7 : LAA Response Message, O-RU to O-DU **LBT_Buffer_Error**									
0 (msb)	1	2	3	4	5	6	7 (lsb)	# of bytes	
laaMsgType = 0100b (LBT_Buffer_Error)				laaMsgLen = 1 (1 word)				1	Octet N
lbtHandle								2	N+1
lbtBufErr	reserved							1	N+3

Section Type 7 : LAA Request Message, O-DU to O-RU **LBT_CWCONFIG_REQ**									
0 (msb)	1	2	3	4	5	6	7 (lsb)	# of bytes	
laaMsgType = 0101b (LBT_CWCONFIG_REQ)				laaMsgLen = 2 (2 words)				1	Octet N
lbtHandle								2	N+1
lbtCWConfig_H								1	N+3
lbtCWConfig_T								1	N+4
lbtMode		lbtTrafficClass			reserved			1	N+5
reserved								2	N+6

Section Type 7 : LAA Request Message, O-RU to O-DU L**BT_CWCONFIG_RSP**									
0 (msb)	1	2	3	4	5	6	7 (lsb)	# of bytes	
laaMsgType = 0110b (LBT_CWCONFIG_RSP)				laaMsgLen = 1 (1 word)				1	Octet N
lbtHandle								2	N+1
lbtCWR_Rst	reserved							1	N+2

7.4.3 Coding of Information Elements – Transport Layer

See clause 5.1.3 for transport header information element details.

7.4.4 Coding of Information Elements – Application Layer, Common

7.4.4.1 dataDirection (data direction (gNB Tx/Rx))

Description: This parameter indicates the gNB data direction.

Value range: {0b=Rx (i.e. UL), 1b=Tx (i.e. DL)}.

Type: binary bit.

Field length: 1 bit.

7.4.4.2 payloadVersion (payload version)

Description: This parameter defines the payload protocol version valid for the following IEs in the application layer. In this version of the specification payloadVersion=001b shall be used.

Value range: {000b-111b=Payload version}.

Type: unsigned integer.

Field length: 3 bits.

Default Value: 001b (version 1 assumed).

7.4.4.3 filterIndex (filter index)

Description: This parameter defines an index to the channel filter to be used between IQ data and air interface, both in DL and UL. For most physical channels filterIndex =0000b is used which indexes the standard channel filter, e.g. 100MHz channel filter for 100MHz nominal carrier bandwidth. Another use case is PRACH in UL, where different filter indices can be used for different PRACH formats, assuming that before FFT processing of PRACH data there is a separate PRACH filter or PRACH filter in addition to the standard channel filter in UL. Please note that for PRACH, when Section Type “3” message is used, there is typically also a frequency offset (see freqOffset) applied before the PRACH filter.

NOTE 1: Filter index is commanded from O-DU to O-RU. Likewise, it is not mandatory to command special filters, and filter index = 0000b is also allowed for PRACH.

NOTE 2: When using filter indices corresponding to PRACH, the first RE of the first PRB addressed inside the section Id shall correspond with the first guard tone used at the lower edge of the PRACH frequency block as specified in 3GPP RAN1 TS 3x.211. In case of fragmentation of 1 data section over multiple U-Plane messages, this is applicable to the first fragmented section.

NOTE 3: Since different PRACH formats are assigned to the same filterIndex value, optionally O-RU can notify by M-plane the specific supported formats (grouped into PRACH format-groups) in o-ran-uplane-conf.yang module on a per-endpoint basis.

Value range: {0000b-1111b}.

Table 7-10 : Filter Index

Value of IE “filter Index”	Usage	PRACH preamble formats	Minimum filter pass band
0000b=0x0	standard channel filter	N/A	
0001b=0x1	UL filter for PRACH preamble formats	LTE-0, LTE-1, LTE-2, LTE-3, NR-0, NR-1, NR-2	839 x 1.25kHz = 1048.75 kHz
0010b=0x2		NR-3	839 x 5 kHz = 4195 kHz
0011b=0x3		NR-A1, NR-A2, NR-A3, NR-B1, NR-B2, NR-B3, NR-B4, NR-C0, NR-C2	139 x Δf^{RA} (See SCS in **Table 5-11**)
0110b=0x6			1151 x 15 kHz = 17265 kHz
0111b=0x7			571 x 30 kHz = 17130 kHz
0100b=0x4	UL filter for NPRACH	LTE-NB0, LTE-NB1 LTE-NB0-a, LTE-NB1-a LTE-NB2	48 x 3.75kHz = 180 kHz 144 x 1.25 kHz = 180 kHz
0101b=0x5	UL filter for PRACH preamble formats	LTE-4	139 x 7.5kHz = 1042.5 kHz
1000b…1111b	Reserved		

Type: unsigned integer.

Field length: 4 bits.

Default Value: 0000b (no special filter).

7.4.4.4 frameId (frame identifier)

Description: This parameter is a counter for 10 ms frames (wrapping period 2.56 seconds), specifically frameId = frame number modulo 256.

Value range: {0000 0000b-1111 1111b}.

Type: unsigned integer.

Field length: 8 bits.

7.4.4.5 subframeId (subframe identifier)

Description: This parameter is a counter for 1 ms sub-frames within 10ms frame.

Value range: {0000b-1111b}.

Type: unsigned integer.

Field length: 4 bits.

7.4.4.6 slotId (slot identifier)

Description: This parameter is the slot number within a 1ms sub-frame. All slots in one sub-frame are counted by this parameter, slotId running from 0 to N_{slot}-1. In this version of the specification the maximum N_{slot}=16, All other values of the 6 bits are reserved for future use.

Value range: {00 0000b-00 1111b=slotID, 01 0000b-11 1111b=Reserved}.

Type: unsigned integer.

Field length: 6 bits.

7.4.4.7 startSymbolid (start symbol identifier)

Description: This parameter identifies the symbol number (within a slot) of the earliest symbol, to which the information of this message is applicable.

Value range: {00 0000b-11 1111b}.

Type: unsigned integer.

Field length: 6 bits.

7.4.4.8 numberOfsections (number of sections)

Description: This parameter indicates the number of data section descriptions (separate citations of section ID even for multiple citations of the same sectionId) included in this C-Plane message.

Value range: {0000 0000b-1111 1111b}.

Type: unsigned integer.

Field length: 8 bits.

7.4.4.9 sectionType (Section Type)

Description: This parameter determines the characteristics of U-plane data to be transferred or received from a beam with one pattern id.

Value range: {0000 0000b-1111 1111b=Section Type, see table at the start of this clause for Section Type meanings}

Type: unsigned integer.

Field length: 8 bits.

7.4.4.10 udCompHdr (user data compression header)

See clause 8.3.3.13 for the description of this parameter.

The udCompHdr information is provided on the U-Plane, instructing the O-RU (on DL) and O-DU (on UL) how to interpret and decompress the received U-Plane data. However, the O-RU is not expected to decide by itself the best UL data format and data compression method, instead the O-DU shall instruct the O-RU via udCompHdr in a C-Plane message. This means the udCompHdr in a C-Plane message only has meaning for UL U-Plane data (dataDirection = 0b) and should be set to 0x00 when dataDirection = 1b (DL data), and the receiving O-RU shall ignore the received udCompHdr value. When static data format and compression is used (see 6.1.2) then udCompHdr is not needed in any C-Plane message and should be set to 0x00, and the receiving O-RU shall ignore the received udCompHdr value.

Note that it is impermissible to specify different values of udCompHdr for the same data section, even if it is sent over separate C-Plane messages discriminated by different reMask values. Only a single compression method per data section is supported (see clause 7.4.5.1). The O-RU response if the C-Plane specifies multiple udCompHdr values for a single data section is undefined.

7.4.4.11 numberOfUEs (number Of UEs)

Description: This parameter applies to Section Type 6 messages and indicates the number of UEs (for which channel information is provided) are included in the message. This allows the parser to determine when the last UE's data has been parsed.

Value range: {0000 0000b-1111 1111b}.

Type: unsigned integer.

Field length: 8 bits.

7.4.4.12 timeOffset (time offset)

Description: This parameter defines the time_offset from the start of the slot to the start of the Cyclic Prefix (CP) in number of samples T_s (=1/30.72MHz as specified in 3GPP TS38.211 clause 4.1). The value shall be less than the slot length. The slot length is based on the SCS on which the startSymbolId numerology is based (see Tables 5-1 and 5-1a in 5.3.2). For the mixed numerology case (non-PRACH), timeOffset points to the same timing pointed by startSymbolId. For PRACH, refer to clause 4.3.2.

time_offset = timeOffset * t_s

Value range: {0000 0000 0000 0000b-1111 1111 1111 1111b}.

Type: unsigned integer.

Field length: 16 bits.

7.4.4.13 frameStructure (frame structure)

Description: This parameter defines the frame structure. The first 4 bits define the FFT/iFFT size being used for all IQ data processing related to this message. The second 4 bits define the sub carrier spacing as well as the number of slots per 1ms sub-frame according to 3GPP TS 38.211, taking for completeness also 3GPP TS 36.211 into account. The parameter μ=0…5 from 3GPP TS 38.211 is extended to apply for PRACH processing.

NOTE: The parameter "FFTSize" is meant to facilitate the above calculation and is not meant to strictly dictate the O-RU's method of time-to-frequency conversion.

Value range: {0000 0000b-1111 1111b}

Bit allocations

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
FFT Size				μ (Subcarrier spacing)				1	Octet 1

Table 7-11 : FFT Size

Value of IE "FFT_size"	FFT/iFFT size
0000b	Reserved (no FFT/iFFT processing)
0001b…0110b	Reserved
0111b	128

1000b	256
1001b	512
1010b	1024
1011b	2048
1100b	4096
1101b	1536
1110,1111b	Reserved

Table 7-12 : Sub-Carrier Spacing

Value of IE "SCS"	3GPP "µ"	Subcarrier spacing Δf	Number of slots per 1ms sub-frame: N_{slot}	Slot length
0000b	0	15kHz	1	1ms
0001b	1	30kHz	2	500µs
0010b	2	60kHz	4	250µs
0011b	3	120kHz	8	125µs
0100b	4	240kHz	16	62.5µs
0101b…1011b	NA	Reserved	Reserved	Reserved
1100b	NA	1.25kHz	1	1ms
1101b	NA	3.75kHz (LTE-specific)	1	1ms
1110b	NA	5kHz	1	1ms
1111b	NA	7.5kHz (LTE-specific)	1	1ms

Type: unsigned integer (concatenated bit fields).

Field length: 8 bits.

7.4.4.14 cpLength (cyclic prefix length)

Description: This parameter defines the length CP_length of the Cyclic Prefix (CP) as follows, based on T_s (=1/30.72MHz as specified in 3GPP TS38.211 clause 4.1):

CP_length = cpLength * T_s

NOTE: cpLength parameter is used with Section Types 0 and 3, and is applicable to all symbols addressed by the C-plane message. In addition to the values defined in 3GPP, the value 0 is supported. The slot start time reference differentiation between normal CP and extended CP is based on cpLength and frameStructure. If cpLength is zero then differentiation between normal CP and extended CP time reference is based on cp-type parameter conveyed in the M-Plane. With Section Type 1, the CP properties shall be configured over M-Plane. In the latter case, the cyclic prefix shall have the following properties:

- cp-type: Normal or extended
- cp-length: same unit as cpLength defined in this section; used for symbol 0 for NR & LTE, and symbol $7*2^{\mu}$ for NR (µ as defined in the table above with "NA" for µ shall be replaced by "0")
- cp-length-other: same unit as cpLength defined in this section; used for other symbols than by cp-length

Value range: {0000 0000 0000 0000b-1111 1111 1111 1111b}.

Type: unsigned integer.

Field length: 16 bits.

7.4.5 Coding of Information Elements – Application Layer, Sections

7.4.5.1 sectionId (section identifier)

Description: If C-Plane and U-Plane coupling via sectionId (see clause 7.4.1.2.1) is used then this parameter identifies individual data sections that are described by data section descriptions within the C-Plane message. The purpose of the sectionId is to map U-Plane data sections to the corresponding C-Plane message (and Section Types) associated with the data. Two or more C-Plane data section descriptions with same SectionId may be cited corresponding to a single U-Plane data section containing a combined payload for both citations (e.g., for supporting mixed CSI RS and PDSCH). This case is applicable when usage of reMask is complimentary (or orthogonal) and different beam directions (i.e. beamIds) are given the resource elements. Data sections are specific to a value of eAxC so different eAxC values may have differently-defined data sections e.g. have different ranges of PRBs contained within.

NOTES:

- sectionIds are specific to a slot, so sectionId values may be “reused” for each slot, and the sectionId value for one slot has no specified relation to the sectionId value for a different slot
- All data in section descriptions with same SectionId value shall have same rb, startPrbc, numPrbc, udCompHdr, and numSymbol IE fields’ content.
- An upper bound on the max number of section IDs that can be addressed per eAxC or per sets of eAxCs, per symbol and per slot, for DL and for UL respectively, shall be conveyed via M-plane messaging as part of the O-RU capabilities description.
- The sectionId cited in a C-Plane message shall have the same value as the sectionId in the corresponding U-Plane message for the given data section as defined by the frameId, subFrameId, slotId, startSymbolid fields and range of relevant PRBs as indicated by the totality of the specified startPrb(c/u) and numPrb(c/u) fields.

If C-Plane and U-Plane coupling via sectionId (see clause 7.4.1.2.1) is not used, then the sectionId value is not used for identification of data sections in U-plane messages and is not used for identification of data section descriptions in C-plane messages. In this case above rules for uniqueness of sectionId value within slot and restriction on rb, startPrbc, numPrbc, udCompHdr, and numSymbol IE fields’ content do not apply. Sender shall send SectionId = 4095. The only exception to this is “Coupling via Frequency and Time with Priorities (Optimized)” where a unique value of sectionId is still used for identification of the highest priority data section descriptions in C-plane messages (clause 7.4.1.2.4, restriction 5). The corresponding U-Plane messages shall still use sectionId value as 4095 in this case.

Value range: {0000 0000 0000b-1111 1111 1111b}.

Type: unsigned integer.

Field length: 12 bits.

7.4.5.2 rb (resource block indicator)

Description: This parameter is used to indicate if every RB is used or every other RB is used. The starting RB is defined by startPrbc and total number of used RBs is defined by numPrbc. Example: RB=1, startPrb=1, numPrb=3, then the PRBs used are 1, 3, and 5. If numPrbc=0 (i.e., all PRBs), then the sending node shall set the rb value to zero, and the receiving node shall ignore whatever rb value is received and assume it is zero.

If section description includes Section Extension #6 or #12 sender shall set parameter rb to zero.

Value range: {0b=every RB used; 1b=every other RB used}.

Type: binary bit.

Field length: 1 bit.

Default Value: 0b (every RB used).

7.4.5.3 symInc (symbol number increment command)

Description: If the section description does not include Section Extension #6, #12 and #19, this parameter is used to indicate which symbol number is relevant to the given section description. If the section description includes Section

Extension #6, #12 or #19 then set of symbols relevant for the section description is given in symbolMask (see clause 7.4.7.6.3, 7.4.7.12.2 and 7.4.7.19.5). In this case, symInc and numSymbol do not affect the set of symbols relevant for the section description but can affect the following section descriptions if they do not include Section Extension #6, #12 and #19. Regardles of presence of Section Extension #6 , #12 and #19 it is expected that for each C-Plane message a symbol number is maintained and starts with the value of startSymbolid. The same value is used for each section in the message as long as symInc is zero. When symInc is one, the maintained symbol number should be incremented to the next symbol, and that new symbol number should be used for that section and each subsequent section until the symInc bit is again detected to be one. In the case of a multiple-symbol data section (numSymbol > 1), the new symbol number shall be the one after the last symbol in the data section. In this manner, multiple symbols may be handled by a single C-Plane message.

A few other points regarding use of symInc in the C-Plane:

a) SymInc may be used when different data section descriptions have a different number of symbols under certain conditions namely that the data section descriptions in the C-Plane message are carefully arranged. The arrangement shall be such that the following procedure as applied by the O-RU works correctly:

> symbol[s] - symbol addressed by section description s=1…N (considers section descriptions in one C-Plane message, s=1 identifies the first section description) shall be calculated as follows:
>
> Let symInc[s] and numSymbols[s] are values of corresponding fields of section description s
> For sake of simplicity let symbol[0] = startSymbolId, symInc[0]=0 and numSymbols[0]=1
> for s=1…N (all section descriptions in a message)
>
> > If symInc[s] = 0 then symbol[s] = symbol[s-1]
> >
> > Else symbol[s] = symbol[s-1] + numSymbols[s-1]

b) SymInc is specific to a data section description and, if Section Extension #6, #12 and #19 is not present, is affecting range of symbols described by the section description. When C-Plane and U-Plane coupling via sectionId (see clause 7.4.1.2.1) is used and a data section is referenced by multiple data section descriptions (e.g. with different reMask values) if the value of symInc is to be set to 1, then only the first invocation of the sectionId shall have symInc=1 and all other invocations of the sectionId shall have symInc=0 in the same C-Plane message. This assures that the above rule can be applied (in the same C-Plane message only) while assuring that all invocations of the data section descriptions will refer to the same symbols, which shall be the case whether the invocations are in the same C-Plane message or in separate C-Plane messages.

Use of symInc in the U-Plane is independent of the use of symInc in the C-Plane. The current view is that U-Plane message includes data for a single symbol (to avoid degrading the latency performance of the U-Plane), so use of symInc=1 is prohibited in the U-Plane as of this version of the specification. A future version may allow use of symInc=1 in the U-Plane to allow multiple symbols to be contained in a single U-Plane message.

Value range: {0b=use the current symbol number; 1b=increment the current symbol number and use that}.

Type: binary bit.

Field length: 1 bit.

Default Value: 0b (do not increment the current symbol number).

7.4.5.4 startPrbc (starting PRB of data section description)

Description: In absence of Section Extensions #6, #12 and #13 in the section description, startPrbc parameter conveys the first (lowest frequency) PRB described by the section description. The interpretation of startPrbc is affected by presence of Section Extensions #6, #12 and #13; see clause 7.4.7.6, 7.4.7.12 and 7.4.7.13 for more details.

Section description shall address resource elements without ambiguity: a resource element addressed by a section description shall not be addressed by other section description, with exception for C-plane and U-plane coupling with coupling via frequency and time with priorities which resolves ambiguity by differences in priorities (see clause 7.4.1.2.3 and 7.4.1.2.4). If coupling of C-plane and U-plane via sectionId value is used a PRB addressed by a section description shall not be addressed by a section description with a different value of sectionId.

NOTE: freqOffset affects the frequency span for specific range of PRB numbers. Therefore "shall address resource elements without ambiguity" shall consider the value of freqOffset.

Center frequency F_{RE} in Hz of a subcarrier corresponding to RE #k (k = 0 … 11) in PRB identified by startPrbc is defined by:

F_{RE} = center_of_channel_bandwidth + frequency_offset + startPrbc * 12 * Δf + k * Δf + dc_skip

where

center_of_channel_bandwidth in Hz is configured in M-plane,

frequency_offset is calculated from freqOffset field (see clause 7.4.5.11) if freqOffset is present in the C-plane message or calculated from M-plane parameter offset-to-absolute-frequency-center (in this case frequency_offset = offset-to-absolute-frequency-center * Δf * 0.5 where Δf is frequency in Hz corresponding to subcarrier spacing configured via M-Plane) otherwise,

Δf is frequency in Hz corresponding to the subcarrier spacing configured in frameStructure (see clause 7.4.4.13) if frameStructure is present in the message, or to the subcarrier spacing configured via M-Plane otherwise,

dc_skip depends on carrier type:

if carrier is of LTE DL type and frequency_offset + startPrbc * 12 * Δf + k * Δf $\geq$ 0 then dc_skip = Δf
else dc_skip = 0.

The above formulation is intended to result in resource grid position in frequency compatible with 3GPP requirements, defined in [8] and [9].

Value range: {00 0000 0000b-11 1111 1111b}.

Type: unsigned integer.

Field length: 10 bits.

7.4.5.5 reMask (resource element mask)

Description: This parameter defines the Resource Element (RE) mask within a PRB. Each bit setting in the reMask indicates if the section control is applicable to the RE sent in U-Plane messages (0=not applicable; 1=applicable). MSB indicates the value for the RE of the lowest frequency in a PRB.

Note that different REs in a PRB may be referenced by different data section descriptions. This is restricted to data section descriptions having the same sectionId but different reMask if C-Plane and U-Plane coupling via sectionId is used (see clause 7.4.1.2.1). If C-Plane and U-Plane coupling method independent on sectionId (see clause 7.4.1.2.2 or 7.4.1.2.2) is used, then data sections descriptions referencing different REs in a PRB shall have different reMask values but sectionId is not relevant. In addition, if C-Plane and U-Plane coupling via frequency and time with priorities (see clause 7.4.1.2.3) is used a RE in a PRB may be referenced by more than one data section description (in terms of reMask) but only configuration carried with data section description of highest priority is applicable (see clause 7.4.1.2.3 and 7.4.7.6.4).

The maximum number of different reMask values that may be applied to a PRB is an O-RU characteristic that is conveyed from the O-RU to the O-DU via the M-Plane. If any RE in a PRB is never pointed to by an reMask (but other REs in that PRB are), the “missing” RE should be set to zero in the U-Plane, and no beamforming ID or other processing should be applied to the “missing” RE. No RE may be referenced more than once in a data section.

Value range: {0000 0000 0000b-1111 1111 1111b}.

Type: unsigned integer (bit mask).

Field length: 12 bits.

Default Value: 1111 1111 1111b (all REs in the block applicable).

7.4.5.6 numPrbc (number of contiguous PRBs per data section description)

Description: In absence of Section Extensions #6, #12 and #13 in the section description, the numPrbc parameter conveys the number of PRBs described by the section description. The interpretation of numPrbc is affected by presence of Section Extensions #6, #12 and #13; see clause 7.4.7.6, 7.4.7.12 and 7.4.7.13 for more details.

Section description shall address resource elements without ambiguity: a resource element addressed by a section description shall not be addressed by other section description, with exception for C-plane and U-plane coupling with coupling via frequency and time with priorities which resolves ambiguity by differences in priorities (see clause 7.4.1.2.3 and 7.4.1.2.4). If coupling of C-plane and U-plane via sectionId value is used a PRB addressed by a section description shall not be addressed by a section description with a different value of sectionId.

Value range: {0000 0001b-1111 1111b, 0000 0000b = all PRBs in the specified SCS and carrier bandwidth}.

Value 0000 0000b is reserved for NR cases wherein the total number of PRBs may be more than 255. In this case, the sending node shall set the startPrbc value to zero and the receiving node shall ignore whatever startPrbc value is received and assume it is zero. For all other cases a non-zero value of numPrbc shall be used.

More specifically, if number of PRBs that needs to be sent is more than 255 but not all the PRBs in the specified SCS and carrier bandwidth, then it is expected that multiple data sections describing smaller numner of PRBs are specified.

Type: unsigned integer.

Field length: 8 bits.

7.4.5.7 numSymbol (number of symbols)

Description: In section description without Section Extension #6, #12 and #19, this parameter defines number of symbols, or number of PRACH repetitions in the case of PRACH, to which the section control is applicable. At minimum, the section control shall be applicable to at least one symbol. However, possible optimizations could allow for several (up to 14) symbols, if e.g., all 14 symbols use the same beam ID.

In section description with Section Extension #6, #12 or #19, the set of symbols referred by the description is conveyed with symbolBitmask field and numSymbol parameter does not reflect number of symbols referred by the description. In this case, the value of numSymbol can be zero or non-zero number of symbols as it affects the set of symbols referred by the following section descriptions that do not include Section Extension #6, #12 and #19 and have symInc flag set to 1.

Refer to clause 7.4.5.3 for more details.

Value range: {0001b-1110b=number of symbols, 0000b=reserved (can be used in section description with Section Extension 6 or 12), 1111b=reserved}.

Type: unsigned integer.

Field length: 4 bits.

7.4.5.8 ef (extension flag)

Description: This parameter is used to indicate if this section has any Section Extensions as described in clause 7.4.6 included in the message

Value range: {0b=no Section Extensions; 1b=one or more Section Extensions are included in this section}.

Type: binary bit.

Field length: 1 bit.

Default Value: 0b (no Section Extensions).

7.4.5.9 beamId (beam identifier)

Description: This parameter defines the beam pattern to be applied to the U-Plane data. beamId = 0 means no beamforming operation will be performed. No beamforming operation implies that the RU shall not apply any phase or amplitude weights to the U-plane data and that the resulting RF signal will be applied to all antenna elements in the group equally.

Note that the beamId encodes the beamforming to be done on the O-RU. This beamforming may be digital, analog or both (“hybrid beamforming”) and the beamId provides all the information necessary for the O-RU to select the correct beam (or weight table or beam attributes from which to create a beam). It is intended that the beamId be global for the O-RU meaning there are 32767 possible beams shared within the O-RU for all Rtcids/Pcids and shared between UL and DL (beamId=0x0000 is reserved for no beamforming). The specific mapping of beamId to e.g. weight table, beam attributes, directionality, beam adjacency or any other beam designator is specific to the O-RU design and shall be conveyed via M-Plane from the O-RU to O-DU upon startup.

NOTE: An upper bound on the max number of beamIDs that can be addressed per eAxC or per sets of eAxCs, per symbol and per slot, for DL and for UL respectively, shall be conveyed via M-plane messaging as part of the O-RU capabilities description. In addition to this, for O-RUs with C-Plane message processing limits, additional limits to restrict the number of highest priority sections per C-Plane message on top of eAxC limits shall be applied (clause 7.6.2).

Value range: {000 0000 0000 0001b-111 1111 1111 1111b; 000 0000 0000 0000b means no BF to be done}

Type: unsigned integer.

Field length: 15 bits.

Default Value: 0000 0000 0000 0000b (no beamforming).

7.4.5.10 ueId (UE identifier)

Description: This parameter provides a label for the UE for which the section contents apply. This is used to support channel information sending from the O-DU to the O-RU. This is just a label and the specific value has no meaning regarding types of UEs that may be supported within the system.

Value range: {000 0000 0000 0000b-111 1111 1111 1111b}

Type: unsigned integer.

Field length: 15 bits.

7.4.5.11 freqOffset (frequency offset)

Description: This parameter defines the frequency offset with respect to the carrier center frequency before additional filtering (e.g. for PRACH) and FFT processing (in UL) in steps of one half the subcarrier spacings Δf. The frequency offset shall be consistent per data section (one cannot use reMask to allow different frequency offsets for different REs in the PRBs).

Frequency offset in Hz is calculated as:

frequency_offset = freqOffset * Δf * 0.5

where Δf is frequency in Hz corresponding to subcarrier spacing provided in frameStructure (see clause 7.4.4.13).

NOTE: Frequency span resulting from frameStructure, freqOffset, startPrbc, numPrbc, and rb shall not exceed channel bandwidth configured for eAxC over M-plane.

Value range: { 0x000000=no offset,
0x000001 – 0x7FFFFF = positive frequency offset,
0x800000 – 0xFFFFFF = negative frequency offset }.

Type: signed integer.

Field length: 24 bits.

7.4.5.12 regularizationFactor (regularization Factor)

Description: This parameter provides a signed value to support MMSE operation within the O-RU when beamforming weights are supported in the O-RU, so related to Section Type 6.

Value range: {0000 0000 0000 0000b-1111 1111 1111 1111b}

Type: **un**signed integer.

Field length: 16 bits.

7.4.5.13 ciIsample, ciQsample (channel information I and Q values)

Description: These values are the channel information complex values relayed from the O-DU to the O-RU, related to Section Type 6. The order of transmission is first Prbc for the first antenna to the last antenna, then second Prbc for the first to last antenna, and so on until reaching the last Prbc for the first to last antenna. The bit-width per I and Q value is variable (determined by M-Plane messaging) so after the very last Q value, some number of padding (set to zero) bits may be inserted to get to the next byte boundary. When the optional "little endian byte order" is chosen via M-plane, the ciIsample/ciQsample shall use little endian byte order to transmit the complex numbers. Refer to Annex D.2 for details.

Value range: {0000 0000 0000 0000b-1111 1111 1111 1111b} for each I and Q value

Type: signed integer.

Field length: variable bit-with per I and Q value.

7.4.5.14 laaMsgType (LAA message type)

Description: This parameter defines the LAA message type being conveyed within the Section Type 7 C-Plane message. Note that here the C-Plane messages may flow from the O-DU to the O-RU (as usual) or from the O-RU to the O-DU. All of these messages relate to the listen-before-talk (LBT) LAA operation and provide a "handshake" between the O-DU and the O-RU to manage the LBT operation.

Value range: {0001b-1111b}.

Table 7-13 : laaMsgType definition

laaMsgType	lssMsgType definition	lssMsgType meaning
0000b	LBT_PDSCH_REQ	O-DU to O-RU request to obtain a PDSCH channel
0001b	LBT_DRS_REQ	O-DU to O-RU request to obtain the channel and send DRS
0010b	LBT_PDSCH_RSP	O-RU to O-DU response, channel acq success or failure
0011b	LBT_DRS_RSP	O-RU to O-DU response, DRS sending success or failure
0100b	LBT_Buffer_Error	O-RU to O-DU response, reporting buffer overflow
0101b	LBT_CWCONFIG_REQ	O-DU to O-RU request, congestion window configuration
0110b	LBT_CWCONFIG_RSP	O-RU to O-DU response, congestion window config. response
0100b – 1111b	reserved for future methods	

Type: unsigned integer.

Field length: 4 bits.

7.4.5.15 laaMsgLen (LAA message length)

Description: This parameter defines number of 32-bit words in the LAA section, where "1" means one 32-bit word, "2" means 2 32-bit words, etc. – including the word containing the lssMsgLen parameter. Zero is a reserved value.

Value range: {0001b-1111b=number of 32-bit words in the section from 1 to 16 words (4 to 64 bytes)
0000b is a reserved value }

Type: unsigned integer.

Field length: 4 bits.

7.4.5.16 lbtHandle

Description: This parameter provides a label that is included in the configuration request message (e.g., LBT_PDSCH_REQ, LBT_DRS_REQ) transmitted from the O-DU to the O-RU and returned in the corresponding response message (e.g., LBT_PDSCH_RSP, LBT_DRS_RSP).

Value range: {0000 0000 0000 0000b-1111 1111 1111 1111b}

Type: unsigned integer.

Field length: 16 bits.

7.4.5.17 lbtDeferFactor (listen-before-talk defer factor)

Description: Defer factor in sensing slots as described in 3GPP TS 36.213 [14] clause 15.1.1. This parameter is used for LBT CAT 4 and can take one of three values: {1,3, 7} based on the priority class. Four priority classes are defined in 3GPP TS 36.213 [14].

Value range: {001b, 011b, 111b} or {1, 3, 7} in decimal

Type: unsigned integer

Field length: 3 bits

7.4.5.18 lbtBackoffCounter (listen-before-talk backoff counter)

Description: LBT backoff counter in sensing slots as described in 3GPP TS 36.213 [14] clause 15.1.1.

Value range: {00 0000 0000b – 11 1111 1111b} (0-1023 decimal)

Type: unsigned integer

Field length: 10 bits

7.4.5.19 lbtOffset (listen-before-talk offset)

Description: LBT start time in microseconds from the beginning of the subframe scheduled by this message

Value range: {00 0000 0000b – 11 1110 0111b} or {0 – 999} in decimal

Type: unsigned integer

Field length: 10 bits

7.4.5.20 MCOT (maximum channel occupancy time)

Description: LTE TXOP duration in subframes as described in 3GPP TS 36.213 [14] clause 15.1.1. The maximum values for this parameter are {2, 3, 8, 10} based on the priority class. Four priority classes are defined in 3GPP TS 36.213 [14].

Value range: {1-10} in decimal

Type: unsigned integer

Field length: 4 bits

7.4.5.21 lbtMode (LBT Mode)

Description: Part of multi-carrier support. Indicates whether full LBT process is carried or partial LBT process is carried (multi carrier mode B according to 3GPP TS 36.213 [14] clause 15.1.5.2).

00b = full LBT (regular LBT, sending reservation signal until the beginning of the SF/slot)
01b = Partial LBT (looking back 25 µsec prior to transmission as indicated in 3GPP TS 36.213[14] clause 15.1.5.2)
10b = Partial LBT (looking back 34 µsec prior to transmission as indicated in 3GPP TS 36.213 [14] clause 15.1.5.2)
11b = full LBT and stop (regular LBT, do NOT send reservation signal; O-RU senses the spectrum for the defer factor + a sensing slot right before OTA when the O-DU is ready to transmit data as indicated in 3GPP TS 36.213 [14] clause 15.1.1. i.e., right before the SF/slot boundary)

Value range: {00b - 11b}

Type: unsigned integer

Field length: 2 bits

7.4.5.22 lbtPdschRes (LBT PDSCH Result)

Description: LBT result of SFN/SF

- 00b = not sensing – indicates that the O-RU is transmitting data
- 01b = currently sensing – indicates the O-RU has not yet acquired the channel
- 10b = success – indicates that the channel was successfully acquired
- 11b = Failure – indicates expiration of the LBT timer. The LBT process should be reset.

Value range: {00b - 11b}

Type: unsigned integer

Field length: 2 bits

7.4.5.23 sfStatus (subframe status)

Description: indicates whether the subframe was dropped or transmitted

- 0 – subframe was dropped
- 1 – subframe was transmitted

Value range: {0,1}

Type: binary bit

Field length: 1 bit

7.4.5.24 lbtDrsRes (LBT DRS Result)

Description: LBT result of SFN/SF

- 0 – SUCCESS – indicates that DRS is sent
- 1 – FAILURE – indicates that DRS is not sent

Value range: {0,1}

Type: binary bit

Field length: 1 bit

7.4.5.25 initialPartialSF (Initial partial SF)

Description: Indicates whether the initial SF in the LBT process is full or partial.

- 0 – full SF (two slots, 14 symbols)
- 1 – partial SF (only second slot, last 7 symbols)

Value range: {0, 1}

Type: binary bit

Field length: 1 bit

7.4.5.26 lbtBufErr (LBT Buffer Error)

Description: Indicates when an LBT buffer overflow has occurred.

- 0 – reserved

- 1 – buffer overflow – data received at O-RU is larger than the available buffer size

Value range: {0, 1}

Type: binary bit

7.4.5.27 sfnSf (SFN/SF End)

Description: SFN/SF which is dropped by O-RU because of time expired or successfully transmitted

Value range: SFN: {0000 0000b – 1111 1111b} or {0 – 255} in decimal, SF: {0000b – 1001b} or {0 – 9} in decimal

Type: unsigned integer

Field length: 12 bits

7.4.5.28 lbtCWConfig_H (HARQ Parameters for Congestion Window management)

Description: Total number of HARQ NACK feedback messages received for the lbtCWConfig_T transport blocks transmitted over LAA cell(s)to be used for adjusting the Congestion Window.

Value range: {0000 0000b – 1111 1111b} or {0 – 255} in decimal

Type: unsigned integer

Field length: 8 bits

7.4.5.29 lbtCWConfig_T (TB Parameters for Congestion Window management)

Description: Total number of Transport Blocks (TB) transmitted over the LAA cell to be used for adjusting the Congestion Window.

Value range: {0000 0000b – 1111 1111b} or {0 – 255} in decimal

Type: unsigned integer

Field length: 8 bits

7.4.5.30 lbtTrafficClass (Traffic class priority for Congestion Window management)

Description: Channel access priority class as defined in TS 36.213 [14]

Value range: {000 - 111} or {0 - 7} in decimal

1 - 4: traffic class priority

0, 5 - 7: reserved

Type: unsigned integer

Field length: 3 bits

7.4.5.31 lbtCWR_Rst (Notification about packet reception successful or not)

Description: Notifies to O-DU whether the O-RU receives LBT_CWCONFIG_REQ message successfully or not

- 0 – SUCCESS – indicates successful reception of LBT_CWCONFIG_REQ
- 1 – FAILURE – indicates failure of receiving LBT_CWCONFIG_REQ

Value range: {0, 1}

Type: binary bit

Field length: 1 bit

7.4.5.32 reserved (reserved for future use)

Description: This parameter is reserved for future use. Transmitter shall send value “0”, while receiver shall ignore the value received.

Value range: {variable}.

Type: variable.

Field length: variable.

7.4.6 Section Extension Commands

The following Section Extension parameters are defined within the C-Plane:

Table 7-14: Section Extension Commands

extType	meaning	extLen	extension parameters	octets	meaning
0	reserved	1 (1 word)	reserved reserved	1 1	for future use for future use
1	beamforming weights	var	bfwCompHdr bfwCompParam bfwI (for TRX 0) bfwQ (for TRX 0) … bfwI (for last TRX) bfwQ (for last TRX)	1 1 var var var var	bitWidth(3:0) | compMeth(3:0) depends on compr. method beamforming weight I value beamforming weight Q value beamforming weight I value beamforming weight Q value
2	beamforming attributes	var	bfaCompHdr bfAzPt bfZePt bfAz3dd bfZe3dd bfAzSl bfZeSl	2 var var var var 3b 3b	BF attributes compr. header BF azimuth pointing param BF zenith pointing param BF azimuth beamwidth param BF zenith beamwidth param BF azimuth sidelobe param BF zenith sidelobe param
3	DL Precoding configuration parameters and indications	var (3 or 4 words)	codebookIndex layerId txScheme numLayers crsReMask crsSymNum crsShift beamIdAP1 beamIdAP2 beamIdAP3	1 4b 4b 4b 12b 4b 1b 15b 15b 15b	precoder codebook layer ID for DL Tx transmission scheme number of layers in DL Tx CRS RE Mask CRS symbol number CRS shift command Beam ID, Antenna Port 1 Beam ID, Antenna Port 2 Beam ID, Antenna Port 3
4	modulation compr. params	0	csf modCompScaler	1b 15b	constellation shift flag mod. compr. scale value
5	modulation compression additional scaling parameters	var	mcScaleReMask csf mcScaleOffset	12b 1b 15b	Position of same scaling bits constellation shift flag added mod. compr. scale values
6	Non-contiguous PRB allocation	2 (2 words)	repetition rbgSize rbgMask priority symbolMask	1b 3b 28b 2b 14b	repetition flag number of PRBs in the group mask of RBGs in the symbol(s) priority of section description mask of symbols in the slot
7	Multiple-eAxC designation	1	eAxCmask	16b	eAxC mask
8	regularization factor	1	regularizationFactor	16b	regularization factor

extType	meaning	extLen	extension parameters	octets	meaning
9	Dynamic Spectrum Sharing parameters	1	technology	1	interface name
10	Multiple ports grouping	var	beamGroupType numPortc beamID (or ueID)	2b 6b 15b	type of beam grouping the number of ports beam ID (or UE ID)
11	Flexible BF weights	var	disableBFWs RAD numBundPrb bfwCompHdr bfwCompParam for bundle 0 beamId (for PRB bundle 0) bfwI (for TRX 0, bundle 0) bfwQ (for TRX 0, bundle 0) … bfwI (for last TRX, bundle 0) bfwQ (for last TRX bundle 0) … bfwCompParam for last bundle beamId (for last PRB bundle) bfwI (for TRX 0, last bundle) bfwQ (for TRX 0, last bundle) … bfwI (for last TRX, last bundle) bfwQ (for last TRX & bundle)	1b 1b 1 1 var 2 var var … var var … var 2 var var var var	disable beamforming weights Reset After (PRB) Discontinuity Number of bundled PRBs bitWidth(3:0) | compMeth(3:0) depends on compr. method Beam ID BF weight I value for bundle 0 BF weight Q value for bundle 0 … BF weight I value for bundle 0 BF weight Q value for bundle 0 … depends on compr. method Beam ID BF weight I value for last bundle BF weight Q value for last bundle … BF weight I value for last bundle BF weight Q value for last bundle

12	non-contiguous PRB allocation with frequency ranges	var	priority,
			symbolMask
			offStartPrb(1)
			numPrb(1)
			…
			offStartPrb(R-1)
			numPrb(R-1)
				2b
				14b
				8b
				8b
				
				8b
				8b
					priority of section description
					mask of symbols in the slot
					offset to start of PRB range #1
					number of PRBs in the range #1
					
					offset to start of PRB range
					number of PRBs in the range

13	frequency hopping	var	nextSymbolId(1) nextStartPrbc (1) … nextSymbolId (R-1), nextStartPrbc (R-1)	4b 10b 4b 10b	start symbol of hop #1 start PRB for hop #1 start symbol of hop #R-1 start PRB for hop #R-1
14	Null-layer Info. for ueId-based beamforming	var	nullLayerInd	1	Nulling-layer indication
15	Mixed-numerology Info. for ueId-based beamforming	var	frameStructure freqOffset cpLength	1 3 2	FFT size, mu (SCS) Frequency offset Cyclic prefix length
16	Antenna mapping for UE channel Info based UL beamforming		antMask … antMask	8 var	Bitmask of Max. 64 antenna Bitmask of Max. 64 antenna
17	User port group indication	var	numUeID	4b	Number of ueIDs per user
18	Uplink Transmission Management	2 (2 words)	transmissionWindowOffset transmissionWindowSize toT	16b 14b 2b	transmission window offset transmission window size type of transmission

extType	meaning	extLen	extension parameters	octets	meaning
19	Compact multiple port beamforming information	var	disableBFWs	1b	disable BF weights
			repetition	1b	repetition flag
			priority	2b	priority of section description
			numPortc	6b	num ports
			symbolMask	14b	resource symbol bitmask
			bfwCompHdr	8b	BF weight compression header
			…		
			portReMask(for port 1)	12b	RE bitmask for port 1
			portSymbolMask(for port 1)	14b	Symbol bitmask for port 1
			bfwCompParam(for port 1)	var	BF weight compression parameter for port 1
			…		
			beamId(for port 1)	2	beam identifier for port 1
			bfwI (for port 1 and TRX 0)	var	BF weight in-phase value for port 1 TRX0
			…		
			bfwQ (for port 1 and TRX 0)	var	BF weight quadrature-phase value port 1 TRX0
			…		
			portReMask(for last port)		RE bitmask for last port
			portSymbolMask(for last port)		Symbol bitmask last port
			bfwCompParam(for last port)		BF weight compression parameter for last port
			…		
			beamId(for last port)		beamId for last port
			bfwI (for last port and TRX 0)		BF weight in-phase value for last port and TRX0
			…		
			bfwQ (for last port and TRX 0)		BF weight quadrature-phase value last port TRX0
			…		
20	Dedicated puncturing	var	numPuncPatterns	8b	number of puncturing patterns
			symbolMask(1)	14b	first puncturing pattern symbol mask
			startPuncPrb(1)	10b	first PRB to which puncturing pattern applies
			…		
			numPuncPrb(1)	10b	number of contiguous PRBs to which first puncturing pattern applies
			…		
			…		
			puncReMask(1)	12b	first puncturing pattern RE mask
			rb(1)	1b	first RB indicator
			rbgIncl(1)	1b	first rbg included flag
			rbgSize(1)	3b	first rbg size
			rbgMask(1)	28b	first rbg bitmask
			symbolMask(last)	14b	last puncturing pattern symbol mask
			startPuncPrb(last)	10b	last PRB to which puncturing pattern applies
			…		
			numPuncPrb(last)	10b	number of contiguous PRBs to which last puncturing pattern applies
			…		
			…		
			puncReMask(last)	12b	last puncturing pattern RE mask
			rb(last)	1b	last RB indicator
			rbgIncl(last)	1b	last rbg included flag
			rbgSize(last)	3b	last rbg size
			rbgMask(last)	28b	last rbg bitmask
21-127	reserved	1 (1 word)	reserved	1	for future use
			reserved	1	for future use

7.4.6.1 extType (extension type)

Description: This parameter provides the extension type which provides additional parameters specific to the subject data extension. An O-RU or O-DU receiving a "reserved" Section Extension shall ignore the extension and all parameters contained within it.

Value range: {all zeros – all ones}. See **Table 7-14** for values.

Type: unsigned integer.

Field length: 7 bits.

7.4.6.2 ef (extension flag)

Description: This parameter is used to indicate if there is another extension present (ef=1) or this is the last extension ef=0)

Value range: {0b=no more extensions; 1b=one or more extensions are included after this one}.

Type: binary bit.

Field length: 1 bit.

7.4.6.3 extLen (extension length)

Description: This parameter provides the length of the Section Extension in units of 32-bit (or 4-byte) words. The value zero is reserved, so there is always at least one word in the extension (the word containing the extType and extLen).

Value range: {all zeros – all ones} – from one word to 255 words (8 bits) or 65535 words (16 bits).

Type: unsigned integer.

Field length: 8 bits for all Section Extensions except for Section Extension #11,19 and 20 for which the field length is 16 bits.

7.4.7 Coding of Information Elements – Application Layer, Section Extensions

7.4.7.1 ExtType=1: Beamforming Weights Extension Type

This section applies to the sending of beamforming weights from the O-DU to the O-RU. When this is done, the weights are sent along with a beamID which is meant to allow those same weights to be used in future C-Plane messages by invoking the same beamID (without the need to send the weights again). This allows downloaded weights to have “persistence” which should save DL throughput by not requiring sending of weights multiple times. The K parameter is defined in clause 12.5.2. Refer to Annex D.1 for details on the usual weight byte order. The optional “little endian byte order” is applied to bfwI/bfwQ fields if chosen via M-plane. Refer to Annex D.2 for details of little endian byte order. This Section Extension applies only to Section Types 1 and 3.

Table 7-15 : Extension Type 1 Data Format

ef	extType = 0x1	1	Octet N
extLen		1	Octet N+1
bfwCompHdr		1	Octet N+2
bfwCompParam		var	Octet N+3
bfwI (for TRX 0)		var	
bfwQ (for TRX0)		var	
remaining beamforming weights bfwI and bfwQ up to K TRXs		var	
zero pad to 4-byte boundary		var	

7.4.7.1.1 bfwCompHdr (beamforming weight compression header)

Description: This parameter defines the compression method and IQ bit width for the beamforming weights in the specific section in the C-Plane message. In this way each set of weights may employ a separate compression method.

Note that for the block compression methods, the block size is the entire vector of beamforming weights, not some subset of them.

Value range: {0000 0000b-1111 1111b}

Bit allocations

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
bfwIqWidth				bfwCompMeth				1	Octet 1

Table 7-16 : udIqWidth definition

bfwIqWidth	Bit width of each I and each Q

0000-1111b	value of bfwIqWidth except a value of zero means 16 bits
	e.g. bfwIqWidth = 0000b means I and Q are each 16 bits wide;
	e.g. bfwIqWidth = 0001b means I and Q are each 1 bit wide;
	e.g. bfwIqWidth = 1111b means I and Q are each 15 bits wide

Table 7-17 : bfwCompMeth definition

bfwCompMeth	compression method	udIqWidth meaning
0000b	no compression	bitwidth of each uncompressed I and Q value
0001b	block floating point	bitwidth of each I and Q mantissa value
0010b	block scaling	bitwidth of each I and Q scaled value
0011b	μ-law	bitwidth of each compressed I and Q value
0100b	beamspace compression	bitwidth of each beamspace I and Q coefficient
0101b – 1111b	reserved for future methods	depends on the specific compression method

Type: unsigned integer (concatenated bit fields).

Field length: 8 bits.

Default Value: 0000 0000b (no compression, 16-bit I and Q).

7.4.7.1.2 bfwCompParam (beamforming weight compression parameter)

Description: This parameter applies to the compression method specified by the associated bfwCompMeth value.

Value range: {0000 0000b-1111 1111b}.

Bit allocations

bfwCompMeth	0 (msb)	1	2	3	4	5	6	7 (lsb)	compParam size
0000b = no compression	absent								0 octets
0001b = block fl. point	reserved (set to all zeros)				Exponent (unsigned)				1 octet
0010b = block scaling	blockScaler (unsigned, 1 integer bit, 7 fractional bits)								1 octet
0011b = μ-law	compBitWidth				compShift				1 octets
0100b = beamspace	activeBeamspaceCoefficientMask								ceil(K/8) octets*
	blockScaler (unsigned,1 integer bit,7 fractional bits)								1 octet
0101b – 1111b	reserved (set to all zeros)								? octets

*K is the number of elements in uncompressed beamforming weight vector (see clause 12.5.2 and Annex J). K is O-RU-specific and is calculated from parameters describing array conveyed from the O-RU to the O-DU as part of the initialization procedure via the M-Plane

Type: unsigned integer (concatenated bit fields).

Field length: zero for bfwCompMeth value 0000b, 8 bits for bfwCompMeth values 0001b, 0010b and 0011b; other bfwCompMeth values may imply other lengths but will always be an integer number of bytes.

7.4.7.1.3 bfwI (beamforming weight in-phase value)

Description: This parameter is the In-phase beamforming weight value. Refer to Annex D.1 for details on the usual weight byte order. When optional “little endian byte order” is chosen via M-plane, refer to Annex D.2 for details on byte order. The total number of weights in the section is O-RU-specific and is conveyed from the O-RU to the O-DU as part of the initialization procedure via the M-Plane.

Value range: {all zeros – all ones}.

Type: signed integer.

Field length: 1-16 bits.

7.4.7.1.4 bfwQ (beamforming weight quadrature value)

Description: This parameter is the Quadrature beamforming weight value. Refer to Annex D.1 for details on the usual weight byte order. When optional “little endian byte order” is chosen via M-plane, refer to Annex D.2 for detail on byte order. The total number of weights in the section is O-RU-specific and is conveyed from the O-RU to the O-DU as part of the initialization procedure via the M-Plane.

Value range: {all zeros – all ones}.

Type: signed integer.

Field length: 1-16 bits.

7.4.7.2 ExtType=2: Beamforming Attributes Extension Type

This Section Extension applies only to Section Types 1 and 3.

Section Extension 2 applies to the sending of beamforming attributes from the O-DU to the O-RU which is described in Annex J.

The following table shows the format of this Section Extension.

Table 7-18 : Extension Type 2 Data Format

ef	extType = 0x02		1	Octet N
extLen			1	Octet N+1
bfaCompHdr			2	Octet N+2
bfAzPt			var	Octet N+3
bfZePt			var	
bfAz3dd			var	
bfZe3dd			var	
zero-padding	bfAzSl	bfZeSl	1	
zero padding to achieve 4-byte alignment as needed				

7.4.7.2.1 bfaCompHdr (beamforming attributes compression header)

Description: This parameter defines the bit width for the beamforming attributes extension parameters. In this way each set of beamforming attributes may employ a different bit width.

Value range: {0000 0000 0000b-1111 1111 1111b}

Bit allocations

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
Reserved		bfAzPtWidth			bfZePtWidth			1	Octet 1
Reserved		bfAz3ddWidth			bfZe3ddWidth			1	Octet 2

Table 7-19 : fzAzPtWidth definition

bfAzPtWidth	Bit width of bfAzPt
000-111b	value of bfAzPtWidth

Table 7-20 : bfZePtWidth definition

bfZePtWidth	Bit width of bfZePt
000-111b	value of bfZePtWidth

Table 7-21 : bfAz3ddWidth definition

bfAz3ddWidth	Bit width of bfAz3dd
000-111b	value of bfAz3ddWidth

Table 7-22 : bfZe3ddWidth definition

bfZe3ddWidth	Bit width of bfZe3dd
000-111b	value of bfZe3ddWidth

For each of the four bitwidth values in this parameter (bfAzPtWidth, bfZePtWidth, bfAz3ddWidth, and bfZe3ddWidth) the following mapping shall be used:

000b = no bits, the field is not applicable (O-RU cannot support it) or the default value shall be used.
001b = 2-bit bitwidth
010b = 3-bit bitwidth
011b = 4-bit bitwidth
100b = 5-bit bitwidth
101b = 6-bit bitwidth
110b = 7-bit bitwidth
111b = 8-bit bitwidth (this is the highest bitwidth anticipated to be needed)

Type: unsigned integer (concatenated bit fields).

Field length: 16 bits (4 bits are reserved).

Default Value: 0011 1111 0011 1111b (8-bit azimuth and zenith pointing angle and 8-bit azimuth and zenith beamwidth).

7.4.7.2.2 bfAzPt (beamforming azimuth pointing parameter)

Description: This parameter is the azimuth beamforming pointing angle in degrees. The valid range of values is O-RU-specific and is conveyed from the O-RU to the O-DU as part of the initialization procedure via the M-Plane.

Value range: {all zeros – all ones}.

Type: signed integer.

Field length: 0-8 bits.

7.4.7.2.3 bfZePt (beamforming zenith pointing parameter)

Description: This parameter is the zenith beamforming pointing angle in degrees. The valid range of values is O-RU-specific and is conveyed from the O-RU to the O-DU as part of the initialization procedure via the M-Plane.

Value range: {all zeros – all ones}.

Type: unsigned integer.

Field length: 0-8 bits.

7.4.7.2.4 bfAz3dd (beamforming azimuth beamwidth parameter)

Description: This parameter is the azimuth beamforming beamwidth in degrees. The valid range of values is O-RU-specific and is conveyed from the O-RU to the O-DU as part of the initialization procedure via the M-Plane. The value (000b) corresponds to the minimum valid beamwidth.

Value range: {all zeros – all ones}.

Type: unsigned integer.

Field length: 0-8 bits.

7.4.7.2.5 bfZe3dd (beamforming zenith beamwidth parameter)

Description: This parameter is the zenith beamforming beamwidth in degrees. The valid range of values is O-RU-specific and is conveyed from the O-RU to the O-DU as part of the initialization procedure via the M-Plane. The value (000b) corresponds to the minimum valid beamwidth.

Value range: {all zeros – all ones}.

Type: unsigned integer.

Field length: 0-8 bits.

7.4.7.2.6 bfAzSl (beamforming azimuth sidelobe parameter)

Description: This parameter is the azimuth beamforming sidelobe suppression value in dB. The valid range of values is O-RU-specific and is conveyed from the O-RU to the O-DU as part of the initialization procedure via the M-Plane. The value of bfAzSl corresponds to a value of 10 dB for all zeros and increments in 5 dB steps (e.g. 001b corresponds to 15 dB, 010b corresponds to 20 dB, and so on). The value 111b corresponds to having a sidelobe suppresion of 45dB or more.

Value range: {all zeros – all ones}.

Type: unsigned integer.

Field length: 3 bits.

7.4.7.2.7 bfZeSl (beamforming zenith sidelobe parameter)

Description: This parameter is the zenith beamforming sidelobe suppression value in dB. The valid range of values is O-RU-specific and is conveyed from the O-RU to the O-DU as part of the initialization procedure via the M-Plane. The value of bfZeSl corresponds to a value of 10 dB for all zeros and increments in 5 dB steps (e.g. 001b corresponds to 15 dB, 010b corresponds to 20 dB, and so on). The value 111b corresponds to having a sidelobe suppresion of 45dB or more.

Value range: {all zeros – all ones}.

Type: unsigned integer.

Field length: 3 bits.

7.4.7.2.8 zero-padding

Description: This parameter is intended to pad out the data to the next 4-byte boundary. Because the preceding parameters are of varying bitwidth, extra padding in most cases will be needed to achieve the 4-byte boundary condition. Transmitter shall send value "0", while receiver shall ignore the value received.

Value range: {variable}.

Type: variable.

Field length: variable.

7.4.7.3 ExtType=3: DL Precoding Extension Type

This Section Extension applies only to Section Types 1 and 3 and is to be used only for LTE TM2, TM3 and TM4. For other LTE transmission modes and for NR, precoding is assumed to be included in the beamforming operation (that is, encoded in the beamforming weights).

The following table shows the format of this Section Extension.

Table 7-23 : Extension Type 3 Data Format – first data layer

0 (msb)	1	2	3	4	5	6	7 (lsb)	# of bytes	
ef	extType = 0x03							1	Octet N
extLen = 0x04 (4 words)								1	N+1
codebookIndex								1	N+2
layerId = 0000b or 1111b				numLayers[3:0]				1	N+3
txScheme[3:0]				crsReMask[11:8]				1	N+4
crsReMask[7:0]								1	N+5
crsShift	reserved			crsSymNum[3:0]				1	N+6
reserved								3	N+7
reserved	beamIdAP1[14:8]							1	N+10
beamIdAP1[7:0]								1	N+11
reserved	beamIdAP2[14:8]							1	N+12
beamIdAP2[7:0]								1	N+13
reserved	beamIdAP3[14:8]							1	N+14
beamIdAP3[7:0]								1	N+15

Table 7-24 : Extension Type 3 Data Format – not first data layer

0 (msb)	1	2	3	4	5	6	7 (lsb)	# of bytes	
ef	extType = 0x03							1	Octet N
extLen = 0x01 (1 word)								1	N+1
codebookIndex								1	N+2
layerId ≠ 0000b or 1111b				numLayers				1	N+3

There may be two or four antenna ports hence two or four beamIDs needed (same beamID for user data and CRS REs). For Antenna Port 0, the beamId is contained in the C-Plane data section header, while the Antenna Ports 1-3 beamIDs are contained in this Section Extension. When there are two antenna ports, the Section Extension only contains the second Antenna Port beam ID ("beamIdAP1") and the Section Extension length is 3 words ("extLen" = 0x3). When there are four antenna ports, the Section Extension contains the second, third and fourth Antenna Port beam IDs ("beamIdAP1", "beamIdAP2". And "beamIdAP3") and the Section Extension length is 4 words ("extLen" = 0x4).

For the txScheme indicating TxD, one Pcid is used for all the user data and one section instantiation is needed using the corresponding C-Plane Rtcid, providing all the beam IDs (up to 4) for the user data; a second section instantiation (same sectionId) with a different reMask may be used to provide the CRS RE beam IDs (also up to 4). In the TxD case the layerId is set to 1111b (“TxD”).

For the txSchemes indicating spatial multiplexing (LD CDD or no CDD), each layer will have its own Pcid for the user data with a corresponding C-Plane Rtcid conveying the user data’s beamId, with this Section Extension showing a different layer number for each layer. Only within the layer ID zero Rtcid will the CRS REs be provided with their beamIDs (one beamId in the section header and the other beamIDs in this Section Extension). For the non-zero layer number Rtcids, this Section Extension will still be provided to guide the precoding operation (provide the layer ID) but will not include beamIds (extLen = 0x1), and the beamId in the section header should be ignored by the O-RU and should be set to the default value by the O-DU.

7.4.7.3.1 codebookIndex (precoder codebook used for transmission)

Description: This parameter defines the indices of the precoder codebook that are used for precoding. It is to be used in conjunction with the numLayers field. (Invalid for TM1, TM2 and TM3)

Value range: {0000 0000b - 1111 1111b}.

Type: unsigned integer.

Field length: 8 bits.

DefaultValue: 0000 0000b (used for invalid mode)

7.4.7.3.2 layerID (Layer ID for DL transmission)

Description: This parameter defines the layer ID that are used for DL transmission in TM1 – TM4.

Value range: {0000b-1111b}. 0000b implies layer0, 0001b implies layer1, 0010b implies layer2, 0011b implies layer3. (for TxD, set to all ones)

Type: unsigned integer.

Field length: 4 bits.

DefaultValue: 1111b (used for TxD mode)

7.4.7.3.3 txScheme (transmission scheme)

Description: This parameter defines the TM scheme used in this Section Type.

Value range: {0000b-1111b}

txScheme	TM scheme
0000b	Spatial multiplexing (CDD)
0001b	Spatial multiplexing (no CDD)
0010b	Transmit diversity
0011b-111b	Reserved

Type: unsigned integer.

Field length: 4 bits.

7.4.7.3.4 numLayers (number of layers used for DL transmission)

Description: This parameter defines the number of layers that are used for DL transmission in TM1 – TM6.

Value range: {0000b-1111b}. 0000b implies 1 layer, 0001 implies 2 layers, 0010 implies 3 layers, 0011b implies 4 layers,

Type: unsigned integer.

Field length: 4 bits.

7.4.7.3.5 crsReMask (CRS resource element mask)

Description: This parameter defines the CRS Resource Element (RE) mask within a PRB. Each bit setting in the crsReMask indicates if the section control is applicable to the RE sent in U-Plane messages (0=not applicable, 1=applicable). MSB indicates the value for the RE of the lowest frequency in a PRB.

Value range: {0000 0000 0000b-1111 1111 1111b}

Type: unsigned integer (bit mask).

Field length: 12 bits.

7.4.7.3.6 crsSymNum (CRS symbol number indication)

Description: This parameter defines the CRS symbol number within a PRB. The value of the crsSymNum index indicates the symbol number to the RE sent in U-Plane messages (0=not applicable, 1=applicable)

Value range: {0000b-1111b}, value indicates symbol number.

0000b – 1101b : use symbol number 0 – 13 respectively;

1110b – 1111b : reserved

Type: unsigned integer.

Field length: 4 bits.

7.4.7.3.7 crsShift (crsShift used for DL transmission)

Description: This parameter indicates the shift pattern to pick up the right index for CRS positions for N Antennas (see **Figure I-8**)

Value range: 0 or 1, implying shift patterns that are layer-dependent according to the table below (see **Tables I-1** through **I-3** for vShift)

	1 Layer	2 Layers	4 Layers
crsShift	**=0 for** 0 ≤ vshift ≤ 5	**0 for** 0 ≤ vshift ≤ 2 **1 for** 3 ≤ vshift ≤ 5	**0 for** 0 ≤ vshift ≤ 2 **1 for** 3 ≤ vshift ≤ 5

Type: binary.

Field length: 1 bit.

7.4.7.3.8 beamIdAP1 (beam id to be used for antenna port 1)

Description: This parameter defines the beam pattern to be applied to the U-Plane data. beamId = 0 means no beamforming operation will be performed. No beamforming operation implies that the RU shall not apply any phase or amplitude weights to the U-plane data and that the resulting RF signal will be applied to all antenna elements in the group equally.

Value range: {000 0000 0000 0001b-111 1111 1111 1111b}

Type: unsigned integer.

Field length: 15 bits.

Default Value: 000 0000 0000 0000b (no beamforming).

7.4.7.3.9 beamIdAP2 (beam id to be used for antenna port 2)

Description: This parameter defines the beam pattern to be applied to the U-Plane data. beamId = 0 means no beamforming operation will be performed. No beamforming operation implies that the RU shall not apply any phase or amplitude weights to the U-plane data and that the resulting RF signal will be applied to all antenna elements in the group equally.

Value range: {000 0000 0000 0001b-111 1111 1111 1111b}

Type: unsigned integer.

Field length: 15 bits.

Default Value: 000 0000 0000 0000b (no beamforming).

7.4.7.3.10 beamIdAP3 (beam id to be used for antenna port 3)

Description: This parameter defines the beam pattern to be applied to the U-Plane data. beamId = 0 means no beamforming operation will be performed. No beamforming operation implies that the RU shall not apply any phase or amplitude weights to the U-plane data and that the resulting RF signal will be applied to all antenna elements in the group equally.

Value range: {000 0000 0000 0001b-111 1111 1111 1111b}

Type: unsigned integer.

Field length: 15 bits.

Default Value: 000 0000 0000 0000b (no beamforming).

7.4.7.4 ExtType=4: Modulation Compression Parameters Extension Type

This Section Extension applies only to Section Types 1, 3 and 5. Section Extension 4 enables the O-DU to convey one set of “csf and modCompScaler values” to the O-RU which is needed for modulation compression described in Annex A.5.

Table 7-25 shows the Section Extension format.

Table 7-25: Section Format for Section Extension 4 (modulation compression parameters)

ef	extType = 0x04	1	Octet N
extLen = 0x01 (1 word)		1	N+1
csf	modCompScaler[14:8]	1	N+2
modCompScaler[7:0]		1	N+3

7.4.7.4.1 csf (constellation shift flag)

Description: This binary flag indicates whether to shift the constellation (csf=1) or not (csf=0). “Shift” means subtract from (during compression) or add to (during decompression) the I and Q samples the value $2^{-udIqWidth}$ where “udIqWidth” is the number of I and Q bits in the U-Plane representation.

Table 7-26:Constellation shift definition

udIqWidth	Shift value
1	1/2
2	1/4
3	1/8
4	1/16
5	1/32

Value range: {0b-1b}

Type: binary.

Field length: 1 bit.

7.4.7.4.2 modCompScaler (modulation compression scaler value)

Description: This parameter is the scale factor to apply to the unshifted constellation points during decompression. It is a fractional floating-point value having an unsigned but negative 4-bit exponent and an unsigned fractional 11-bit mantissa.

Value range: { 0 through +(1-2^{-11}) }.

Type: unsigned fractional floating-point value.

$$"mantissa" = \sum_{k=0}^{10} modCompScaler[k] \cdot 2^{k-11}$$

$$"exponent" = \sum_{k=11}^{14} modCompScaler[k] \cdot 2^{k-11}$$

"exponent" is the most significant 4 bits of the 15-bit modCompScaler field and "mantissa" is the least-significant 11 bits of the modCompScaler field. "modCompScaler[k] refers to the kth bit of the modCompScaler field. Therefore, the actual value of modCompScaler is:

$$modCompScaler = mantissa \cdot 2^{-exponent}$$

Field length: 15 bits.

7.4.7.5 ExtType=5: Modulation Compression Additional Parameters Extension Type

This Section Extension applies only to Section Types 1, 3 and 5. Section Extension #5 enables the O-DU to convey one (or more) set(s) of "mcScaleReMask, csf and mcScaleOffset values" to the O-RU which is needed for modulation compression described in Annex A.5. **Table 7-27** and **Table 7-28** shows the Section Extension format when one set and two sets of "mcScaleReMasks, csf and mcScaleOffset values" are conveyed. Please note that Section Extension 5 may be used to convey more than 2 sets of "mcScaleReMasks, csf and mcScaleOffset values" in which case the frame structure is extended in similar fashion, i.e. the zero padding bits are added at the end of the Section Extension to maintain 4-byte alignment.

Table 7-27 : Section Format for Section Extension 5 (one scaler value, modulation compression parameters)

<table>
<tr><td>ef</td><td colspan="7">extType = 0x05</td><td>1</td><td>Octet N</td></tr>
<tr><td colspan="8">extLen = 0x2 (2 words)</td><td>1</td><td>N+1</td></tr>
<tr><td colspan="8">mcScaleReMask[11:4]</td><td>1</td><td>N+2</td></tr>
<tr><td colspan="4">mcScaleReMask[3:0]</td><td>csf</td><td colspan="3">mcScaleOffset [14:12]</td><td>1</td><td>N+3</td></tr>
<tr><td colspan="8">mcScaleOffset [11:4]</td><td>1</td><td>N+4</td></tr>
<tr><td colspan="4">mcScaleOffset [3:0]</td><td colspan="4">zero padding</td><td>1</td><td>N+5</td></tr>
<tr><td colspan="8">zero padding</td><td>1</td><td>N+6</td></tr>
<tr><td colspan="8">zero padding</td><td>1</td><td>N+7</td></tr>
</table>

Table 7-28 : Section Format for Section Extension 5 (two scaler values, modulation compression parameters)

<table>
<tr><td>ef</td><td colspan="7">extType = 0x05</td><td>1</td><td>Octet N</td></tr>
<tr><td colspan="8">extLen = 0x03 (3 words)</td><td>1</td><td>N+1</td></tr>
</table>

mcScaleReMask[11:4]			1	N+2
mcScaleReMask[3:0]	csf	mcScaleOffset [14:12]	1	N+3
mcScaleOffset [11:4]			1	N+4
mcScaleOffset [3:0]	mcScaleReMask[11:8]		1	N+5
mcScaleReMask[7:0]			1	N+6
csf	mcScaleOffset [14:8]		1	N+7
mcScaleOffset [7:0]			1	N+8
zero padding			1	N+9
zero padding			1	N+10
zero padding			1	N+11

For a given extLen value, there may be two possible cases for the number of sets of 'mcScaleReMask-csf-mcScaleOffset'. For example, when extLen equals to 4, the number of sets of 'mcScaleReMask-csf-mcScaleOffset' may be either 3 or 4; i.e. both cases will fit within 16bytes (extLen=4). This happens when extLen minus 2 bytes equals to an integer multiple of 3.5 bytes (28bits). In such cases, if the last 28 bits (length of one set) of Section Extension 5 parameters are all set to 0, then O-RU shall consider the smaller number of parameter sets. Otherwise, if they are not all set to 0, O-RU shall consider the larger number of parameter sets.

Table 7-29: Section Format for Section Extension 5 when extLen = 4 (three or four modulation compression parameters may be present)

ef	extType = 0x05		1	Octet N
extLen = 0x04 (4 words)			1	N+1
mcScaleReMask[11:4]			1	N+2
mcScaleReMask[3:0]	csf	mcScaleOffset [14:12]	1	N+3
mcScaleOffset [11:4]			1	N+4
mcScaleOffset [3:0]	mcScaleReMask[11:8]		1	N+5
mcScaleReMask[7:0]			1	N+6
csf	mcScaleOffset [14:8]		1	N+7
mcScaleOffset [7:0]			1	N+8
mcScaleReMask[11:4]			1	N+9
mcScaleReMask[3:0]	csf	mcScaleOffset [14:12]	1	N+10
mcScaleOffset [11:4]			1	N+11
mcScaleOffset [3:0]	mcScaleReMask[11:8](or zero padding)		1	N+12
mcScaleReMask[7:0] (or zero padding)			1	N+13
csf(or zero padding)	mcScaleOffset [14:8] (or zero padding)		1	N+14
mcScaleOffset [7:0] (or zero padding)			1	N+15

7.4.7.5.1 mcScaleReMask (modulation compression power scale RE mask)

Description: This parameter defines the Resource Element (RE) mask to indicate the position of RE with same scaling and modulation type within a PRB. Each bit setting in the mcScaleReMask indicates if the mcScaleOffset and csf fields are applicable to the RE sent in U-Plane messages or not (0=not applicable; 1=applicable). MSB indicates the value for the RE of the lowest frequency in a PRB.

Note that different REs in a PRB may be indicated by different invocations of mcScaleReMask within extension field type 5. If any RE in a PRB is never pointed to by a mcScaleReMask (but other REs in that PRB are), the “missing” RE should be considered to represent not populated REs (e.g. no user data to transmit).

There is a relationship between the mcScaleReMask values and the section’s reMask: no bit in any of the mcScaleReMasks should be set (=1) in a position where the reMask has a zero, and every reMask bit that is set (=1) should have exactly one bit =1 in one of the mcScaleReMasks. If these rules are violated, the O-RU’s reaction is undefined.

Value range: {0000 0000 0000b-1111 1111 1111b}.

Type: unsigned integer (bit mask).

Field length: 12 bits.

Default Value: 1111 1111 1111b (all REs in the block applicable).

7.4.7.5.2 csf (constellation shift flag)

Description: refer to clause 7.4.7.4.1

7.4.7.5.3 mcScaleOffset (scaling value for modulation compression)

Description: This parameter is the scale factor to apply to the unshifted constellation points during decompression. It is a fractional floating-point value having an unsigned but negative 4-bit exponent and an unsigned fractional 11-bit mantissa.

Value range: {0 through +(1-2^{-11}) }.

Type: unsigned integer.

$$"mantissa" = \sum_{k=0}^{10} mcScaleOffset[k] \cdot 2^{k-11}$$

$$"exponent" = \sum_{k=11}^{14} mcScaleOffset[k] \cdot 2^{k-11}$$

"exponent" is the most significant 4 bits of the 15-bit mcScaleOffset field and "mantissa" is the least-significant 11 bits of the mcScaleOffset field. "mcScaleOffset[k] refers to the kth bit of the mcScaleOffset field. Therefore, the actual value of mcScaleOffset is:

$$mcScaleOffset = mantissa \cdot 2^{-exponent}$$

Field length: 15 bits.

7.4.7.6 ExtType=6: Non-contiguous PRB allocation in time and frequency domain

This Section Extension applies only to Section Types 1, 3 and 5.

This Section Extension enables allocation of non-contiguous sets of PRBs (Resource Block Groups, or RBGs) in frequency and time domain. This will reduce significantly the C-Plane overhead when users or channels are allocated with non-contiguous sets of PRBs in time and frequency.

If this Section Extension is present in a section description, then the field startSymbolId in the message header and the fields rb, symInc, and numSymbol in the section description are not used for identification of symbols and PRBs referred by the section description (the values of these parameters can be used by other section descriptions in the message; note that sender may set symInc and numSymbol to any allowed value without restriction to values corresponding to symbols actually referred by the section description with Section Extension =6). Sender shall set rb to the value of zero. Sender shall set startSymbolId to the earliest symbol referred by any of section descriptions in the message, including, but not limited to, section descriptions with Section Extension =6. The earliest symbol referenced by a section description with Section Extension =6 is the earliest symbol selected by a set bit in the symbolMask. That means symbolMask's bit n, such that n < startSymbolId, shall be zero.

Usage of this Section Extension does not affect the operation of user plane. It is noted that a data section in the user plane is not required to contain a contiguous range of PRBs. By invoking the sectionId multiple times in the user plane, either within a single message or in different messages, sets of non-contiguous PRBs can be handled.

Note that utilization of the non-contiguous PRB allocation Section Extension does not put any restriction on utilization of sections with contiguous PRB utilization except for the general rules of utilization of sections e.g. in the context of one eAxC, a resource element shall be referenced by at most one data section description in the control plane.

In case a control plane section is cited multiple times, then all citations shall allocate same set of PRBs and symbols.

Note that when utilizing this Section Extension together with Section Type 3 where freqOffset parameter is present, then freqOffset affects the frequency span for the specific range of PRB numbers.

Table 7-30: Section Format for Section Extension 6 (non-contiguous PRB allocation in time and frequency)

<table>
<tr><td>ef</td><td colspan="3">extType = 0x06</td><td>1</td><td>Octet N</td></tr>
<tr><td colspan="4">extLen = 0x02 (2 words)</td><td>1</td><td>N+1</td></tr>
<tr><td>repetition</td><td colspan="2">rbgSize [2:0]</td><td>rbgMask [27:24]</td><td>1</td><td>N+2</td></tr>
<tr><td colspan="4">rbgMask [23:16]</td><td>1</td><td>N+3</td></tr>
<tr><td colspan="4">rbgMask[15:8]</td><td>1</td><td>N+4</td></tr>
<tr><td colspan="4">rbgMask[7:0]</td><td>1</td><td>N+5</td></tr>
<tr><td colspan="2">priority[1:0]</td><td colspan="2">symbolMask[13:8]</td><td>1</td><td>N+6</td></tr>
<tr><td colspan="4">symbolMask[7:0]</td><td>1</td><td>N+7</td></tr>
</table>

7.4.7.6.1 rbgSize (resource block group size)

Description: This parameter indicates the size in number of PRBs of the resource block groups allocated by the bit mask. The size of the resource block group to be used by the application shall be taken from the mapping table given in the Value range field in the rbgMask description below. See rbgMask for special handling of exceptional resource block groups.

Value range: {000b-111b}.

rbgSize	Number of PRBs in resource block
000b	Reserved
001b	1
010b	2
011b	3
100b	4
101b	6
110b	8
111b	16

Type: unsigned integer.

Field length: 3 bits.

Default Value: 001b.

7.4.7.6.2 rbgMask (resource block group bit mask)

Description: This parameter is a bit mask where each bit indicates whether a corresponding resource block group is allocated. If bit n in the mask is set, then the resource block group n is allocated where n can take values in range [0, lastRbgid] where

lastRbgid = ceiling((numPrbc + (startPrbc mod rbgSize))/ rbgSize) - 1.

The range of PRBs included in a resource block group is given as follows

Resource block group 0 include PRBs with id in the range [startPrbc, startPrbc+f(0)-1].

Resource block group n include PRBs with id in the range [startPrbc + f(0) + (n-1)*f(n), startPrbc + f(0) + n*f(n)-1] where 0 < n < lastRbgid.

Resource block group n = lastRbgid include PRBs with id in the range [startPrbc + numPrbc – f(lastRbgid), startPrbc+ numPrbc -1].

Where f(n) gives the number of PRBs included in resource block group n is given as follows:

f(0) = rbgSize – (startPrbc mod rbgSize)

f(n) = rbgSize, where 0<n<lastRbgid.

if (startPrbc + numPrbc) mod rbgsize > 0 then f(lastRbgid) = (startPrbc + numPrbc) mod rbgSize

else f(lastRbgid) = rbgSize

Only PRBs within the range defined by startPrbc and numPrbc can be allocated i.e., [startPrbc, startPrbc+numPrbc-1]. In case numPrbc is zero the PRB range is all PRBs as defined by the eAxC and startPrbc shall be set to zero. Transmitter shall ensure that rbgMask does not have non-zero bits outside the valid range [0, lastRbgid]; if receiver detects non-zero bits outside the valid range, those shall be ignored.

Transmitter shall ensure that combinations of startPrbc, numPrbc and rbgSize leading to a value of lastRbgid larger than 27 (number of bits available in the rbgMask field - 1) are never used. Such combinations are invalid and if detected by the receiver the section shall be ignored.

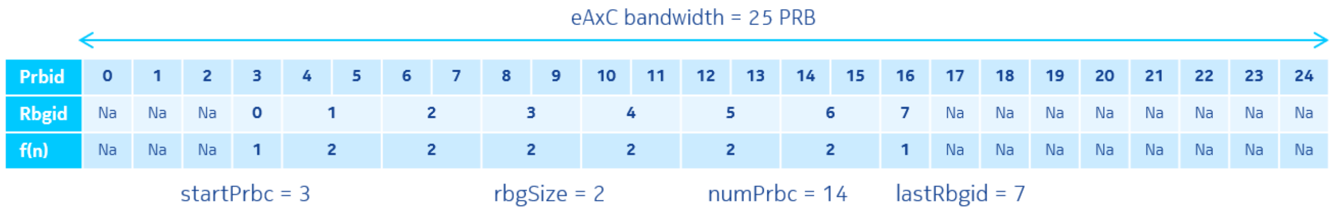
MSB of rbgMask indicates the highest frequency resource block group.

Value range: {0000 0000 0000 0000 0000 0000 0000b - 1111 1111 1111 1111 1111 1111 1111b}.

Type: unsigned integer (bit mask).

Field length: 28 bits.

Default Value: 1111 1111 1111 1111 1111 1111 1111b (all resource block groups allocated).



Prbid	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Rbgid	Na	Na	Na	0	1		2		3		4		5		6		7	Na	Na	Na	Na	Na	Na	Na	Na
f(n)	Na	Na	Na	1	2		2		2		2		2		2		1	Na	Na	Na	Na	Na	Na	Na	Na

Figure 7-9 Resource block group definition example.

7.4.7.6.3 symbolMask (symbol bit mask)

Description: This parameter is a bit mask where each bit indicates whether the rbgMask applies to a given symbol in the slot. If bit n is set then the rbgMask is applied to symbol n, where n has range [0..13]. If no bits are set this implies the rbgMask may be applied to no RBGs in the slot, in other words there are no RE allocations in the slot (or a different data section contains the RE allocation for the slot). If the symbolMask values indicate allocations beyond a slot boundary, such allocations shall be ignored (e.g. when there are fewer than 14 symbols in a slot).

LSB of symbolMask indicates symbol zero (the first symbol to arrive in a slot).

Value range: {00 0000 0000 0000b-11 1111 1111 1111b}.

Type: unsigned integer (bit mask).

Field length: 14 bits.

Default Value: 11 1111 1111 1111b (section repeated in 14 symbols).

7.4.7.6.4 priority

Description: This parameter is used in conjunction with C-Plane and U-Plane coupling via frequency and time with priorities (see clause 7.4.1.2.3). If this coupling method is not used (e.g. O-RU or O-DU do not have corresponding capability) sender shall send value zero. This parameter is used by O-RU to find the section description with the highest priority among section descriptions referencing the same RE that are present in one C-Plane message. O-RU restricts the scope of the search to one C-plane message to avoid beamforming configuration errors that would occur if search would span multiple messages and one of them was lost. If Section Extension #6 is absent, then data section description priority is zero (this corresponds to the default priority). Within one C-plane message a set of priority values is restricted to up to two priority values including value zero i.e. {0, +1} or {0, -1}.

Value -2 is reserved and shall not be used by sender.

Value range: {-1, 0, +1}.

Type: signed integer, 2's complement.

Field length: 2 bits.

Default Value: 0 (normal priority).

7.4.7.6.5 repetition (repetition flag)

Description: This parameter is used to indicate repetition of a highest priority data section inside a C-Plane message. This is an optional field for O-RU i.e., the O-DU shall populate this flag, however, O-RU can choose to process or ignore this flag. This field is applicable only when extension type 6 and extension type 19 is used with coupling via frequency and time with priorities (Optimized) for highest priority section descriptions (since they are repeated across C-Plane messages).

Value range: {0b=no repetition; 1b=repeated highest priority data section in the C-Plane message}.

Type: binary bit.

Field length: 1 bit

7.4.7.7 ExtType=7: eAxC Mask Section Extension

This Section Extension is meant to allow a specific C-Plane message to apply not to just a single eAxC value but to multiple eAxC values. The intended use case applies to Section Type 0, wherein the intention is to allow blanking (designated by Section Type 0) to be indicated for multiple eAxC values e.g., all component carriers. **Table 7-31** shows the Section Extension format.

Table 7-31 : Section Format for Section Extension 7 (eAxC Mask parameter)

ef	extType = 0x07	1	Octet N
extLen = 0x01 (1 word)		1	N+1
eAxCmask[15:8]		1	N+2
eAxCmask[7:0]		1	N+3

When a C-Plane message with this Section Extension is received by an O-RU, the O-RU should react as if multiple C-Plane messages were received with all of the eAxC values permitted by the mask. Note that using the mask, it is possible the O-RU may assume the C-Plane message content applies to eAxC values that are not intended and are not used (e.g. if two bits of the mask are set to zero to indicate the message applies to all 3 sectors, there will be a non-existent fourth sector that will be implied too). In this case there will be no U-Plane data specified for the spurious eAxC value(s) and the O-RU should not consider this as an error.

Support of this Section Extension is optional: if an O-RU reports a capability excluding support of this Section Extension, then the O-DU shall not employ it, and if the O-DU does not support this Section Extension, the same effect can be obtained using multiple single-eAxC C-Plane messages so no functionality is lost (though the C-Plane will be less efficient).

7.4.7.7.1 eAxCmask (eAxC Mask)

Description: This binary mask indicates which eAxC values the C-Plane message applies to. A “0” bit in the mask means the specified eAxC bit is “don’t care” while a “1” bit in the mask means the specified eAxC bit should be considered. An all-zero eAxCmask means “use this message for all eAxC values” while an all-ones eAxCmask means “use this message for only the designated eAxC value”. In the former case (“use this message for all eAxC values”), the actual eAxC value in the C-Plane message header could be any value because the message would apply to all eAxC values regardless of the actual value in the C-Plane message. In the latter case (“use this message for only the designated eAxC value”) this Section Extension adds no value because that is the normal operation.

The intended use of the mask is to indicate which sub-field of the eAxC the message applies to. For example, if a single sector single band of the O-RU will be blanked for all component carriers, the RU-Port-ID and CC-ID sub-field portions of the eAxCmask would be all zeros (“use this message for all layers/spatial streams and all component carriers”) while the other subfield portions of the eAxCmask would be all ones (“use this message for the specific band, sector, and DU_Port_ID values provided in the eAxC”). In this way a Section Type 0 message could indicate the blanking of all component-carriers in a band-sector in a single C-Plane message.

Value range: {0000 0000 0000 0000b-1111 1111 1111 1111b}.

Type: unsigned integer (bit mask).

Field length: 16 bits.

Default Value: 1111 1111 1111 1111b (all bits of eAxC are considered).

7.4.7.8 ExtType=8: Regularization factor

This Section Extension is for regularizationFactor which is used to generate MU beamforming weight for the scheduled UEs in a slot. regularizationFactor is the noise variance used for MMSE (Minimum Mean Square Error) computation of generating MU BFW. The intended use case applies to Section Type 5 for sending regularization factor instead of sending regularization factor over Section Type 6.

Support of this Section Extension is mandatory for O-RUs supporting Section Type 5 in this version of the specification. If an O-RU reports a capability excluding support of this Section Extension (O-RUs supporting previous version of specification or O-RUs not supporting Section Type 5), then the O-DU shall not employ it, and if the O-RU reports a capability including support of this Section Extension then O-DU supporting this version of specification shall employ this Section Extension instead Section Type 6 to convey regularization factor.

See Annex J for further description of the regularization factor’s use in UL and DL.

The following table shows the format of this Section Extension.

Table 7-32 : Section Format for Section Extension 8 (Regularization factor parameter)

ef	extType = 0x08	1	Octet N
extLen = 0x01 (1 word)		1	N+1
regularizationFactor[15:8]		1	N+2
regularizationFactor[7:0]		1	N+3

7.4.7.8.1 regularizationFactor (regularization Factor)

Description: This parameter provides a regularization factor to support MMSE operation when UE is scheduled, so related to Section Type 5.

Value range: {0000 0000 0000 0000b-1111 1111 1111 1111b}

Type: unsigned integer.

Field length: 16 bits.

7.4.7.9 ExtType=9: Dynamic Spectrum Sharing parameters

This Section Extension applies to all Section Types.

The following table shows the format of this Section Extension.

Table 7-33 : Section Format for Section Extension 9 (DSS parameter)

ef	extType = 0x09	1	Octet N
extLen = 0x01 (1 word)		1	N+1
technology[7:0]		1	N+2
reserved		1	N+3

This Section Extension is needed to support Dynamic Spectrum Sharing (DSS) operation. O-RU should know if the information transmitted to the eCPRI interface is LTE or NR in real time. If “technology” = 0, it means LTE support. If “technology” = 1, it means NR support. O-RU shall consider that the C-plane message received is for NR, if Section Extension =9 is not present in the C-plane message for a carrier configured as DSS via the M-plane. If Section Extension =9 is received in a C-plane message for a carrier configured as LTE or NR (not DSS), the O-RU shall ignore the Section Extension 9 and may optionally log the error. Please see clause 7**.3.6** for more details on methods for DSS support.

7.4.7.9.1 technology (interface name)

Description: This parameter indicates if the data transmitted through eCPRI interface is LTE or NR in real time. In this way, O-RU knows the data format transmitted from O-DU and could decode the associated user data.

Value range: {0000 0000b-1111 1111b} :

- 0000 0000b : LTE support
- 0000 0001b : NR support
- 0000 0010b – 1111 1111b : reserved.

Type: unsigned integer.

Field length: 8 bits.

Default Value: 0000 0000b (LTE).

7.4.7.10 ExtType=10: Section description for group configuration of multiple ports

This Section Extension applies to Section Types 1, 3 and 5. C-plane section information for the multiple ports (i.e., layers or Tx/Rx paths) may be the same except beam IDs or UE IDs in most cell scheduling case. When the multiple ports share the common section information within O-RU, C-plane sections to be sent via the corresponding ports are merged into a C-plane section via a specified port using this Section Extension. M-plane pre-configures to group the ports to be merged and to specify the port to represent them. The beam ID in the section of the representative port can be considered an index which indicates a set of multiple beam vectors (beam IDs) each of which applies to the corresponding port (layer or Tx/Rx path).

Table 7-34 : Section Extension =10 for beamGroupType = 00b or 01b

ef	extType = 0x0A	1	Octet N
extLen = 0x01 (1 word)		1	N+1
beamGroupType	numPortc	1	N+2
reserved		1	N+3

Table 7-35 : Section Extension =10 for beamGroupType = 10b

ef	extType = 0x0A	1	Octet N
extLen		1	N+1
beamGroupType	numPortc	1	N+2
reserved	2nd port beamID[14:8] (or ueID[14:8])	1	N+3
2nd port beamID[7:0] (or ueID[7:0])		1	N+4
reserved	3rd port beamID[14:8] (or ueID[14:8])	1	N+5
3rd port beamID[7:0] (or ueID[7:0])		1	N+6
reserved	…	1	var
…		1	var
reserved	(numPortc+1)th port beamID[14:8] (or ueID[14:8])	1	var
(numPortc+1)th port beamID[7:0] (or ueID[7:0])		1	var
filler to ensure 4-byte boundary		var	var

7.4.7.10.1 beamGroupType

Description: This parameter indicates the type of beam grouping.

Value range: {00b - 11b}.

- **00b** (common beam): the beamID in the section header is used as a common beam ID for all the ‘numPortc’ ports which are grouped by M-plane. This type is not used for Section Type 5, and extLen = 0x01.

- **01b** (beam matrix indication): the consecutive ‘numPortc’ beamIDs subsequent to the beamID in the section heade r apply to the ‘numPortc’ ports. This type is not used for Section Type 5, and extLen = 0x01.

- **10b** (beam vector listing): the beamIDs listed in the Section Extension apply to the ‘numPortc’ ports. The Section E xtension should include ‘numPortc’ beam IDs or ueIDs.

- **11b** : reserved

Type: unsigned integer.

Field length: 2 bits.

Default Value: 00b (common beam).

7.4.7.10.2 numPortc

Description: This parameter indicates the number of eAxC ports indicated by the Section Extension. It can cover up to 64 ports.

Value range: {000000b-111111b}. 00 0000b expresses 64 ports.

Type: unsigned integer.

Field length: 6 bits.

Default Value: 000000b.

7.4.7.10.3 Interaction with other Section Extensions

Section Extension =10 can be used in all situations when same information is present in section headers and in Section Extension headers across multiple eAxC IDs.

Table 7-36 : Section Extension =10 Interactions with other Section Extensions

Section Extension	Title	Interaction with existing Section Extensions
1	Beamforming Weights	SE1 can be used with Section Extension =10 only if the Beamforming Weights transferred using Section Extension =1 are the same for all streams (or layers). In general, the BF weight vector will be different per data layer (eAxC) and Section Extension =10 cannot be used in conjunction. Once the Beamforming Weights are downloaded, then if the O-DU uses same BF vector by addressing Beam Id for a number of slots after the BF Weight is downloaded, then in that case, O-DU can use Section Extension 10 to combine multiple C-plane messages to one single C-plane message using the representative eAxC id.
2	Beamforming Attributes	Section Extension =2 can be used with Section Extension =10 only if the Beamforming Attribute transferred using Section Extension =2is same for all streams (or layers), otherwise, Section Extension =2 cannot be used with Section Extension =10. If O-DU uses beam attributes downloaded using Section Extension =2 later using beam id, then Section Extension =10 can be used to combine multiple C-plane messages to one C-plane message.
3	DL Precoding	This Section Extension has different format for different layers (first layer has one set of parameters and rest of the layers have different set of parameters). So this Section Extension =3 cannot be used with the Section Extension =10. There is a possibility of using the Section Extension =10 with layers other than first layer (if the codebook index is same for other layers) and also combining Section Extension =3 for all layers in future with some changes. This is FSS.
4	Modulation Compression	Section Extension =4 can be used with Section Extension =10 (see 5.5.9).
5	Modulation Compression (Additional)	Section Extension =5 can be used with Section Extension =10 (see 5.5.9).
6	Non-contiguous PRB	Section Extension =6 can be used with the Section Extension =10, if the section information of non-contiguous PRB allocation has same pattern for each spatial stream or data layer (eAxC).
7	eAxC Mask	Section Extension =7 cannot be used with Section Extension =10 because the use of Section Extension =10 includes the functionality of this Section Extension =7.

8	Regularizatoin Factor	Section Extension =8 can be used with Section Extension =10 only if the regularization parameter transferred using Section Extension =8 is the same for all streams (or layers), otherwise Section Extension =8 cannot be used with Section Extension =10
9	DSS Parameters	Section Extension =9 can be used with Section Extension =10 because all the spatial streams or data layers (eAxCs) grouped by Section Extension =10 are used for the same technology.
11	Flexible Beamforming Weights	Section Extension =11 can be used with Section Extension =10 only if the Beamforming Weights transferred using Section Extension =11are same for all streams (or layers). In general, the BF weight vector will be different per data layer (eAxC) and Section Extension =11 cannot be used in conjunction. Once the Beamforming Weights are downloaded, then if the O-DU uses same BF vector by addressing Beam Id for a number of slots after the BF Weights are downloaded, then in that case, the O-DU can use Section Extension =11 to combine multiple C-plane messages to one single C-plane message using the representative eAxC id.
14	Null-layer Info	If this Section Extension can be used with Section Extension = 10, extLen in this Section Extension can be enlarged as much as the total number of ueIds transmitted by both section header and Section Extension = 10, then this Section Extension indicates the nulling information for whole of the respective scheduled layers.

7.4.7.11 ExtType=11: Flexible Beamforming Weights Extension Type

This section applies to the flexible sending of beamforming weights from the O-DU to the O-RU. This enables the O-DU to provide different beamforming weights for different PRBs within one section to facilitate, e.g. zero-forcing precoding. The O-DU provides the numBundPrb parameter, which informs the O-RU how many PRBs are bundled together and share the same beamforming weights. This Section Extension applies only to Section Types 1 and 3.

Table 7-37: Extension Type 11 Data Format (when disableBFWs = 0)

ef	extType = 0x0B		1	Octet N
extLen[15:0]			2	Octet N+1
disableBFWs	RAD	reserved (6 bits)	1	Octet N+3
numBundPrb[7:0]			1	Octet N+4
bfwCompHdr[7:0]			1	Octet N+5
bfwCompParam (for PRB bundle 0)			var	
reserved	beamId[14:8] (for PRB bundle 0)		1	
beamId[7:0] (for PRB bundle 0)			1	
bfwI (for TRX 0 and PRB bundle 0)			var	
bfwQ (for TRX0 and PRB bundle 0)			var	
remaining beamforming weights bfwI and bfwQ up to L TRXs and PRB bundle 0			var	
bfwCompParam (for PRB bundle 1)			var	
reserved	beamId[14:8] (for PRB bundle 1)		1	
beamId[7:0] (for PRB bundle 1)			1	
bfwI (for TRX 0 and PRB bundle 1)			var	
bfwQ (for TRX0 and PRB bundle 1)			var	
remaining beamforming weights bfwI and bfwQ up to L TRXs and PRB bundle 1			var	
…				
bfwCompParam (for last PRB bundle)			var	
reserved	beamId[14:8] (for last PRB bundle)		1	
beamId[7:0] (for last PRB bundle)			1	
bfwI (for TRX 0 and last PRB bundle)			var	
bfwQ (for TRX0 and last PRB bundle)			var	
remaining beamforming weights bfwI and bfwQ up to L TRXs and last PRB bundle			var	
zero pad to 4-byte boundary			var	

7.4.7.11.1 bfwCompHdr (beamforming weight compression header)

Description: refer to clause 7.4.7.1.1

Note that this parameter defines the compression method and IQ bit width for the beamforming weights in the specific section in the C-Plane message. For the block compression methods, the block size is the vector of beamforming weights for a specific PRB bundle.

7.4.7.11.2 bfwCompParam for PRB bundle x (beamforming weight compression parameter)

Description: refer to clause 7.4.7.1.2

Note that this parameter applies to the following vector of beamforming weights for a specific PRB bundle (i.e., bundle x).

7.4.7.11.3 numBundPrb (Number of bundled PRBs per beamforming weights)

Description: This parameter is the number of bundled PRBs per beamforming weight sets. The number of beamforming weight sets per TRX in Section Extension 11 should be equal to the total number of PRBs selected by section description in the C-plane message (using startPrbc and numPrbc and other parameters present in other Section Extensions present in the section description) divided by numBundPrb. When the division outcome is fractional value, one additional beamforming weight for each respective TRX is included to cover the orphan PRBs. The value zero is reserved.

Value range: {all zeros – all ones}.

Type: unsigned integer.

Field length: 8 bits.

7.4.7.11.4 bfwI (beamforming weight in-phase value)

Description: This parameter is the In-phase beamforming weight value. The total number of weights in the section depends on the number TRX at the RU and the number of bundled PRBs per beamforming weight

Value range: {all zeros – all ones}.

Type: signed integer.

Field length: 1-16 bits.

7.4.7.11.5 bfwQ (beamforming weight quadrature value)

Description: This parameter is the Quadrature beamforming weight value. The total number of weights in the section depends on the number TRX at the RU and the number of bundled PRBs per beamforming weight

Value range: {all zeros – all ones}.

Type: signed integer.

Field length: 1-16 bits.

7.4.7.11.6 disableBFWs (disable beamforming weights)

Description: This parameter is used to indicate if beamforming weights under Section Extension are disabled or not. The default is that the beamforming weights exist since this is the main objective of the Section Extension. However, these can be disabled by setting this parameter to convey only the beamIds. In this case, all bfwI, bfwQ, and bfwCompParam fields as well as the bfwCompHdr field don’t exist in the Section Extension (see **Table 7-38** below).

Value range: {0b=beamforming weights under Section Extension exist; 1b=don’t exist}.

Type: binary bit.

Field length: 1 bit.

Default Value: 0b (beamforming weights under Section Extension exist)

Table 7-38: Extension Type 11 Data Format (when disableBFWs = 1)

ef	extType = 0x0B		1	Octet N
extLen[15:0]			2	Octet N+1
disableBFWs	RAD	reserved (6 bits)	1	Octet N+3
numBundPrb[7:0]			1	Octet N+4
reserved	beamId[14:8] (for PRB bundle 0)		1	
beamId[7:0] (for PRB bundle 0)			1	
reserved	beamId[14:8] (for PRB bundle 1)		1	
beamId[7:0] (for PRB bundle 1)			1	
…				
reserved	beamId[14:8] (for last PRB bundle)		1	
beamId[7:0] (for last PRB bundle)			1	
zero pad to 4-byte boundary			var	

7.4.7.11.7 RAD (Reset After PRB Discontinuity)

Description: This parameter is used when Section Extension =11 is used in conjunction with Section Extension allowing non-contiguous frequency allocation (Section Extensions =6, 12 and 13). In regular cases, where the section parameters refer to a continuous set of PRBs, the default value should be used, which is RAD = 1. When Section Extension allowing non-contiguous frequency allocation is used with Section Extension =11, this parameter indicates whether the PRB bundle boundary is reset after discontinuity in the PRB allocation (RAD = 1), or the PRB bundle boundaries are mapped to the PRBs regardless of the discontinuity (see examples in clause 7.5.4.1). Note if this extension is used with Section Extension =13 the discontinuity shall be interpreted as discontinuity in set of frequency ranges from all hops combined over time (see examples in clause 7.5.4.3).

Value range: {0b = No reset; 1b = reset)

Type: binary bit

Field length: 1 bit.

Default Value: 1b (Reset after PRB discontinuity)

7.4.7.12 ExtType=12: Non-Contiguous PRB Allocation with Frequency Ranges

This Section Extension applies only to Section Types 1, 3 and 5. It cannot be used with Section Extension 6 in the same section description.

This Section Extension enables allocation of non-contiguous sets of PRBs (Resource Block Groups, or RBGs) in time domain and frequency domain. This extension reduces the C-Plane overhead when users or channels are allocated with non-contiguous sets of PRBs in time or frequency. This extension is more space-efficient than Section Extension 6 if the allocation is continuous in frequency or extends over a frequency span wider than the range defined by rbgMask size and rbgMaskSize value.

Table 7-39: Section Format for Section Extension 12 (Non-Contiguous PRB Allocation with Frequency Ranges)

0 (msb)	1	2	3	4	5	6	7 (lsb)	# of bytes	
ef	extType[6:0] = 0x0C							1	Octet N
extLen (variable)								1	N+1
priority[1:0]		symbolMask[13:8]						1	N+2
symbolMask[7:0]								1	N+3
offStartPrb(1)								1	variable
numPrb(1)								1	variable
offStartPrb(2)								1	variable
numPrb(2)								1	variable

…		
offStartPrb(R-2)	1	variable
numPrb(R-2)	1	variable
offStartPrb(R-1)	1	variable
numPrb(R-1)	1	variable

This extension consists of a fixed part (fields priority and symbolMask) and a variable size part (pairs of offStartPrb(r) and numPrb(r) for r = 1 … R-1). Note that variable part may be empty allowing to use priority and symbolMask without overhead.

If this Section Extension is present in a section description, then the section description is applied to symbols identified by symbolMask and one or more frequency ranges calculated from startPrbc and numPrbc from the section description and offStartPrb(r) and numPrb(r) pairs from this extension.

If this Section Extension is present in a section description, then the field startSymbolId in the message header and the fields rb, symInc, and numSymbol in the section description are not used for identification of symbols and PRBs referred by the section description (the values of these parameters can be used by other section descriptions in the message; note that sender may set symInc and numSymbol to any allowed value without restriction to values corresponding to symbols actually referred by the section description with Section Extension =12). Sender shall set rb to the value of zero. Sender shall set startSymbolId to the earliest symbol referred by any of section descriptions in the message, including, but not limited to, section descriptions with Section Extension =12. The earliest symbol referenced by a section description with Section Extension =12 is the earliest symbol selected by a set bit in the symbolMask. That means symbolMask's bit n, such that n < startSymbolId, shall be zero.

If this Section Extension is present in a section description, then the section description is applied to one or more PRB ranges. Specifically, the description is applied to R spectrum fragments identified by pairs startPrb(r) and numPrb(r) for r = 0, …, R-1 where startPrb(r) values are calculated by recurrence:

> startPrb(0) and numPrb(0) are values of section description's parameter startPrbc and numPrbc; values of numPrb(1), numPrb(2), … numPrb(R-1) are from the variable part of section description.
>
> startPrb(r) = startPrb(r-1) + numPrb(r-1) + offStartPrb(r) for r = 1, ..., R-1

The description applies to PRB #p if startPrb(r) ≤ p < startPrb(r) + numPrb(r) for any r = 0, ..., R-1.

Number of frequency ranges R is derived from extLen: R = (extLen – 1) * 2 + 1. Note that empty PRB ranges are allowed e.g. numPrb(r) = 0 can be used to shift the following PRB range beyond limit of 8-bit offset. Similarly, offStartPrb(r) = 0 can be used to concatenate two PRB ranges and effectively extend range width beyond the limit of 8-bit numPrb(r). If the last pair in the extension are present (due to extension size alignment to multiple of 4 bytes) but not used, then they shall have offStartPrb(R-1) = 0 and numPrb(R-1) = 0.

Usage of this Section Extension does not affect the operation of user plane. It is noted that a data section in the user plane is not required to contain a contiguous range of PRBs. By invoking the sectionId multiple times in the user plane, either within a single message or in different messages, sets of non-contiguous PRBs can be handled.

Note that utilization of the non-contiguous PRB allocation Section Extension does not put any restriction on utilization of sections with contiguous PRB utilization except for the general rules of utilization of sections e.g. in context of one eAxC a resource element shall be referenced by at most one data section description in the control plane.

In case a control plane section is cited multiple times with C-plane and U-plane coupling via sectionId then all citations shall allocate same set of PRBs and symbols.

Note that when utilizing this Section Extension together with Section Type 3 where freqOffset parameter is present, then freqOffset affects the frequency span for the specific range of PRB numbers.

7.4.7.12.1 priority (priority of section description)

Description: This parameter has same format and semantics as priority parameter in Section Extension 6 (see clause 7.4.7.6.4).

7.4.7.12.2 symbolMask (symbol bit mask)

Description: This parameter has same format and semantics as symbolMask parameter in Section Extension 6 (see clause 7.4.7.6.3).

7.4.7.12.3 offStartPrb(r) (offset of PRB range start)

Description: This parameter indicates the offset to start of the r-th PRB range for r=1, …, R-1.

Value range: 0 ... 255.

Type: unsigned integer.

Field length: 8 bits.

7.4.7.12.4 numPrb(r) (number of PRBs in PRB range)

Description: This parameter indicates the number of PRBs in the r-th PRB range for r=1, …, R-1.

Value range: 0 ... 255.

Type: unsigned integer.

Field length: 8 bits.

7.4.7.13 ExtType=13: PRB Allocation with Frequency Hopping

This Section Extension applies only to Section Types 1, 3 and 5.

This Section Extension allows to describe two or more PRB allocations starting at different symbols and different PRB. It is intended to be used for allocations with intra-slot frequency hopping. This extension significantly reduces the C-Plane overhead when users or channels are allocated with intra-slot frequency hopping. This extension is more space-efficient than the use of several section descriptions.

Table 7-40: Section Format for Section Extension 13 (PRB Allocation with Frequency Hopping)

0 (msb)	1	2	3	4	5	6	7 (lsb)	# of bytes	
ef	extType[6:0] = 0x0D							1	Octet N
extLen (variable)								1	N+1
reserved[1:0]		nextSymbolId(1)[3:0]				nextStartPrbc(1)[9:8]		1	N+2
nextStartPrbc(1)[7:0]								1	N+3
…									
reserved[1:0]		nextSymbolId(R-2)[3:0]				nextStartPrbc(R-2)[9:8]		1	variable
nextStartPrbc(R-2)[7:0]								1	variable
reserved[1:0]		nextSymbolId(R-1)[3:0]				nextStartPrbc(R-1)[9:8]		1	variable
nextStartPrbc(R-1)[7:0]								1	variable

This extension is of variable size and conveys list of pairs nextSymbolId(n) and nextStartPrbc(n) for n = 1,…,R-1.

If this Section Extension is present in a section description, then the section description and all other Section Extensions included in it, are interpreted by O-RU as if startPrbc value was changed at specified symbols given in nextSymbolId(n) to values of nextStartPrbc(n) provided in the extension. Note that the value of numPrbc from the section description applies to all frequency hops.

Specifically, the description and extensions are interpreted as if startPrbc value was as in nextStartPrbc(n) for symbols s, where nextSymbolId(n) ≤ s < nextSymbolId(n+1) for n=1 to R-1 and value of nextSymbolId(R) is a value greater by 1 than number of the last symbol addressed by the description. Note nextSymbolId(R) is guard value and does not have to identify actual symbol. If the section description includes a Section Extension that provides symbolMask parameters (Section Extension #6, #12 and #19) then nextSymbolId(R) is derived from symbolMask. Otherwise, nextSymbolId(R) is derived from startSymbolId from the message header and symInc and numSymbol parameters in the section description and in section descriptions preceding it in the message. Note that startPrbc from the section description applies to symbols s0 ≤ s < nextSymbolId(1) where s0 is the first symbol addressed by the section description (s0 is determined by symbolMask if Section Extension #6, #12 or #19 is present; otherwise s0 is determined by startSymbolId from the message header and symInc and numSymbol parameters in the section description and in section descriptions preceding it in the message). The set of symbols where startPrbc or nextStartPrbc(n) applies may be further restricted by symbolMask if Section Extension #6, #12 or #19 is also present.

Number of frequency hops R is derived from extLen: R = extLen * 2. If the last pair in the extension are present (due to extension size alignment to multiple of 4 bytes) but not used, then they shall have values as in the preceding pair (e.g. if pair R-1 is not used then nextSymbolId(R-1) and nextStartPrbc(R-1) shall be set to nextSymbolId(R-2) and nextStartPrbc(R-2) respectively).

7.4.7.13.1 nextSymbolId(n) (offset of PRB range start)

Description: This parameter indicates the symbol at which n-th frequency hop occurs for n=1, …, R-1. The value shall correspond to one of symbols addressed by the section description. Note that:

- if Section Extension #6, #12 and #19 are not present in the section description then set of symbols addressed by the description is determined by startSymbolId from the message header and symInc and numSymbol parameters in the section description and in section descriptions preceding it in the message
- if Section Extension #6, #12 or #19 is present in the section description then set of symbols addressed by the description is determined by symbolMask from Section Extension #6, #12 or #19.

Sender shall ensure values are ordered in increasing order i.e. nextSymbolId(n+1) ≥ nextSymbolId(n).

Sender shall set nextSymbolId(n) to value of nextSymbolId(n-1) and set nextStartPrbc(n) to value of nextStartPrbc(n-1) if pair (nextSymbolId(n), nextStartPrbc(n)) is used for padding.

Value range: {0000b-1111b}.

Type: unsigned integer.

Field length: 4 bits.

7.4.7.13.2 nextStartPrbc(n) (number of PRBs in PRB range)

Description: This parameter indicates the value to be used instead of startPrbc for the n-th frequency hop for n=1, …, R-1.

Sender shall set nextSymbolId(n) to value of nextSymbolId(n-1) and set nextStartPrbc(n) to value of nextStartPrbc(n-1) if pair (nextSymbolId(n), nextStartPrbc(n)) is used for padding.

Value range: {00 0000 0000b-11 1111 1111b}.

Type: unsigned integer.

Field length: 10 bits.

7.4.7.14 ExtType= 14: Nulling-layer Info. for ueId-based beamforming

This Section Extension applies to Section Types 5. This enables the O-DU to provide layer-by-layer indication, which denotes that the corresponding ueId is for nulling-layer indication.

The following table shows the format of this Section Extension.

Table 7-41: Extension Type 14

ef	extType = 0x0E	1	Octet N
extLen = 0x01 (1 word)		1	N+1
nullLayerInd		1	N+2
reserved		1	N+3

O-RU generates beamforming weights considering all the UeIds, but once the beamforming weight is generated, O-RU shall set to zero weights for the layer(s) corresponding to those UeIds for which nullLayerInd is set to 0000 0001b.

7.4.7.14.1 nullLayerInd (null layer indication)

Description: This parameter indicates whether the corresponding layer is nulling-layer or not.

Value range: {0000 0000b - 0000 0001b}.

The following mapping shall be used:

0000 0000b = the corresponding ueId is given for actually scheduled layer, no specific operation based on this field is performed by O-RU.

0000 0001b = The corresponding ueId is given for making beam-nulling dimension, so this layer shall be nulled. No user data is transmitted for the layer(s) corresponding to the UeId for which nullLayerInd is set to 0000 0001b, i.e. beamforming weights corresponding to the layer(s) shall be set to zero.

Type: unsigned integer.

Field length: 8 bits.

7.4.7.15 ExtType= 15: Mixed-numerology Info. for ueId-based beamforming

This Section Extension applies to Section Types 5 and 6. When this Section Extension is applied to Section Type 6, the value of FFT type in frameStructure and cpLength can be set as ‘0’.

The following table shows the format of this Section Extension.

Table 7-42: Extension Type 15

ef	extType = 0x0F	1	Octet N
extLen = 0x02 (2 words)		1	N+1
frameStructure		1	N+2
freqOffset		3	N+3
cpLength		2	N+6

7.4.7.15.1 frameSturcture (frame structure)

See clause 7.4.4.13 for the description of this parameter.

7.4.7.15.2 freqOffset (frequency offset)

See clause 7.4.5.11 for the description of this parameter.

7.4.7.15.3 cpLength (cyclic prefix length)

See clause 7.4.4.14 for the description of this parameter.

7.4.7.16 ExtType=16: Section description for antenna mapping in UE channel information based UL beamforming

This Section Extension applies to Section Type 5. The Section Extension includes bitmask per Rx endpoint to indicate the antennas to be pre-combined into the Rx endpoint (=eAxC). It can be used together with Section Extension 10. In this case, it has a list of the bitmasks as many the Rx endpoints as used for the Section Extension 10.

Table 7-43: Section Extension 16 w/o Section Extension 10

ef	extType = 0x10	1	Octet N
extLen = 0x03 (3 words)		1	N+1
antMask [63:0]		8	N+2
filler to ensure 4-byte boundary		var	N+10

Table 7-44: Section Extension 16 with Section Extension 10

ef	extType = 0x10	1	Octet N
extLen [7:0]		1	N+1
antMask (1st Rx eAxC) [63:0]		8	N+2
antMask (2nd Rx eAxC) [63:0]		8	N+10
…		…	…
antMask (16th Rx eAxC) [63:0]		8	N+122
filler to ensure 4-byte boundary		var	N+130

7.4.7.16.1 antMask

Description: This parameter indicates the indices of antennas to be pre-combined per Rx endpoint.

Value range: {0x0000 0000 0000 0000 – 0xFFFF FFFF FFFF FFFF}.

Each bit indicates whether the antenna of the corresponding digit is pre-combined or not. The maximum number of antennas is 64.

Type: unsigned integer.

Field length: 64 bits.

Default Value: 0x0000000000000000

7.4.7.17 ExtType= 17: Section description for indication of user port group

This Section Extension applies to Section Extension 10 for beamGroupType=10b following Section Type 5 and cannot be used in standalone manner. This Section Extension provides the number ueIDs of the users scheduled in preceding Section Type and Section Extension messages. A user may have more than one ueIDs (i.e. more than one channel information, for e.g. if the UE supports Transmit Antenna Switching feature for SRS transmission; by which O-DU can obtain different channel information corresponding to each of the transmit antennas). When this Section Extension is used, two restrictions apply: First, ueIDs of each user are consecutive by exploiting the three reserved bits of ueID[2:0]. The three reserved bits mean that the maximum number of ueIDs per user is assumed to be 8. Second, in Section Extension 10, ueIDs whose three reserved bits are all zero are configured repeatedly as many times as the number of layers scheduled for the corresponding user. Therefore, the preceding Section Type and extension messages implicitly provide the number of scheduled users(=number of different ueIDs) and number of layers of each user(=number of same ueIDs). Finally, the number of ueIDs associated with each user is informed via the information provided in this Section Extension.

Table 7-45: Section Extension 17

ef	extType = 0x11	1	Octet N
extLen		1	N+1
numUeID of 1st user	numUeID of 2nd user	var	N+2
…			
..	numUeID of last user		
filler to ensure 4-byte boundary			

7.4.7.17.1 numUeID

Description: This parameter indicates the number of ueIDs per user

Value range: {0001b - 1000b}. {1001b - 1111b} are reserved.

Type: unsigned integer.

Field length: 4 bits.

Default Value: 100b (4 ueIDs per user).

7.4.7.18 ExtType=18: Section description for Uplink Transmission Management

This Section Extension applies to sending of transmission windows information for management of the transmission of uplink user data from the O-RU. With the two parameters included in this Section Extension the O-RU shall transmit the corresponding user data within that time window. The data shall be sent in normal transmission mode or be uniformly distributed in time depending on value of toT parameter.

Table 7-46: Section Extension 18

ef	extType = 0x12	1	Octet N
extLen		1	N+1
transmissionWindowOffset[15;8]		1	N+2
transmissionWindowOffset[7:0]		1	N+3
Reserved	transmissionWindowSize[13:8]	1	N+4
transmissionWindowSize[7:0]		1	N+5
Reserved	toT	1	N+6
zero pad to 4-byte boundary		1	N+7

7.4.7.18.1 transmissionWindowOffset

Description: This parameter indicates the start of the transmission window as an offset to when the transmission window would have been without this parameter, i.e. (Ta3_max-Ta3_min). The resolution of the parameter is symbols, where the length in time of a symbol is either as determined via M-plane (when used with message Type 1 and 5) or according to information in parameter frameStructure (when used with message Type 3) and assuming normal cyclic prefix.

If numSymbol in the section header is > 1 then the number of different transmission windows will be equal to numSymbol. The start of each transmission window follows the same procedure as described above for every symbol.

See clause 4.5.2.1 for more detailed information about overlapping transmission windows.

If message is used for an eAxC configured for delay-managed traffic, then transmissionWindowOffset value in combination with the value of transmissionWindowSize shall not result in calculated buffering load that would exceed the O-RU buffering capacity. See clause 4.5.4 for more details.

If message is used for an eAxC configured for non-delay-managed traffic, then transmissionWindowOffset value can exceed the O-RU buffering capacity. See clause 4.5.4 for more details.

Value range: {0 – 65535}.

Type: unsigned integer.

Field length: 16 bits.

Default Value: 0

7.4.7.18.2 transmissionWindowSize

Description: This parameter indicates the size of the transmission window in resolution µs. If window size is set to a value smaller than (ta3_max – ta3_min) the O-RU will if possible, transmit the requested data reliably during that window size. If not possible due to for instance lack of transmission capability on link caused by other higher prioritized traffic, then the O-RU will use a default transmission window size of (ta3_max – ta3_min).

Maximum window size is 10000 µs = 10 ms.

See clause 4.5.2.1 for more detailed information about overlapping transmission windows.

If message is used for an eAxC configured for delay-managed traffic, then transmissionWindowSize value in combination with the value of transmissionWindowOffset shall not result in calculated buffering load that would exceed the O-RU buffering capacity.

If message is used for an eAxC configured for non-delay-managed traffic, then transmissionWindowSize which is not affecting O-RU operation shall be set to 0 (zero). See clause 4.5.4 for more details.

Value range: {0x0 – 0x2710}

Type: unsigned integer.

Field length: 14 bits.

Default Value: 0x00.

7.4.7.18.3 Type of Transmission (toT)

Description: This parameter indicates to the O-RU if the associated user plane data should be sent in normal transmission mode or be transmitted uniformly in time over the transmission window. The O-RU will indicate via the M-Plane if it supports the possibility to change the type of transmission. If the O-RU does not support this feature, the O-RU will ignore this parameter.

See clause 4.5.2 for more detailed information about uniformly distribution of data.

The value of Type of Transmission shall be set to the same value for all sections that have Section Extension 18 within one C-Plane message.

Value range: {00b – 11b}.
00b = normal transmission mode, data can be distributed in any way the O-RU is implemented to transmit data
01b = uniformly distributed over the transmission window.
10b = Reserved
11b = Reserved

Type: bits.

Field length: 2 bits.

Default Value: 00b

7.4.7.18.4 Interaction with other Section Extensions

The Section Extension =18 is used when the UL traffic needs to be managed e.g., avoid peaks in the traffic on shared links between an O-DU and multiple O-RUs.

The table below lists how Section Extension 18 is used in combination with other Section Extensions.

Section Extension	Title	Interaction with section extension 18
1	Beamforming weights	No special handling needed
2	Beamforming attributes	No special handling needed
3	DL Precoding configuration parameters and indications	No special handling needed
4	Modulation compression parameters	No special handling needed
5	Modulation compression additional scaling parameters	No special handling needed
6	Non-contiguous PRB allocation	No special handling needed
7	Multiple-eAxC designation	The usage of this Section Extension has the effect that more than one eAxC is targeted with one C-plane message. For UL this means that the same transmission time and window size will be used for all targeted eAxCs.
8	Regularization factor	No special handling needed
9	Dynamic spectrum sharing parameters	No special handling needed
10	Multiple ports grouping	The usage of this Section Extension has the effect that more than one eAxC is targeted with one C-plane message. For UL this means that the same transmission time and window size will be used for all eAxCs.
11	Flexible BF weights	No special handling needed

Section Extension	Title	Interaction with section extension 18
12	Non-contiguous PRB allocation with frequency ranges	No special handling needed
13	Frequency hopping	No special handling needed
14	Null-layer information for UE-ID-based beamforming	No special handling needed
15	Mixed-numerology information for UE-ID-based beamforming	No special handling needed
16	Antenna mapping for UW channel information-based UL beamforming	No special handling needed
17	User port group indication	The usage of Section Extension 17 is always combined with Section Extension 10, same attention as for number 10 is thus applicable.

7.4.7.19 ExtType=19: Section Compact multiple port beamforming information

This Section Extension applies to Section Types 1 and 3. This Section Extension is required for sending compact beamforming information for multiple antenna ports (the term ‘port’ used henceforth in context of this Section Extension refers to logical antenna port). CSI-RS channel will benefit the most from using this extension, considering large number of CSI-RS ports and multiple CSI resource set.

This Section Extension is structured into a common Section Extension header specifying total number of ports, consolidated symbol bitmask and beamforming weight compression header applicable for all the ports. This is followed by fields describing per port information. Per port fields in the extension specify the location of port using a separate portReMask and portSymbolMask. The per port section further contains fields to specify per port beamforming information. When using this for CSI-RS one instance of this extension can be used to specify one CSI-RS resource set. For section description with this extension, reMask in section header is an aggregate of portReMask for all ports specified in Ext-19. Also, if section description has Ext-19, the beamId in section header shall be ignored.

First instance of this extension shall be used with ‘repetition=0’ and shall contain beamweights/beamIds for all ports in the range of startPrb and numPrb. When used with highest priority sections, and the section is repeated, this extension can be sent with ‘repetition=1’, in which case, per port fields are excluded from the extension indicating the associated beamforming information has already been sent to the O-RU. This extension can be used with any coupling method. Interaction with other extensions is present in **Table 7-48**.

Table 7-47: Section Extension 19

0 (msb)	1	2	3	4	5	6	7 (lsb)	# of bytes	
ef	extType = 0x13							1	Octet N
extLen[15:0]								2	N+1
disableBFWs	repetition	numPortc[5:0]						1	N+3
priority		symbolMask[13:8]						1	N+4
symbolMask[7:0]								1	N+5
bfwCompHdr								var	
reserved				portReMask[11:8] (for port 1)				1	
portReMask[7:0] (for port 1)								1	
reserved		portSymbolMask[13:8] (for port 1)						1	
portSymbolMask[7:0] (for port 1)								1	
reserved	beamID[14:8] (for port 1)							1	
beamID[7:0] (for port 1)								1	
bfwCompParam (for port 1)								var	
bfwI (for port 1 and TRX 0)								var	
bfwQ (for port 1 and TRX 0)								var	

remaining beamforming weights bfwI and bfwQ up to L TRXs		var	
…			
reserved	portReMask[11:8] (for last port)	1	
portReMask[7:0] (for last port)		1	
reserved	portSymbolMask[13:8] (for last port)	1	
portSymbolMask[7:0] (for last port)		1	
bfwCompParam (for last port)		var	
reserved	beamID[14:8] (for last port)	1	
beamID[7:0] (for last port)		1	
bfwI (for last port and TRX 0)		var	
bfwQ (for last port and TRX 0)		var	
remaining beamforming weights bfwI and bfwQ up to L TRXs		var	
Zero pad to 4-byte boundary		var	

7.4.7.19.1 disableBFWs (disable beamforming weights)

Description: refer to clause 7.4.7.11.6 Note: This parameter as used in ExtType=11 is used to enable/disable sending beamforming weights as part of this extension.

7.4.7.19.2 repetition (repeat port info flag)

Description: This parameter is used to indicate repetition of port beamforming information within a C-Plane message with Ext-19.

(see clause 7.4.7.6.5).

Value range: {0b=per port info present in the extension; 1b=per port information not present in the extension}.

Type: binary bit.

Field length: 1 bit

7.4.7.19.3 numPortc

Description: refer to clause 7.4.7.10.2

Note: For this extension, parameter indicates the number of logical antenna ports for which associated RE information and beamforming information is contained in this extension. It can cover up to 64 ports.

7.4.7.19.4 priority (priority of section description)

Description: This parameter has same format and semantics as priority parameter in Section Extension 6 (see clause 7.4.7.6.4).

7.4.7.19.5 symbolMask (resource symbol bitmask)

Description: This parameter, similar in functionaility to symbolMask in clause 7.4.7.6.3, is a consolidated symbol bitmask of all the ports described in this extension. This parameter applies to all PRBs specified by startPrbc and numPrbc in the section header containing this extension. If the symbolMask values indicate allocations beyond a slot boundary, such allocations shall be ignored (e.g. when there are fewer than 14 symbols in a slot). LSB of symbolMask indicates symbol zero (the first symbol to arrive in a slot).

This parameter in the common part of this extension along with reMask in section header helps the O-RU in optimized implementation where looping over per port information in the extension is not required for consolidated port information(which is useful for puncturing low priority sections). Also, for the case where ‘repetition = 1’ in this extension, and per port information is not present in the extension, ‘symbolMask’ together with ‘reMask (in section header)’ is sufficient for carrying consolidated port information.

Value range: {00 0000 0000 0001b-11 1111 1111 1111b}.

Type: unsigned integer (bit mask).

Field length: 14 bits.

7.4.7.19.6 bfwCompHdr

Description: refer to clause 7.4.7.1.1

If ‘repetition =1’ or ‘disableBFWs=1’ beamforming weights are not present in the extension, and hence ‘bfwComHdr’ parameter is also not present in the extension header.

7.4.7.19.7 portReMask (RE bitmask per port)

Description: This parameter defines the port Resource Element (RE) mask within a PRB. Each bit set in the portReMask indicates the RE associated with the port. MSB indicates the value for the RE of the lowest frequency in a PRB.

Value range: {0000 0000 0001b-1111 1111 1111b}.

Type: unsigned integer (bit mask).

Field length: 12 bits.

7.4.7.19.8 portSymbolMask (symbol bitmask per port)

Description: This bitmask specifies the symbols associated with a specific port. It is a subset of symbolMask specified in clause 7.4.7.19.5.

Value range: {00 0000 0000 0001b-11 1111 1111 1111b}.

Type: unsigned integer (bit mask).

Field length: 14 bits.

7.4.7.19.9 bfwCompParam (beamforming weight compression parameter)

Description: refer to clause 7.4.7.1.2 7
Note: This parameter provides beamforming compression parameters associated with a port.

7.4.7.19.10 beamId (beam identifier for a port)

Description: refer to clause 7.4.5.9
Note: This parameter provides the beamID associated with a port

7.4.7.19.11 bfwI (beamforming weight in-phase value)

Description: refer to clause 7.4.7.1.3

7.4.7.19.12 bfwQ (beamforming weight quadrature-phase value)

Description: refer to clause 7.4.7.1.4

7.4.7.19.13 Interaction with Other Section Extensions

Table 7-48: Section Extension =19 Interactions with other Section Extensions

Section Extension	Title	Interaction with existing Section Extensions

1	Beamforming Weights	Section Extension = 1 shall not be used in the same section as Section Extension = 19, as Section Extension =19 provides an alternate way of specifying Beam Weights.
2	Beamforming Attributes	Section Extension = 2 shall not be used in the same section as Section Extension = 19, as Section Extension = 19 provides an alternate way of specifying beams.
3	DL Precoding	Section Extension = 3 is not compatible with Section Extension = 19 and shall not be used together in the same section.
4	Modulation Compression	Can be used with Section Extension = 19.
5	Modulation Compression (Additional)	Can be used with Section Extension = 19.
6	Non-Contiguous PRB	Section Extension = 6 shall not be used, as Section Extension =19 provides an alternate way of specifying priorities and symbol Masks.
7	eAxC Mask	Can be used with Section Extension = 19.
8	Regularization factor	Not applicable as Section Extension = 19 is not intended to be used with Section Type 5.
9	DSS Parameters	Can be used with Section Extension = 19.
10	Group Configuration for multiple ports	Section Extension =19 can be used with Section Extension =10 only if the Beamforming Weights transferred using Section Extension =19 are same for all streams (or layers). In general, the BF weight vector will be different per data layer (eAxC) and Section Extension =19 cannot be used in conjunction. Once the Beamforming Weights are downloaded, then if the O-DU uses same BF vector by addressing Beam Id for a number of slots after the BF Weights are downloaded, then in that case, the O-DU can use Section Extension =19 to combine multiple C-plane messages to one single C-plane message using the representative eAxC id.
11	Flexible Beamforming Weights	Section Extension = 11 could be used with Section Extension = 19, for providing per port flexible beamforming weights. Section Extension = 19 shall appear before Section Extension = 11. There would be one instance of Section Extension =11 per port. Beam IDs specified in Section Extension = 19 shall be ignored and disableBFWs in section Extension =19 shall be set to ‘1’ to exclude beamweights in Section Extension = 19.
12	Non-Contiguous PRB Allocation with Frequency Ranges	Section Extension = 12 shall not be used, as Section Extension =19 provides an alternate way of specifying priorities and symbol Masks.
13	PRB Allocation with Frequency Hopping	Can be used with Section Extension = 19.
14	Nulling-Layer Info	Not applicable as Section Extension = 19 is not intended to be used with Section Type 5.
15	Mixed Numerology Info for ueId-based beamforming	Not applicable as Section Extension = 19 is not intended to be used with Section Type 5.
16	Antenna Information in UE Channel Information based UL beamforming	Not applicable as Section Extension = 19 is not intended to be used with Section Type 5.
17	Indication of User Port group	Not applicable as Section Extension = 19 is not intended to be used with Section Type 5.

18	Uplink Transmission Management	Can be used with Section Extension = 19.
20	Puncturing Extension	Can be used with Section Extension = 19.

7.4.7.20 ExtType=20: Puncturing Extension

This extension specifies the Puncturing Pattern to be applied to a section. This extension contains a common header with the number of puncturing patterns, followed by symbolMask, PRB ranges, puncReMask and optionally RBG mask fields for each puncturing pattern.

O-RU shall process the fields of the puncturing pattern and remove all the overlapping REs from the current section description. For Coupling via Time and Frequency with Priorities (or optimized), Ext-20 is restricted to be used with only the lower priority section to avoid any ambiguity in application of puncturing patterns.The number of puncturing patterns in an Ext-20 and the number of highest priority sections in a c-plane message together should be less than “max-highest-priority-sec-per-cplane-message” limits defined in M-plane.

Table 7-49: Section Extension 20

0 (msb)	1	2	3	4	5	6	7(lsb)	# of bytes	
ef	extType = 0x14							1	Octet N
extLen[15:0]								2	N+1
numPuncPatterns[7:0]								1	N+3
symbolMask[13:6](1)								1	N+4
symbolMask[5:0](1)						startPuncPrb[9:8] (1)		1	N+5
startPuncPrb[7:0](1)								1	N+6
numPuncPrb[7:0](1)								1	N+7
puncReMask[11:4](1)								1	N+8
puncReMask[3:0](1)				rb(1)	reserved(1)		rbgIncl(1)	1	N+9
reserved(1)	rbgSize[2:0](1)			rbgMask[27:24](1)				0-1	var
rbgMask[23:16](1)								0-1	var
rbgMask[15:8](1)								0-1	var
rbgMask[7:0](1)								0-1	var
…									
symbolMask[13:6](last)								0-1	
symbolMask[5:0](last)						startPuncPrb[9:8](last)		0-1	
startPuncPrb[7:0](last)								0-1	
numPuncPrb[7:0](last)								0-1	
puncReMask[11:4](last)								0-1	
puncReMask[3:0](last)				rb(last)	reserved(last)		rbgIncl(last)	0-1	
reserved(last)	rbgSize[2:0](last)			rbgMask[27:24](last)				0-1	
rbgMask[23:16](last)								0-1	
rbgMask[15:8](last)								0-1	
rbgMask[7:0](last)								0-1	
Zero pad to 4-byte boundary								var	

7.4.7.20.1 numPuncPatterns (number of puncturing pattern)

Description: This parameter is used to indicate total number of puncturing patterns contained within single instance of this extension.

Value range: {000000b-111111b}.

Type: unsigned integer.

Field length: 8 bits.

7.4.7.20.2 symbolMask (puncturing pattern symbol mask)

Description: This parameter (also see clause 7.4.7.6.3 and 7.4.7.19.8) is a bitmask where each bit indicates the symbols associated with the puncturing pattern. A value of '1' indicates that the symbol shall be considered for puncturing. A value of '0' indicates the symbol need not be considered for puncturing.

Value range: {00 0000 0000 0001b-11 1111 1111 1111b}.

Type: unsigned integer (bit mask).

Field length: 14 bits.

7.4.7.20.3 startPuncPrb (starting PRB to which one puncturing pattern applies)

Description: This parameter conveys the first PRB of the puncturing pattern.

Value range: {00 0000 000b-11 1111 1111b}.

Type: unsigned integer.

Field length: 10 bits.

7.4.7.20.4 numPuncPrb (number of contiguous PRBs to which one puncturing pattern applies)

Description: This parameter conveys the number of PRBs of the puncturing pattern

Value range: {0000 0000b-1111 1111b}.

Type: unsigned integer (bit mask).

Field length: 8 bits.

7.4.7.20.5 puncReMask (puncturing pattern RE mask)

Description: This parameter defines the Resource Element (RE) mask of the puncturing pattern within a PRB. Each bit in the puncReMask indicates the presence/absence of a puncturing RE within a PRB. MSB indicates the value for the RE of the lowest frequency in a PRB.

Value range: {0000 0000 0001b-1111 1111 1111b}.

Type: unsigned integer (bit mask).

Field length: 12 bits.

7.4.7.20.6 rb (resource block indicator)

Description: refer to clause 7.4.5.2
Note: This shall not be applicable if rbgIncl flag is set to '1' in this extension

7.4.7.20.7 rbgIncl (rbg included flag)

Description: This parameter is used to indicate presence/absence of resource block group for the case of non-contiguous PRB allocation. If this flag is set to ‘0’, 2 fields following this parameter i.e. rbgSize and rbgMask are absent and vice versa.

Value range: {0b=resource block group included; 1b=resource block group not included}.

Type: binary bit.

Field length: 1 bit

7.4.7.20.8 rbgSize (rbg size)

Description: refer to clause 7.4.7.6.1

7.4.7.20.9 rbgMask (rbg bitmask)

Description: refer to clause 7.4.7.6.2

7.4.7.20.9 Interaction with Other Extensions

Ext 20 is compatible with all other Extensions. Ext-20 is used to describe the puncturing information of ***other channels and their associated sections*** on top of the current section. Whereas, other extensions are used to describe the Beamforming and Scheduling information of the current section.

7.5 C-Plane Optimizations

7.5.1 C-Plane Optimization using Section Extension =6

Section Extension =6 is used for non-contiguous PRB allocation in both time and frequency domains. This is realized by two bitmasks: symbolMask and rbgMask. The first allows to select an arbitrary subset of symbols within a slot. The second allows to select arbitrary subset of blocks of subcarriers (each block has 12*rbgSize subcarriers) between startPrbc and startPrbc+numPrbc. The selected set of RE is cross-section of symbols and subcarriers selected by both masks. This allows to describe a wide range of non-continuous resource allocations with one section description.

In addition, O-RU may support optional feature allowing it to interpret a non-zero value priority carried in this Section Extension. This is optimization described in clause 7.5.5.

7.5.2 C-Plane Optimization using Section Extension =7

Material to come.

7.5.3 C-Plane Optimization using Section Extension =10

In general, the O-DU uses unique eAxC ids to address each layer or spatial stream when sending C-plane and U-plane messages to the O-RU. In many situations, information contained in C-plane messages for the different spatial streams is the same or similar. For example, a SU MIMO allocation with 8 layers may have same values for startPrbc, NumPrbc, reMask, and numSymbol in the section header for all 8 C-plane messages, one message per layer. In this case, Section Extension =10 can be used along with a ‘representative eAxC’ ID (M-plane configured) to reduce C-plane overhead of sending multiple messages to the overhead of sending one single C-plane message. This is an optional feature which be taken advantage of by the O-DU if the O-RU capability indicates that the O-RU supports Section Extension =10. Based on this capability, the O-RU can be configured with a ‘representative eAxC’ ID representing all 8 eAxC IDs in above example. Once a ‘representative eAxC’ id is configured, the O-DU can send one single C-plane message addressed to the ‘representative eAxC’ ID along with the Section Extension 10 instead of 8 C-plane messages in above example case. Once the O-RU receives a C-plane message addressed to the ‘representative eAxC’ ID and with Section Extension 10, then the O-RU applies this message to all endpoints pointed to by the ‘representative eAxC’ ID whereby performing same operation just like it had received 8 different C-plane messages. **Figure 7-10** below illustrates the example.

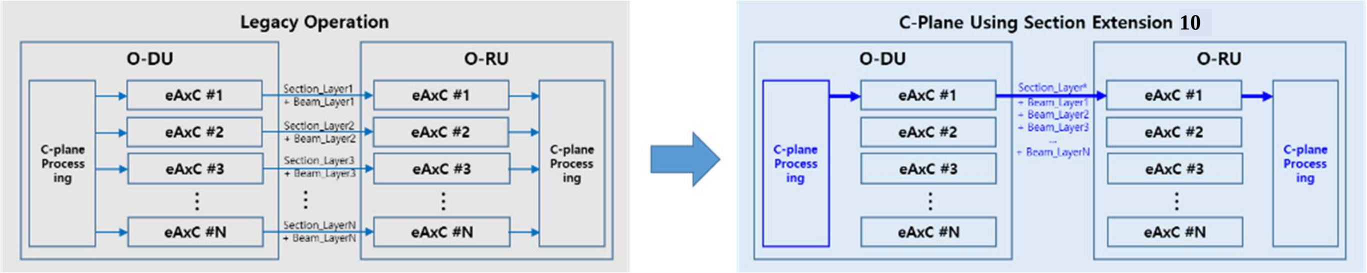

Legacy Operation
O-DU
O-RU
C-plane Process ing
eAxC #1
eAxC #2
eAxC #3
eAxC #N
Section_Layer1 + Beam_Layer1
Section_Layer2 + Beam_Layer2
Section_Layer3 + Beam_Layer3
Section_LayerN + Beam_LayerN
C-Plane Using Section Extension 10
Section_Layer* + Beam_Layer1 + Beam_Layer2 + Beam_Layer3 ... + Beam_LayerN

Figure 7-10 : Operation of Section Extension =10

7.5.4 C-Plane Optimization using Section Extension =11

Section Extension =11 referes to flexible beamforming weight transmission. One way to transfer per-PRB beamforming weights is using Section Extension =1 and creating one section for each PRB. This introduces high overhead in the form of section headers and Section Extensions. The objective of Section Extension =11 is to eliminate this problem by allowing the inclusion of the beamforming weights for multiple PRBs in a single section.

Example use case include zero-forcing beamforming (or similar methods) by using per-PRB channel information to calculate per-PRB beamforming weights. Specifically, the O-RU sends UL pilots (SRS) to O-DU, which in return calculates per-PRB beamforming weights in the O-DU, then transfer them to the O-RU. These weights can be updated every slot, as an example.

Section Extension =11 gives the flexibility of either sending the sets of beamforming weights along with the Section Extension beamIds, or only the Section Extension beamIds (after, obviously, loading the beamforming weights in a previous time).

Notes:

1. By default, the section beamID will have no meaning if used with Section Extension 11 and will be neglected by the O-RU

7.5.4.1 Interaction between Section Extension =11 and Section Extension =6

Section Extension =6 is used for non-contiguous PRB allocation in both time and frequency domains. The main parameter in Section Extension =6 that is used to identify the PRB groups is rbgSize. On the other hand, numBundPrb is used in Section Extension =11 to identify the number of bundled PRBs that share the same beamforming weights vector for L TRXs. Both Section Extensions can be used together, however, few rules need to be followed:

First case: RAD = 1

1) In principle, numBundPrb may not be the same as rbgSize.
2) For each contiguous set of PRBs as defined under Section Extension =6 (if used), the O-RU starts counting for the number of bundled PRBs and applying the sets of beamforming weights to PRB bundles in an increasing order of PRBs.
    a. If the last bundled PRB doesn’t coincide with the last RBG in the contiguous set of PRBs, then the unassigned PRBs (i.e., PRBs with unassigned beamforming weights) will be considered orphan PRB(s).
3) Orphan PRBs are allocated the following set of beamforming weights (i.e., set after the last set that was assigned to a regular PRB bundle) although the number of PRBs is less than a PRB bundle. For any set of PRBs that are not being assigned using Section Extension =6, no beamforming weight sets are assigned.

4) For the following set of contiguous PRBs allocated in the same symbol after the discontinuity, the O-RU will apply the following set of beamforming weights (starting from the last set of beamforming weights allocated to the last PRB bundle in the previous contiguous PRB set).
5) The process continues until all the PRBs defined under Section Extension =6 are assigned beamforming weights. At this point, all the sets of beamforming weights under Section Extension =11 should have been assigned.
6) Based on the symbol mask, the O-RU applies the same sets of beamforming weights to the corresponding PRB bundles similar to the first assigned symbol.

Example 1: RAD = 1b (see **Figure 7-11** below)
symbolMask = 10 0001 0001 0000b
rbgSize = 3
numBundPRB = 2
Total number of L beamforming weights set = 7

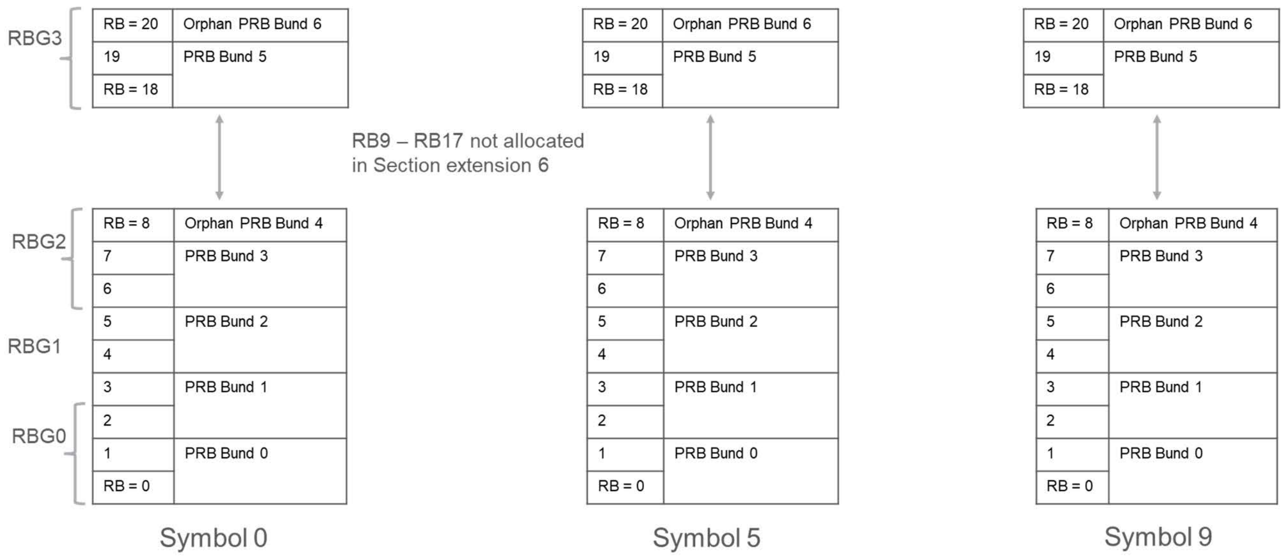


Figure 7-11 : Example for how Section Extension 6 interacts with Section Extension =11: Case RAD=1

Second case: RAD = 0

In this case, the set of beamforming weights for a given bundle are applied to the PRBs in increasing order regardless of the discontinuity in the PRBs allocation in Section Extension =6. If the gap in the PRBs allocation is equal to an integer number of PRB bundles, then no beamforming weights are sent by the O-DU for the PRBs in those gaps.

Example 2: RAD = 0b (see **Figure 7-12** below)
symbolMask = 10 0001 0001 0000b
rbgSize = 3
numBundPRB = 10
Total number of L beamforming weights set = 3

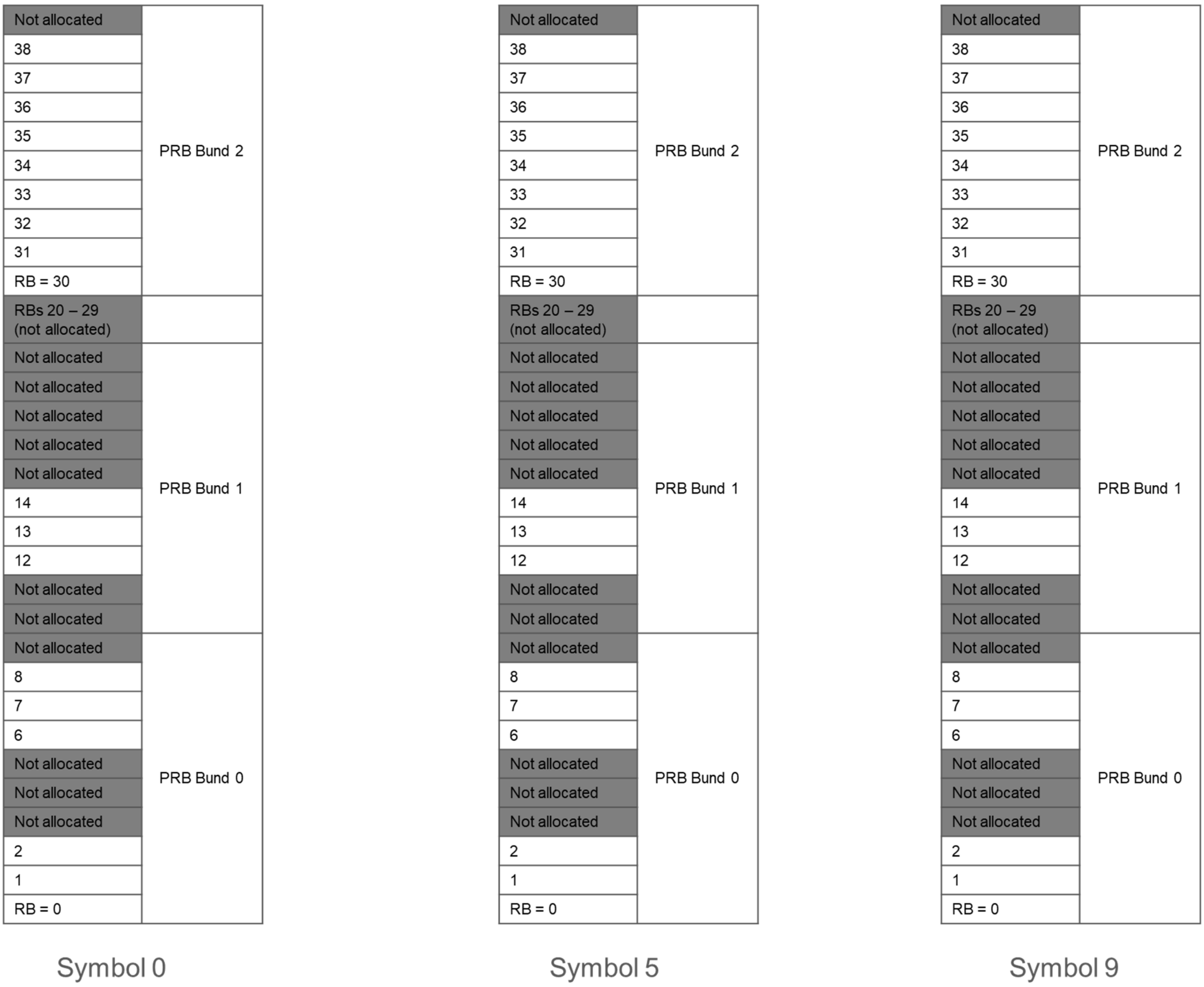


Figure 7-12 : Example for how Section Extension =6 interacts with Section Extension =11: Case RAD = 0

7.5.4.2 Interaction between Section Extension =11 and Section Extension =12

Section Extension =12 is used for non-contiguous PRB allocation in both time and frequency domains. Section Extension =11 can be combined with Section Extension =12 in one section description in the same way as how Extension =11 may be combined with Section Extension =6. That is, description in clause 7.5.4.2 applies also to Section Extension =12. See also the examples below.

Example for RAD = 0

symbolMask = 00 0000 0000 0101b
grey rectangles represent unallocated frequency ranges (a single PRB discontinuity starting at PRB 5)
numBundPRB = 3
Total number of L beamforming weights set = 4

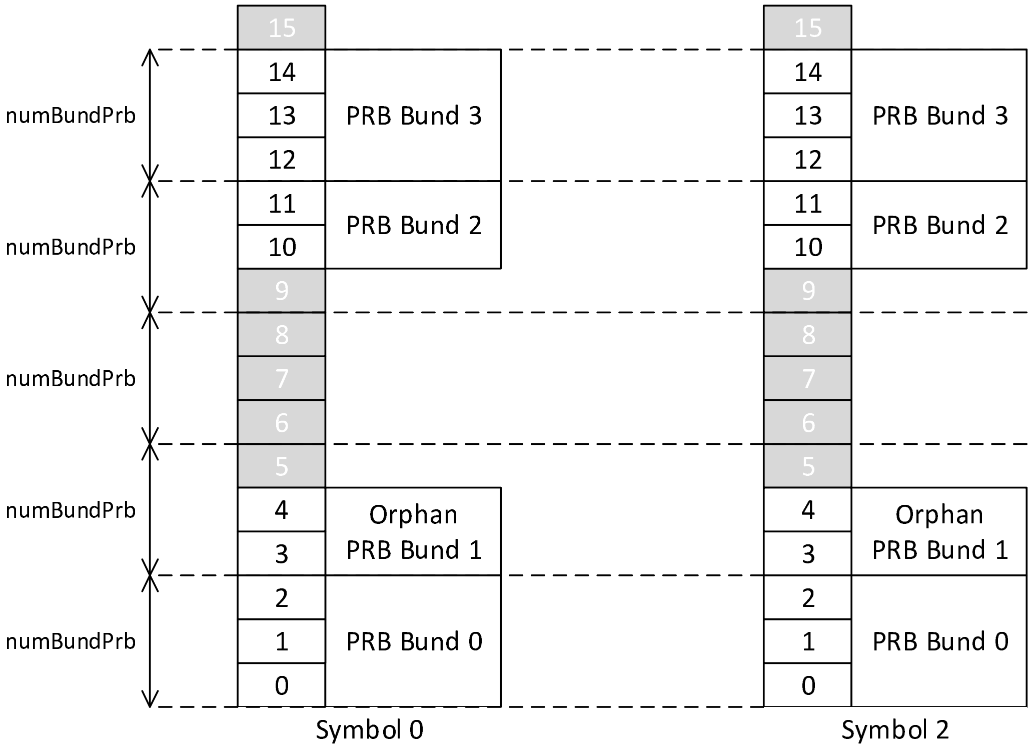


Figure 7-13 : Example for how Section Extension 12 interacts with Section Extension =11: Case RAD=0

Example for RAD = 1

symbolMask = 00 0000 0000 0101b
grey rectangles represent unallocated frequency ranges (a single PRB discontinuity starting at PRB 5)
numBundPRB = 3
Total number of L beamforming weights set = 4

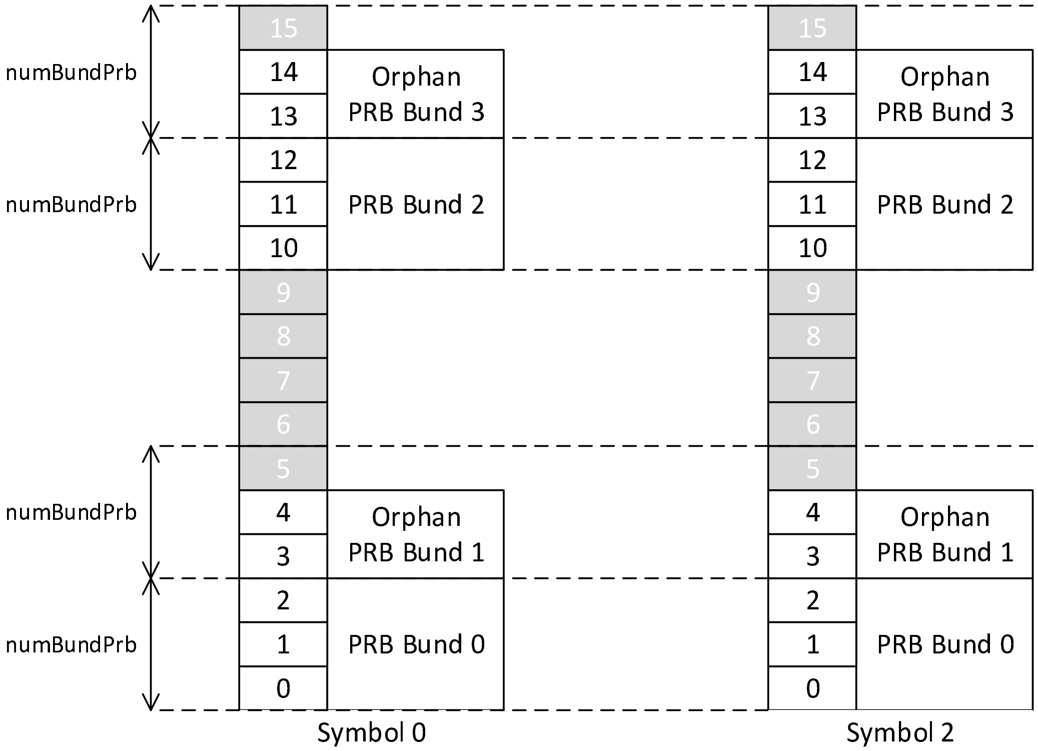


Figure 7-14 : Example for how Section Extension 12 interacts with Section Extension =11: Case RAD = 1

7.5.4.3 Interaction between Section Extension =11 and Section Extension =13

Section Extension =13 is used for non-contiguous PRB allocation with frequency hopping. Section Extension =11 can be combined with Section Extension =12 in one section description in the same way as how Extension =11 may be combined with Section Extension =6. That is, description in clause 7.5.4.2 applies also to Section Extension =12 with two modifications:

- beamIds and beamforming weights carried by Section Extension =11 are provided for a combined PRB set that is result of merging PRB sets of all hops. This rule is general and applies to Section Extension =13 combined with Section Extensions allowing non-contiguous PRB allocation (Section Extensions 6 and 12).
- PRB discontinuities are discontinuities of the combined PRB set.

See examples below.

Example for RAD = 0

startSymbolId=0 (and message includes only one section description)
startPrbc=10, numPrbc=5, numSymbol=3
nextStartSymbolId(0)=1, nextStartPrbc(0) = 0,
nextStartSymbolId(1)=2, nextStartPrbc(1) = 7,
grey rectangles represent unallocated frequency ranges
note the combined set of PRB has one discontinuity starting at PRB 5
numBundPRB = 3
Total number of L beamforming weights set = 5

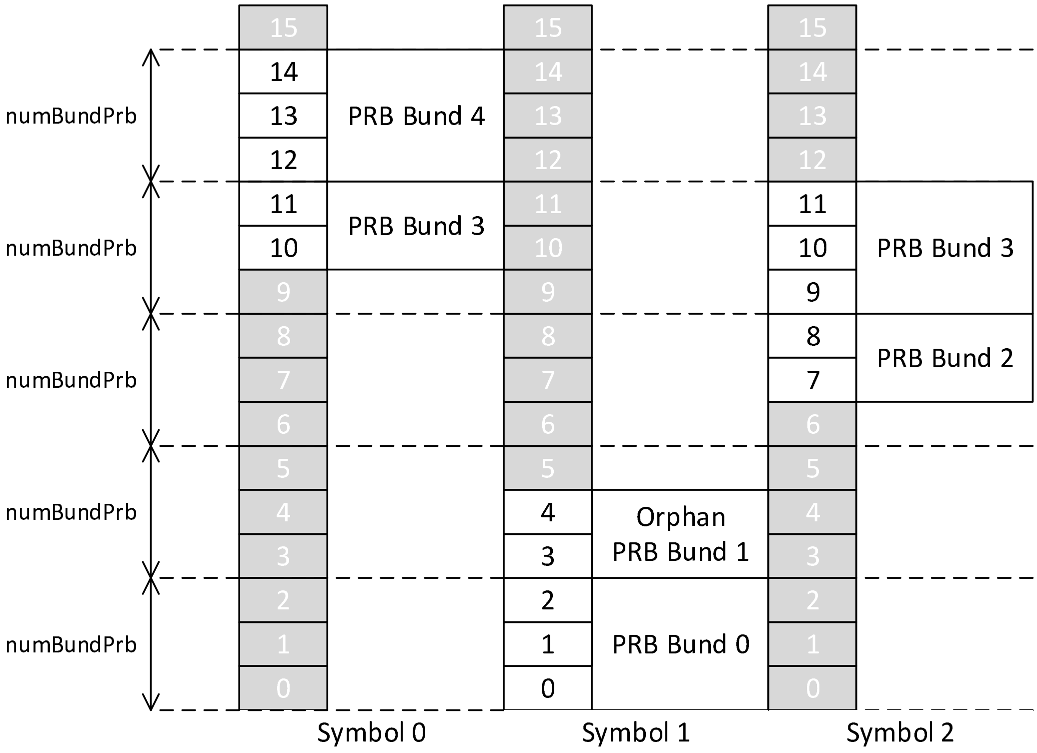


Figure 7-15 : Example for how Section Extension 13 interacts with Section Extension =11: Case RAD = 0

Example for RAD = 1

startSymbolId=0 (and message includes only one section description)
startPrbc=10, numPrbc=5, numSymbol=3
nextStartSymbolId(1)=1, nextStartPrbc(1) = 0,
nextStartSymbolId(2)=2, nextStartPrbc(2) = 7,
gray rectangles represent unallocated frequency ranges
note the combined set of PRB has one discontinuity starting at PRB 5
numBundPRB = 3
Total number of L beamforming weights set = 5

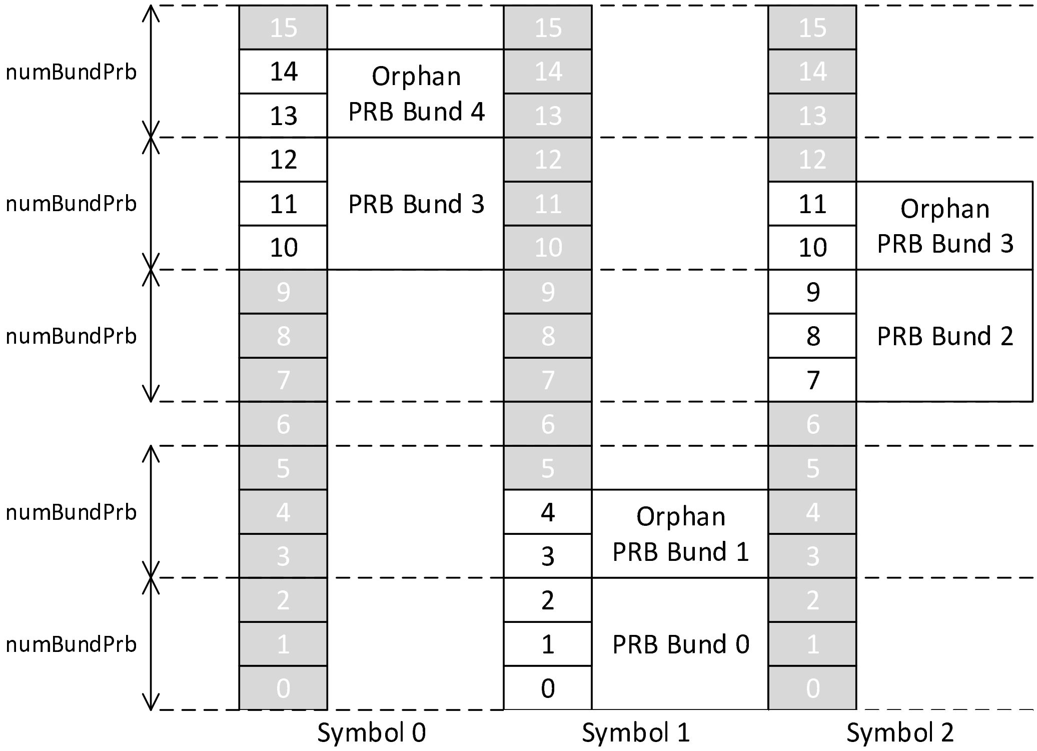


Figure 7-16 : Example for how Section Extension 13 interacts with Section Extension =11: Case RAD = 1

Example for RAD = 0 with a non-contiguous PRB allocation using Section Extension=12

startPrbc=8, numPrbc=2,
nextStartSymbolId(1)=2, nextStartPrbc(1) = 0,
symbolMask= 00 0000 0000 0101b
offStartPrb(1)=4, numPrb(1)=2
gray rectangles represent unallocated frequency ranges
note the combined set of PRB has two discontinuities starting at PRB 2 and PRB 10
numBundPRB = 3
Total number of L beamforming weights set = 5

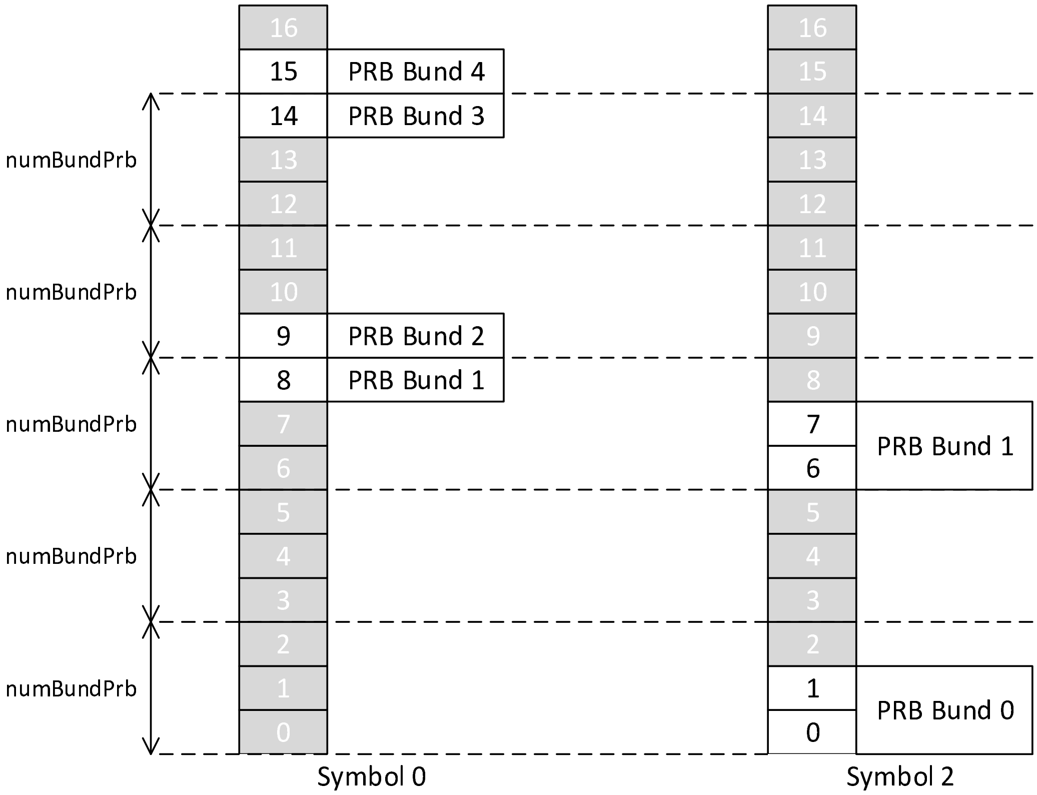


Figure 7-17 : Example for how Section Extension 13 and 12 interact with Section Extension =11: Case RAD = 0

Example for RAD = 1 with a non-contiguous PRB allocation using Section Extension=12

startPrbc=8, numPrbc=2,
nextStartSymbolId(1)=2, nextStartPrbc(1) = 0,
symbolMask= 00 0000 0000 0101b
offStartPrb(1)=4, numPrb(1)=2
gray rectangles represent unallocated frequency ranges
note the combined set of PRB has two discontinuities starting at PRB 2 and PRB 10
numBundPRB = 3
Total number of L beamforming weights set = 4

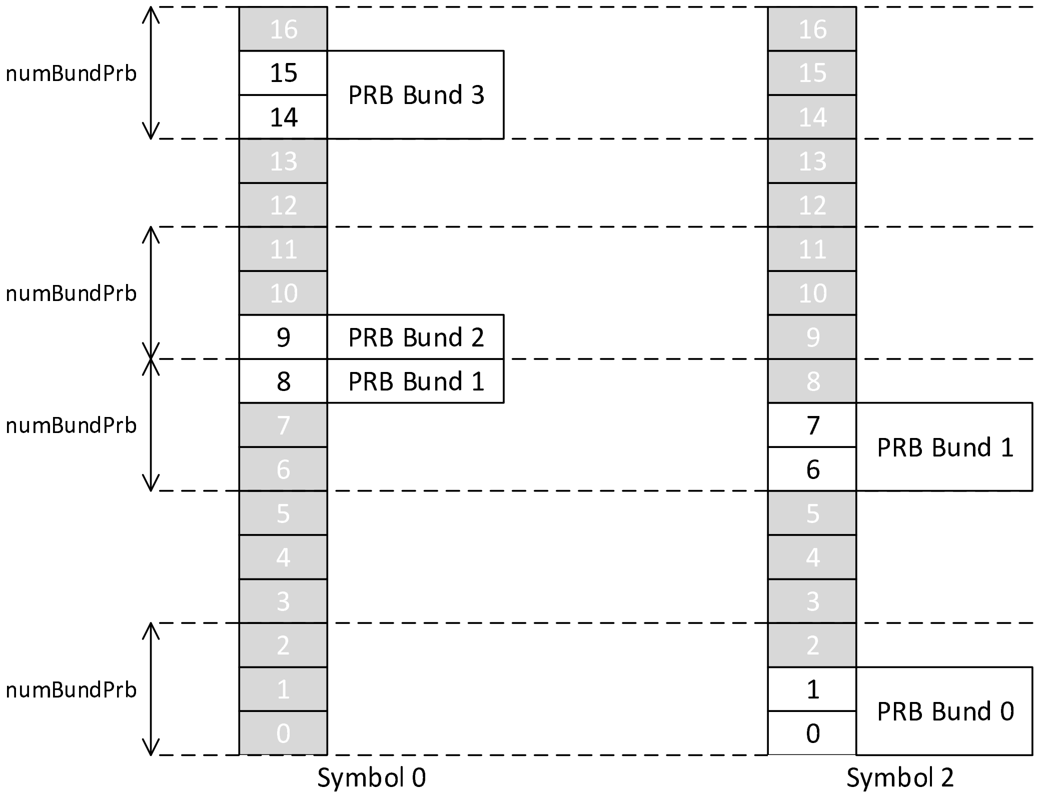


Figure 7-18 : Example for how Section Extension 13 and 12 interact with Section Extension =11: Case RAD = 0

7.5.5 C-Plane Optimization using Section Extension =12

The Section Extension =12 is useful for PRB allocation non-contiguous in time and frequency domains. The Section Extension =12 conveys priority, symbolMask and a variable size list of frequency ranges. For that is more space efficient than Section Extension =6 in scenarios where:

PRB allocation is continuous in frequency domain but non-contiguous in time domain (in this case the list of frequency ranges can be empty)

PRB allocation consists of few continuous fragments of spectrum that are spread over wide frequency span (in this case size of rbgMask is the factor limiting application of Section Extension =6).

This optimization cannot be combined with Section Extension =6 in one section description.

7.5.6 C-Plane Optimization using Section Extension =13

The Section Extension =13 is useful for PRB allocation with intra-slot frequency hopping. The Section Extension =13 conveys a variable size list of pairs of parameters indicating a value to be used in place of startPrbc and a time instant (symbol number) starting from which the startPrbc shall be substituted.

Size of Section Extension is smaller than size of section description and for that it is more efficient in scenarios where intra-slot frequency hopping is used.

The Section Extension =13 can be combined with other Section Extensions. In such case interpretation of frequency parameters conveyed by the section description and all Section Extensions shall be as if value of startPrbc was modified at time instances according to parameters in Section Extension =13.

7.5.7 Coupling via Frequency and Time with Priorities

This optimization uses the coupling mechanism described in 5.4.1.2.3 and allows the reduction of the number of section descriptions. The coupling mechanism with priorities allows to unambiguously describe beamforming configuration even if section descriptions overlap (i.e. refer to the same RE) provided they have different priorities. Message size reduction is achieved by moving beamforming configuration of selected REs (e.g. REs allocated to reference signals) to a combined section description with a higher priority. With this other section descriptions of lower priority can be simplified (e.g. details of RE allocation in PRB conveyed in reMask may be simplified by setting reMask to all ones) and may be combined. Combining section descriptions requires use of coupling mechanism that does not use sectionId. Note that “higher priority” and “lower priority” are relative e.g. priority 0 is higher than priority -1.

To enable O-RU to decide beamforming of a RE by processing a single C-plane message (this is property of coupling by sectionId) O-DU shall ensure that each message that has a section description referring to a RE includes the section description referring to the RE with highest priority. This may result in duplicating a subset of section description in more than one message but does not result in beamforming configuration ambiguity. Note that “highest priority” is relative e.g. if a RE is referenced only by section description with priority -1 then this is the section description with the highest priority.

The following example is simplified (e.g. symbolMask and reMask have less bits than actual fields in a message) to demonstrate the optimization mechanism and does not reflect a realistic resource allocation.

Initial set of section descriptions (assuming coupling via sectionId):

#	sectionId	symbolMask	startPrbc	numPrbc	reMask	beamId	Note
1	100	0110110	0	4	111111	100	UE1
2	101	1001000	0	4	110110	100	UE1
3	101	1001000	0	4	001000	1	Reference signal
4	101	1001000	0	4	000001	2	Reference signal
5	200	0110110	4	4	111111	200	UE2
6	201	1001000	4	4	110110	200	UE2
7	201	1001000	4	4	001000	1	Reference signal
8	201	1001000	4	4	000001	2	Reference signal
9	300	0110110	8	4	111111	300	UE3
10	301	1001000	8	4	110110	300	UE3
11	301	1001000	8	4	001000	1	Reference signal
12	301	1001000	8	4	000001	2	Reference signal
13	400	0000001	5	2	111111	3	Special channel

Optimized set of section descriptions (sectionId value is fixed and not significant):

#	priority	symbolMask	startPrbc	numPrbc	reMask	beamId	Note
1	+1	1001000	0	12	001000	1	Reference signal
2	+1	1001000	0	12	000001	2	Reference signal
3	0	1111110	0	4	111111	100	UE1
4	0	1111110	4	4	111111	200	UE2
5	0	1111110	8	4	111111	300	UE3
6	0	0000001	5	2	111111	3	Special channel

Note that in the optimized set of section descriptions, section descriptions 3, 4, 5 and 6 span over a continuous set of symbols and have priority 0 (default). They can be represented without Section Extension 6.

If optimized set of section descriptions would not fit in one message, then it could be divided into several message as presented below. Note the split into message is not realistic in order to demonstrate duplication of highest priority section descriptions.

Message #1:

#	priority	symbolMask	startPrbc	numPrbc	reMask	beamId	Note
1	+1	1001000	0	12	001000	1	Reference signal
2	+1	1001000	0	12	000001	2	Reference signal
3	0	1111110	0	4	111111	100	UE1
4	0	1111110	4	4	111111	200	UE2

Message #2:

#	priority	symbolMask	startPrbc	numPrbc	reMask	beamId	Note
1	+1	1001000	0	12	001000	1	Reference signal
2	+1	1001000	0	12	000001	2	Reference signal
3	0	1111110	8	4	111111	300	UE3

In this message section descriptions 1 and 2 are duplicated because they are highest priority section descriptions referring to REs referred by section descriptions in the message (here section description 3).

Message #3:

#	Priority	symbolMask	startPrbc	numPrbc	reMask	beamId	Note
1	0	0000001	5	2	111111	3	Special channel

In this message section description 1 is the section description with highest priority and there is no need to duplicate section descriptions with priority +1. In other words: for every RE referenced from the message the message contains the highest priority section description referring to the RE.

7.5.7.1 Coupling via Frequency and Time with Priorities (Optimized)

This coupling method is an optimization over Coupling via Frequency and Time with Priorities to identify duplication of highest priority section description. When highest priority section descriptions are duplicated, it may cause O-RU to process the duplicated sections multiple times. Depending on the O-RU implementation, this may lead to erroneous behavior and/or increased processing. To identify duplication only when ‘coupling via frequency and time with priorities optimized’ is enabled, O-DU shall set each highest priority section to have unique sectionId (clause 7.4.1.2.4). This enables O-RU to identify duplicate highest priority section and take any optimized action.

Using the same example as ‘coupling via frequency and time with priorities’ optimized set of section descriptions with sectionId column added (sectionId value is fixed to 4095 for lower priority sections whereas highest priority section will have unique sectionId values in a defined range):

#	sectionId	priority	symbolMask	startPrbc	numPrbc	reMask	beamId	Note
1	0	+1	1001000	0	12	001000	1	Reference signal
2	1	+1	1001000	0	12	000001	2	Reference signal
3	4095	0	1111110	0	4	111111	100	UE1
4	4095	0	1111110	4	4	111111	200	UE2
5	4095	0	1111110	8	4	111111	300	UE3
6	4095	0	0000001	5	2	111111	3	Special channel

As described in similar table above when section description does not fit in one message, the message is divided across several message where the highest priority section is repeated.

Message #1:

#	sectionId	priority	symbolMask	startPrbc	numPrbc	reMask	beamId	Note
1	0	+1	1001000	0	12	001000	1	Reference signal
2	1	+1	1001000	0	12	000001	2	Reference signal
3	4095	0	1111110	0	4	111111	100	UE1
4	4095	0	1111110	4	4	111111	200	UE2

In this message highest priority sections are associated with unique sectionIds to be identified by O-RU when same highest priority sections are duplicated across C-Plane messages. Sections with priority 0 are still assigned 4095 as the sectionId since they cannot be duplicated.

Message #2:

#	sectionId	priority	symbolMask	startPrbc	numPrbc	reMask	beamId	Note
1	0	+1	1001000	0	12	001000	1	Reference signal
2	1	+1	1001000	0	12	000001	2	Reference signal
3	4095	0	1111110	8	4	111111	300	UE3

Duplicated section descriptions across Message#1 and Message#2 can now be identified by O-RU using sectionId.

Message #3:

#	sectionID	Priority	symbolMask	startPrbc	numPrbc	reMask	beamId	Note
1	4095	0	0000001	5	2	111111	3	Special channel

7.5.8 U-Plane Operation Without C-Plane

O-RU may support U-plane operation without C-plane. In general, this function can be used for channels with resource allocation known (at least approximately, see below) during eAxC activation (e.g. PRACH and SRS). If this function is enabled via M-Plane for given eAxC then for this eAxC C-plane is not used to provide control information. Instead O-RU receives a static control information (resource allocation details and beamforming configuration are set before activation) via M-Plane. When array carrier operation is enabled O-RU handles U-Plane operation as configured over M-Plane. For more details see clause 8.2.2.

This function can be used also for channels where only approximate resource allocation is known during activation (e.g. SRS channel reception is scheduled periodically at known interval even if no UE is present). In this case resource allocation conveyed via M-plane generally would exceed actual resource allocation e.g. O-RU could be configured to receive channel related IQ data at times when channel could possibly be allocated without reflecting actual, varying over time allocations. Such approximations generally cause additional UL U-Plane traffic related to unallocated (at given time) resource elements; note that allocation DL U-Plane traffic is not expected.

This C-plane optimization method is not compatible with any Section Extensions since C-Plane messages are not used for selected eAxC.

7.5.9 Modulation Compression with Section Extension 10

Section Extension =10 is used for group configuration of multiple ports. Section Extension =4 and Section Extension =5 are used for modulation compression. Extension type 10 can be used together with extension type 4 or 5. When all parameter values of Section Extension 4 or 5 is same for all eAxC ids, append one single Section Type 4 or 5 after Section Extension 10. When parameter values are different for all eAxCs, append extension type 4 or 5 for all eAxCs in sequence based on eAxC order after extension type 10.

If O-RU receives only one extension type 4 or 5 with extension type 10, O-RU applies same parameters to all eAxC ids. If O-RU receives equal to the number of the eAxC ids grouped, O-RU applies the extension type 4 or 5 in the order of how eAxC ids are grouped. Any other number of extension type 4 or 5 is an error condition. O-RU knows how many extension type 4 or 5 will be present based on numPortc parameter in extension type 10.

Example #1: 4 Layers (numPortc=3), beamGroupType=00b or 01b for extension type 10 and all eAxC ids share same modulation compression parameters. One single extension5 is appended after Section Extension 10 in this case (Section Extension 5 with two scaler values, modulation compression parameters is used in this example).

Table 7-50: Section Extension =10 for beamGroupType = 00b or 01b with one single extension 5

0 (msb)	1	2	3	4	5	6	7(lsb)	# of bytes	
ef	extType = 0x0A							1	Octet N
extLen = 0x01 (1 word)								1	N+1
beamGroupType		numPortc=3						1	N+2
reserved								1	N+3
ef	extType = 0x05							1	N+4
extLen = 0x3 (3 words)								1	N+5
mcScaleReMask[11:4]								1	N+6
mcScaleReMask[3:0]				csf	mcScaleOffset [14:12]			1	N+7
mcScaleOffset [11:4]								1	N+8
mcScaleOffset [3:0]				mcScaleReMask[11:8]				1	N+9
mcScaleReMask[7:0]								1	N+10
csf	mcScaleOffset [14:8]							1	N+11
mcScaleOffset [7:0]								1	N+12
zero padding								1	N+13
zero padding								1	N+14
zero padding								1	N+15

Example #2: 4 Layer (numPortc=3), beamGroupType=00b or 01b for extension type 10 and eAxC ids use different modulation compression parameters. Four Section Extension5's are appended after Section Extension 10 in this case (Section Extension 5 with one scaler value, modulation compression parameters is used in this example).

Table 7-51: Section Extension =10 for beamGroupType = 00b or 01b with multiple extension 5

0 (msb)	1	2	3	4	5	6	7(lsb)	# of bytes	
ef	extType = 0x0A							1	Octet N
extLen = 0x01 (1 word)								1	N+1
beamGroupType		numPortc=3						1	N+2
reserved								1	N+3
ef	extType = 0x05							1	N+4
extLen = 0x2 (2 words)								1	N+5
mcScaleReMask[11:4]								1	N+6
mcScaleReMask[3:0]				csf	mcScaleOffset [14:12]			1	N+7
mcScaleOffset [11:4]								1	N+8
mcScaleOffset [3:0]				zero padding				1	N+9
zero padding								1	N+10
zero padding								1	N+11
ef	extType = 0x05							1	N+12
extLen = 0x2 (2 words)								1	N+13
mcScaleReMask[11:4]								1	N+14
mcScaleReMask[3:0]				csf	mcScaleOffset [14:12]			1	N+15
mcScaleOffset [11:4]								1	N+16
mcScaleOffset [3:0]				zero padding				1	N+17
zero padding								1	N+18
zero padding								1	N+19
ef	extType = 0x05							1	N+20
extLen = 0x2 (2 words)								1	N+21
mcScaleReMask[11:4]								1	N+22
mcScaleReMask[3:0]				csf	mcScaleOffset [14:12]			1	N+23
mcScaleOffset [11:4]								1	N+24
mcScaleOffset [3:0]				zero padding				1	N+25
zero padding								1	N+26
zero padding								1	N+27
ef	extType = 0x05							1	N+28
extLen = 0x2 (2 words)								1	N+29
mcScaleReMask[11:4]								1	N+30
mcScaleReMask[3:0]				csf	mcScaleOffset [14:12]			1	N+31
mcScaleOffset [11:4]								1	N+32
mcScaleOffset [3:0]				zero padding				1	N+33
zero padding								1	N+34
zero padding								1	N+35

Example #3: 4 Layer (numPortc=3), beamGroupType=10b for extension type 10, and all eAxC ids share same modulation compression parameters. One single extension5 is appended after Section Extension 10 in this case (Section Extension 5 with two scaler values, modulation compression parameters in this example).

Table 7-52: Section Extension =10 for beamGroupType = 10b with one single extension 5

0 (msb)	1	2	3	4	5	6	7(lsb)	# of bytes	
ef	extType = 0x0A							1	Octet N
extLen = 0x03 (3words)								1	N+1
beamGroupType		numPortc=3						1	N+2
reserved	2nd port beamID[14:8] (or ueID[14:8])							1	N+3
2nd port beamID[7:0] (or ueID[7:0])								1	N+4

reserved	3rd port beamID[14:8] (or ueID[14:8])							1	N+5
3rd port beamID[7:0] (or ueID[7:0])								1	N+6
reserved	4th port beamID[14:8] (or ueID[14:8])							1	N+7
4th port beamID[7:0] (or ueID[7:0])								1	N+8
zero padding								1	N+9
zero padding								1	N+10
zero padding								1	N+11
ef	extType = 0x05							1	N+12
extLen = 0x3 (3 words)								1	N+13
mcScaleReMask[11:4]								1	N+14
mcScaleReMask[3:0]				csf	mcScaleOffset [14:12]			1	N+15
mcScaleOffset [11:4]								1	N+16
mcScaleOffset [3:0]				mcScaleReMask[11:8]				1	N+17
mcScaleReMask[7:0]								1	N+18
csf	mcScaleOffset [14:8]							1	N+19
mcScaleOffset [7:0]								1	N+20
zero padding								1	N+21
zero padding								1	N+22
zero padding								1	N+23

Example #4: 4 Layer (numPortc=3), beamGroupType=10b for extension type 10, and eAxC ids use different modulation compression parameters. Four Section Extension5's are appended after Section Extension 10 in this case (Section Extension 5 with one scaler value, modulation compression parameters is used in this example).

Table 7-53: Section Extension =10 for beamGroupType = 10b with multiple extension 5

0 (msb)	1	2	3	4	5	6	7(lsb)	# of bytes	
ef	extType = 0x0A							1	Octet N
extLen = 0x03 (3words)								1	N+1
beamGroupType		numPortc=3						1	N+2
reserved	2^nd port beamID[14:8] (or ueID[14:8])							1	N+3
2^nd port beamID[7:0] (or ueID[7:0])								1	N+4
reserved	3rd port beamID[14:8] (or ueID[14:8])							1	N+5
3rd port beamID[7:0] (or ueID[7:0])								1	N+6
reserved	4th port beamID[14:8] (or ueID[14:8])							1	N+7
4th port beamID[7:0] (or ueID[7:0])								1	N+8
zero padding								1	N+9
zero padding								1	N+10
zero padding								1	N+11
ef	extType = 0x05							1	N+12
extLen = 0x2 (2 words)								1	N+13
mcScaleReMask[11:4]								1	N+14
mcScaleReMask[3:0]				csf	mcScaleOffset [14:12]			1	N+15
mcScaleOffset [11:4]								1	N+16
mcScaleOffset [3:0]				zero padding				1	N+17
zero padding								1	N+18
zero padding								1	N+19
ef	extType = 0x05							1	N+20
extLen = 0x2 (2 words)								1	N+21
mcScaleReMask[11:4]								1	N+22
mcScaleReMask[3:0]				csf	mcScaleOffset [14:12]			1	N+23
mcScaleOffset [11:4]								1	N+24
mcScaleOffset [3:0]				zero padding				1	N+25

zero padding			1	N+26
zero padding			1	N+27
ef	extType = 0x05		1	N+28
extLen = 0x2 (2 words)			1	N+29
mcScaleReMask[11:4]			1	N+30
mcScaleReMask[3:0]	csf	mcScaleOffset [14:12]	1	N+31
mcScaleOffset [11:4]			1	N+32
mcScaleOffset [3:0]	zero padding		1	N+33
zero padding			1	N+34
zero padding			1	N+35
ef	extType = 0x05		1	N+36
extLen = 0x2 (2 words)			1	N+37
mcScaleReMask[11:4]			1	N+38
mcScaleReMask[3:0]	csf	mcScaleOffset [14:12]	1	N+39
mcScaleOffset [11:4]			1	N+40
mcScaleOffset [3:0]	zero padding		1	N+41
zero padding			1	N+42
zero padding			1	N+43

7.5.10 Optimization with Ext Type = 19

Benefits of Ext 19 can be explained best with a CSI-RS example. Consider a CSI-RS resource with 32 ports, FD-CDM2. Each port is associated with a separate beam. Up to 16 ports (belonging to one CDM group) can be sent in a single layer (eAxC). Currently this requires 16 section invocations. Each section invocation is 8 bytes for a total of 128 bytes. Assuming 4 CSI-RS resources, this amounts to 64 sections and 512 bytes.

When used with Time and Frequency coupling with priorities, this will result in 64 highest priority sections. Fragmentation of C-Plane packets further results in duplicating these highest priority sections in every C-plane message, adding further to the fronthaul load in addition to O-DU and O-RU processing.

With ExtType=19, as opposed to processing up-to 16 section headers per CSI-RS resource, only one section header needs be processed by the O-RU. Processing highest priority sections for removing overlapping reMask bits becomes lot more efficient. Rather than accumulating reMasks and symbolMask across 16 high priority sections, the combined reMask and symbolMask of the entire CSI-RS resource is provided in one shot via reMask in Section header & symbolMask in extension. This makes the O-DU, O-RU processing more efficient, especially when dealing with fragmentation of C-plane packets.

Moreover, when repeating highest priority sections, the port information associated with Ext 19 need not be repeated, leading to significant Fronthaul savings

7.5.11 Optimizations with ExtType=20

This extension is like Coupling Via Frequency and Time with Priorities (see clause 7.4.1.2.3 and 7.5.7), but more efficient and more widely applicable. In that approach, if there is C-plane application layer packet fragmentation, each C-Plane message shall contain the highest priority section descriptions referring to any RE that is referred in a message (see 5.4.1.2.3 item #3). This may lead to a lot of fronthaul and processing overhead.

Ext-20 optimizes the method of specifying the overlapping RE information (puncturing information) with reduced fronthaul and processing overhead. This extension contains only the puncturing information (specified via Resource Element masks) of the channels without any beamforming information. It further consolidates the puncturing information of several channels into a single extension. If there is C-plane application layer packet fragmentation, beamforming information needs to be sent only once and only puncturing information (which is compact) needs to be repeated in every C-plane message.

Like Coupling Via Frequency and Time with priorities, using this extension simplifies the section description (e.g. details of RE allocation in PRB conveyed in reMask may be simplified by setting reMask to all ones). This can significantly reduce the number of sections required.

This extension can be used with all coupling methods as described in clause 7.4.1.2, thereby benefiting all coupling methods with reduced number of sections.

7.6 O-RUs per endpoint and per C-Plane message limits

This section is introduced to specify details of O-RU processing limits specified by per endpoint limits and per C-Plane limits. Whether one or both limits apply to the O-RU is based on the O-RUs processing architecture i.e. if the O-RU is endpoint processing based or has additional per C-Plane processing prior to endpoint processing.

7.6.1 O-RU per endpoint processing limits

When O-RUs processing granularity is endpoint based i.e. processing resources in O-RU to handle CU-Plane messages are allocated per endpoint, certain limits may be imposed by the O-RU per endpoint e.g. endpoint-section-capacity, endpoint-beam-capacity, endpoint-prb-capacity. When such limits are imposed by the O-RU, the O-DU shall comply with the limits, otherwise the stated O-RU capacity may be compromised.

7.6.2 O-RU C-Plane message limits

For O-RUs with per C-Plane message processing limits in addition to per endpoint processing limits, the O-RU can choose to advertise its limitations on a per C-Plane message basis. O-DU can choose to indicate that if it adheres to the associated limitations, otherwise the stated O-RU capacity may be compromised. The defined limits are the maximum number of beams and maximum number of highest priority sections within a C-Plane message. Refer clause 15.8 of the M-Plane Specification [7] for details on usage of this feature for various scenarios.

8 U-plane Protocol

8.1 General

8.1.1 U-plane Transport

Either eCPRI or IEEE 1914.3 is used as an encapsulation mechanism for the user-plane messages. Due to the nature of these messages (very strict delay constraints), it is assumed that message acknowledgements are not possible. Likewise, different data flows may be used for the User-Plane and Control-Plane messages.

8.1.2 U-plane Data Compression

U-Plane IQ data, (both DL and UL) including user data, PRACH and control channels may be transmitted in compressed format. There are several envisioned compression methods including an "uncompressed" format. The method of compression is variable based on sectionId but is constant for every U-Plane data section referenced in a single C-Plane message. Likewise, the I and Q bitwidth of the compressed samples is variable based on sectionId but is constant for every U-Plane data section referenced in a single C-Plane message. The block compression methods are performed on Physical Resource Block (PRB) basis (i.e. 12 x Resource Elements per PRB). More specific details on the supported compression methods may be found in Annex A.

A static IQ format and compression method, in which case the M-Plane configures the static IQ format (bitwidth) and compression method, is also defined. In this case there are four fewer bytes in the U-Plane section header, at the cost of less flexibility. For static IQ data format and compression method, there are actually **four** sets of data formats and compression methods: for LTE DL, LTE UL, NR DL and NR UL. This allows the taking advantage of the static format while still allowing some compression flexibility e.g. allowing modulation compression on the DL (which only applies to DL) and some other compression method on the UL.

8.1.3 Digital Power Scaling

8.1.3.1 Definition of IQ Power in dBFS

IQ power level in dBFS (dB full scale) is a logarithmic representation of the power level for an IQ sample carried over the digital interface. IQ power level in dBFS is proportional to logarithm of I^2+Q^2:

IQ power level [dBFS] = $10 \cdot \log_{10}(I^2+Q^2) - 10 \cdot \log_{10}(FS) = 10 \cdot \log_{10} (I^2+Q^2) - 10 \cdot \log_{10}(FS_0 \cdot 2^{-FS\_Offset})$

where

-*FS_Offset* is an M-Plane parameter (value 0 is used if this parameter is not supported by O-RU or not set by O-DU), $FS_0 = \max(I^2) = \max(Q^2) = \max(I^2+Q^2)$ with max over all IQ values that can be represented by IQ data format in U-plane message. Note that actual IQ values that may occur in a U-plane message are restricted by

$$I^2 + Q^2 \leq FS = FS_0 \cdot 2^{-FS\_Offset}.$$

For frequency domain IQ data, 0 dBFS is the maximum power level which can be carried by one subcarrier. The smallest non-zero IQ power level is defined by the interface resolution.

It is expected that an O-RU will normalize any received DL value to its internal representation of full scale so that a 0 dBFS can be properly handled.

Example 1:

FS_Offset=0

I=min I, Q=0

With 9bit mantissa 2's complement + 4bit exponent compression: $\min I = -256 \cdot 2^{15} = -2^{23}$ →

$FS_0=2^{46}$, $FS=FS_0 \cdot 2^{FS\_Offset}=2^{46}$

0 dBFS ⇔ average(I^2+Q^2) = 2^{46}

Interface resolution ⇔ $1/2^{46}$ ⇔ -138.47dBFs

Example 2:

FS_Offset=10

I=min I, Q=0

With 14bit mantissa 2's complement + 4bit exponent compression: min I = $-2^{13}\cdot2^{15} = -2^{28}$ →

$FS_0=2^{56}$, $FS=FS_0\bullet2^{-FS\_Offset}=2^{56-10}=2^{46}$

0 dBFS ⇔ average(I^2+Q^2) = 2^{46}

Interface resolution ⇔ $1/2^{46}$ ⇔ -138.47dBFs

8.1.3.2 Definition of Gain over Fronthaul Interface

The gain of an array defines the relation between the levels of a test signal seen at its input and output, also called digital power scaling. The gain of an array can be calculated from the gain of one element of the array while assuming all elements have same gain**. Figure 8-1** depicts the gain relations between the digital interface and the RF reference plane to an assumed lossless antenna (i.e., antenna insertion losses are counted as part of the gain in both DL and UL direction).

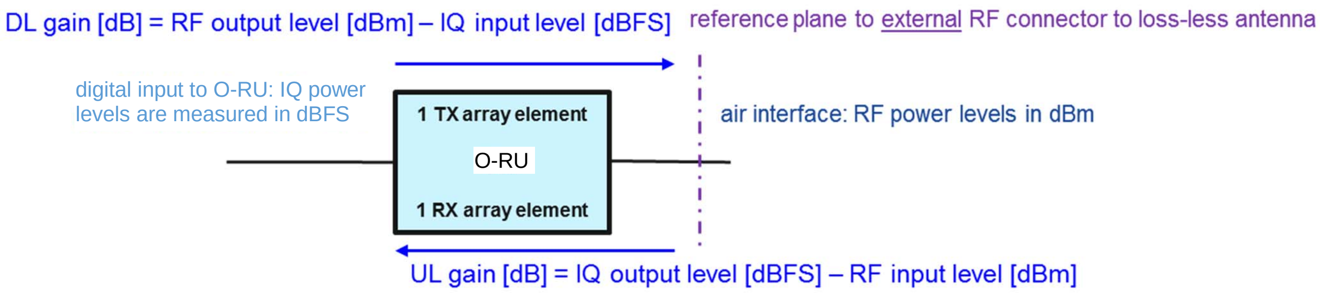

DL gain [dB] = RF output level [dBm] – IQ input level [dBFS]
reference plane to external RF connector to loss-less antenna
digital input to O-RU: IQ power levels are measured in dBFS
1 TX array element
O-RU
1 RX array element
air interface: RF power levels in dBm
UL gain [dB] = IQ output level [dBFS] – RF input level [dBm]

Figure 8-1 : DL and UL Gain over fronthaul interface

Both DL and UL gains in [dB] are described by the following relations

DL gain [dB] = RF output level [dBm] – IQ input level [dBFS]

UL gain [dB] = IQ output level [dBFS] – RF input level [dBm]

Where:

'RF output level [dBm]' and 'RF input level [dBm]' are RF signal level in dBm at reference plane to an assumed lossless antenna when the antenna is integrated inside the O-RU, or at TAB connector if antenna is not integrated inside the O-RU. The rms output power [dBm] is measured per array element as Total Radiated Power TRP (i.e., excluding any antenna directivity gain). The rms input power [dBm] is measured per array element after including the antenna directivity gain. Note that the antenna directivity gain results from the array element directivity + $10\cdot\log_{10}$(number of elements). Hence, when evaluating the array element gain, the antenna directivity is reduced to the array element directivity.

'IQ input level [dBFS]' and 'IQ output level [dBFS]' are $10\cdot\log_{10}$(average(I^2+Q^2)) normalized such that 0dBFS is the maximum nominal (r.m.s.) power level which can be achieved with a constant IQ signal with arbitrary phase (i.e., for frequency-domain IQ signal for **one sub-carrier,** constant over time).

In O-RUs supporting beamforming, the actual DL gain and UL gain of the array element can be impacted by the gain level of beam weight used and which can change dynamically during operation. **Such dependency on the beam pattern shall be excluded when describing any configurable gain or reported gain (as O-RU capability).**

For an exact definition of DL and UL gain, the respective input test signals need to be declared by the O-RU vendor; this is not in scope of this specification. [For information only: This can be e.g., CW signal at carrier center or test signals/reference signals defined by 3GPP; for DL gain those being defined to test accuracy of the maximum output power (3GPP TS36.141 clause 6.2→ 38.141) and for UL gain those being defined to test absolute accuracy of Received Interference Power (3GPP TS 36.133 clause 10.1.1 → 38.133)]. In **Figure 8-2** the details of DL gain are described. For every tx-array, O-RU reports (as capability) the maximal configurable DL gain of **one element of the array** (alternatively the nominal power per tx-array element mapped to 0dBFS). In addition, the O-DU configures over the M-Plane the DL gain to be used per tx-array element for a certain carrier configuration.

The values for the max configurable DL gain and configured DL gain assume:

- No power loss/gain due to beamforming weights;

- all available DL power can be allocated to one RE in a single eAxC (if dynamic range optimization is used by configuring non-zero value in M-plane parameter Reference_Level then single RE can have allocated available DL power less Reference_Level value; see clause 8.1.3.3);

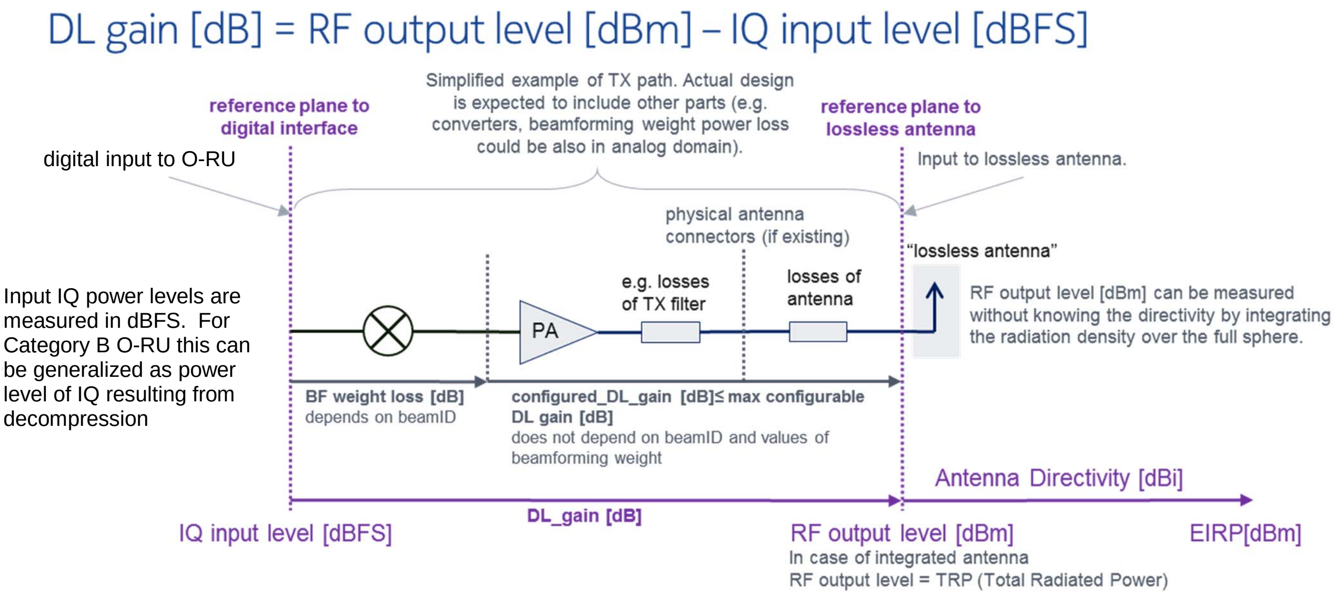

DL gain [dB] = RF output level [dBm] – IQ input level [dBFS]
Simplified example of TX path. Actual design is expected to include other parts (e.g. converters, beamforming weight power loss could be also in analog domain).
reference plane to digital interface
reference plane to lossless antenna
digital input to O-RU
Input to lossless antenna.
physical antenna connectors (if existing)
"lossless antenna"
Input IQ power levels are measured in dBFS. For Category B O-RU this can be generalized as power level of IQ resulting from decompression
e.g. losses of TX filter
losses of antenna
PA
RF output level [dBm] can be measured without knowing the directivity by integrating the radiation density over the full sphere.
BF weight loss [dB]
depends on beamID
configured_DL_gain [dB]≤ max configurable DL gain [dB]
does not depend on beamID and values of beamforming weight
Antenna Directivity [dBi]
DL_gain [dB]
IQ input level [dBFS]
RF output level [dBm]
In case of integrated antenna
RF output level = TRP (Total Radiated Power)
EIRP[dBm]

Figure 8-2 : Details of DL Gain

In **Figure 8-3** the details of UL gain are described. For every rx-array, O-RU reports (as capability) an UL gain_correction_range in dB of **one element of the array (applicable to all the elements of the array)**. The gain_correction_range is signed, has a max and min value, and a step size. In addition, the O-DU configures over the M-Plane a gain_correction value to be applied for contribution of signal received by array-element to IQ data in eAxC (for backward compatibility this is configured as a sum of a common gain correction applicable to all eAxCs of given array carrier and an individual gain correction of each eAxC). The O-RU can then configure its internal UL gain of the rx-array element for that carrier (rx-array carrier element) if the IQ compression method is configured as static. In case the compression method is dynamic, the internal gain of the O-RU will be dynamic and depending on compression information received in realtime over the C-Plane.

The values for the configured UL gain assume:

- No power loss/gain due to beamforming weights.

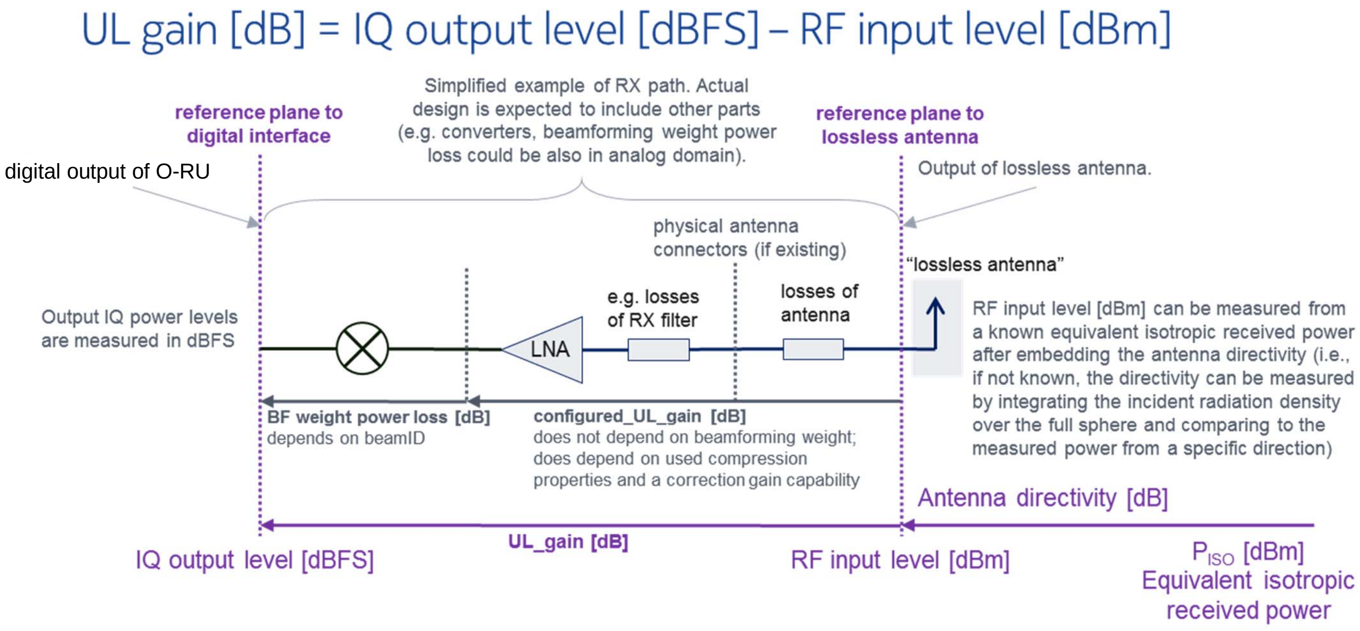

UL gain [dB] = IQ output level [dBFS] – RF input level [dBm]
reference plane to digital interface
Simplified example of RX path. Actual design is expected to include other parts (e.g. converters, beamforming weight power loss could be also in analog domain).
reference plane to lossless antenna
digital output of O-RU
Output of lossless antenna.
physical antenna connectors (if existing)
"lossless antenna"
Output IQ power levels are measured in dBFS
e.g. losses of RX filter
losses of antenna
LNA
RF input level [dBm] can be measured from a known equivalent isotropic received power after embedding the antenna directivity (i.e., if not known, the directivity can be measured by integrating the incident radiation density over the full sphere and comparing to the measured power from a specific direction)
BF weight power loss [dB] depends on beamID
configured_UL_gain [dB] does not depend on beamforming weight; does depend on used compression properties and a correction gain capability
Antenna directivity [dB]
UL_gain [dB]
IQ output level [dBFS]
RF input level [dBm]
P_ISO [dBm] Equivalent isotropic received power

Figure 8-3 : Details of UL Gain

8.1.3.2.1 DL Gain Guideline

The DL gain of a TX array carrier element shall be provided as part of the O-RU carrier set-up procedure. The configured (by the O-DU over the M-Plane) DL gain of a TX array carrier element (i.e. "TX array carrier element'' refers to the TX array element serving the respective carrier configured on the respective TX array) is defined by mapping Reference_Level dBFs (see clause 8.1.3.3) carried over any subcarrier (of the carrier) on the digital interface to the maximum RMS power the TX array carrier element is supposed to output at the assumed-lossless antenna connector port (i.e., including all insertion losses). In single-carrier operation, the configured DL gain applies to the carrier. In multiple-carrier operation, the configured DL gain should be lower accommodating the number of multiple carriers and their bandwidths. It is the responsibility of the O-DU to scale down the level of the signals at the interface and/or set tx-array-carrier gain values to avoid saturation in the O-RU for single or multiple-carrier configurations.

Configured DL_gain [in dB] ~≤ maximum TX power per array element [in dBm] – Reference_Level [in dBFS], valid for each individual spatial stream served under this TX array carrier element.

NOTE 1: the maximum configured DL gain of a TX array carrier is equivalent to mapping 0dBFs carried over any subcarrier (of the carrier) on the digital interface to the maximum RMS power the TX array carrier element is supposed to output + $10\cdot\log_{10}$ (the number of array elements) – Reference_Level [in dBFS].

NOTE 2: for a category A O-RU, the tx-array output rms power is measured over 1 polarization

NOTE 3: for a category B O-RU, the tx-array output rms power shall be measured over all polarizations addressed by the respective precoding operations

NOTE 4: Configured DL_gain [in dB] = target TX carrier power per array element [in dBm] - Reference_Level [in dB]

where target TX carrier power per array element ≤ maximum TX power per array element.

8.1.3.2.2 UL Gain Definition

The UL gain or scaling of an rx-array carrier element (i.e. "rx-array carrier element'' refers to the rx-array element serving the respective eAxC configured on the respective rx-array) is defined by mapping -152dBm at the assumed-lossless antenna port to the smallest power level an IQ sample can carry over the digital interface (i.e., average(I^2+Q^2) = 1) while considering a configured gain_correction value and the IQ compression properties to be used; configured gain_correction is sum of RX carrier specific gain_correction and eAxC specific gain _correction. In addition, in order to avoid saturation over the interface when beamforming is used over the rx-array carrier, the largest power level that can be received at the assumed-lossless antenna port by the rx-array carrier element is equivalent to 0dBFS -

$10 \cdot \log_{10}$(number of array elements). The configured gain-correction allows to adjust the level of the smallest & largest receivable power.

The following principles apply:

1. the UL gain of an rx-eAxC is equal to the UL gain of the rx-array carrier element, and contributions to IQ in the eAxC from all rx-array elements are considered to have the same gain
2. the minimum and maximum power levels that can be received by the rx-eAxC are scaled by $+10 \cdot \log_{10}$(number of array elements) from the respective values of the rx-array carrier element (i.e. when signal is received with equal levels from all the elements of the rx array meaning no tapering used and all rx elements have same design properties). In case an rx-array has only one rx-array element, then the scale factor is unity. The reason behind this principle is to allow applying tapering (beam ID dependent) to an rx-array without impacting the quality of the IQ data being transferred over the interface (i.e., in a worst case scenario, the signal is received from only one rx-array element from the rx-array).
3. assumed-lossless antenna port means antenna losses are considered part of the unit under test.
4. the -152dBm is considered as the smallest level that can be measured by a narrow subcarrier of 1.25kHz (i.e., CW tone) for a system with 3dB of equivalent noise figure and without being impacted by the interface noise (i.e. 20dB margin considered).
5. Configured_UL_gain [in dB] = Interface resolution [dBFs] - (-152 dBm) + gain_correction [dB]; valid for each received individual spatial stream (i.e. configured_UL_gain is configured by O-RU and not by the O-DU).

The UL gain depends on the digital interface resolution (in dBFS) representing the smallest level that can be used. The interface resolution depends on the compression scheme which can be static or dynamic. For this reason, when multiple compression methods or IQ bitwidths are used for data streams received from an rx-array carrier element, the configured gain shall accommodate all the intended compression methods and IQ bitwidths.

For example, when using block floating point compression, the interface resolution is defined as

Interface resolution [dBFs]$=-20 \times \log_{10}\left(2^{Mantissa_{bits}-1} \times 2^{2^{Exponent}-1}\right)$

It is assumed that the gain_correction is 0dB by default, unless a different value is needed due to special circumstances (details beyond the scope of this specification).

The gain_correction is configured as one value per rx array carrier and does not change once an array carrier is activated. The O-RU applies a configured value of gain_correction regardless of used compression.

Example with rx-array formed by 10 array elements and gain_correction of 0dB:

9bit mantissa 2's complement + 4bit exponent → Interface resolution=-138.5dBFS

Default UL gain [dB]=-138.5dBFs + 152dBm +0dB =13.5dB

The UL gain and power scaling are summarized in **Table 8-1.**

Table 8-1 : Example of UL gain and power scaling for an rx array with 10 elements and for block floating point compression based on 9-bit mantissa and 4-bit exponent

Interface resolution -138.5dBFS	*rx-array carrier element*	*Rx-array carrier with 10 elements*
UL Gain	*13.5dB*	*13.5dB*
Lowest received power level without degradation due to interface (excluding losses due to beam weights)	*-152dBm → -138.5dBFS*	*-142dBm → -128.5dBFS*
Largest possible received power level	*-23.5dBm → -10dBFS*	*-13.5dBm → 0dBFS*

8.1.3.3 TX Power Budget Guideline for Category A and Category B O-RUs

This clause describes a guideline for handling of power budget in a category A and category B O-RU. In general, care shall be taken to avoid exceeding the maximum rms power rating of a tx-array element.

In this clause, $m_{\text{a,k}}$ is the maximum rms power rating (in W) of tx-array element *k* of a tx-array *a* with *K* elements. For simplicity it is assumed all *K* elements of the array have the same maximum rms power rating (i.e., for every array *a* and every k and k', $m_{\text{a,k}} = m_{\text{a,k}'}$). **For an O-RU with tx-array *a*, the maximum rms power rating of tx-array element shall be reported as read-only parameter. This will be a common value for all array elements of the tx-array *a*.**

NOTE 1: maximum-power-rating can be reported as form of gain when mapped to 0dBFS.

The maximum rms power rating of array can be derived from $m_{\text{a,k}}$ by scaling linearly with the number of elements *K* (i.e., $m_{\text{a}}=K\cdot m_{\text{a,k}}$)

In addition, let $g_{c,a,k}$ be the gain (in dB) configured for tx-array element k of tx-array a for array carrier c. It is considered that the configured gain is same to every element (i.e., for every tx-array carrier c, every tx-array a and every element k and k', $g_{c,a,k} = g_{c,a,k'}$). **Hence, the O-DU should configure the O-RU by providing a value for $g_{c,a,k}$ in parameter 'gain' of tx-array-carrier.**

The total gain of array carrier *c* served on array *a* can be derived from the configured gain $g_{c,a,k}$ as:

$$g_{c,a}\,[\text{dB}]= g_{c,a,k}\,[\text{dB}]+10\cdot\log_{10}(K).$$

Hence, the power $\text{p}_{c,a,k}$ of array carrier *c* served on tx-array element *k* of array *a* can be derived as

$$\text{p}_{c,a,\text{k}}\,[\text{dBm}]= g_{c,a,k}\,[\text{dB}]+0\text{dBFS}.$$

As a result, the total power of $\text{p}_{c,a}$ of array carrier *c* served on tx-array *a* is derived as

$$\text{p}_{c,a}\,[\text{dB}]= \text{p}_{c,a,k}\,[\text{dB}]+10\cdot\log_{10}(K).$$

Finally, the total power $\text{p}_{a.\text{k}}$ used on tx-array element *k* of tx-array *a* serving all configured carriers is derived as

$$p_{a,k} = 10\cdot\log_{10}\sum_{c} 10^{\frac{p_{c,a,k}}{10}}.$$

In scenarios where a tx-array element k is shared between multiple tx-arrays, the O-DU shall ensure that the configured gains $g_{c,a,k}$ are constrained by:

$$\sum_{a}\sum_{c} 10^{\frac{g_{c,a,k}+Reference\_Level_c}{10}} \leq 1000\cdot m_k$$

where $m_k = \min_a(m_{a,k})$ and *Reference_Level*$_{\text{c}}$ (in dB) is the array-carrier specific IQ normalization level optionally configured via M-plane per array carrier (the value 0 is used if this parameter is not supported by the O-RU or not configured by the O-DU). The index *a* spans over every tx-array *a* that shares array element *k* and has array carrier configured. The summing over *c* includes every array carrier *c* that is configured for tx-array *a*.

NOTE 2: in general $\text{m}_{a,k}$ can be different for different values of tx-array a. However, tx-arrays not used in a given configuration do not contribute to the above constraint.

For every configured array carrier *c* over a tx-array *a*, the O-DU shall ensure that the input power levels of all resource elements used over all eAxCs x are constrained by:

$$\sum_{x}\sum_{n} 10^{\frac{RE_{n,x,c,a}^{dBFS}}{10}} \leq 10^{\frac{Reference\_Level_c}{10}}$$

and

$$RE_{n,x,c,a}^{dBFS} \leq 0$$

where $RE_{n,x,c,a}^{dBFS}$ represents input power level in dBFS of an RE *n* of an eAxC *x* of array carrier *c* configured over tx-array a. Note $RE_{n,x,c,a}^{dBFS} = 10\cdot\log_{10}\left({I_{n,x,c,a}}^2 + {Q_{n,x,c,a}}^2\right) - 10\cdot\log_{10}(FS_0\cdot 2^{-FS\_Offset})$ where $I_{n,x,c,a}$ and $Q_{n,x,c,a}$ are decompressed IQ sample value components, FS_0 and *FS_Offset* are defined in clause 8.1.3.1. The summing over *x* includes every eAxC *x* of array carrier *c* that is used simultaneously in DL on the tx-array a. The summing over *n* includes every RE of the eAxC *x* that is used simultaneously in DL.

When beamforming is used in O-RU, an additional constraint applies to all the beamforming weights to be used for beamforming in order to ensure that the tx power per tx-array element after the beamforming has well-defined upper bound and does not exceed the limit of the maximum rms power rating of the tx-array element. To simplify the notation and cover all beamforming types, i.e. frequency-domain beamforming and time-domain beamforming, any individual beamforming weight (a complex number multiplier used by O-RU in beamforming operation) is denoted as *w*.

For any beamforming weight *w* to be used by O-RU to perform beamforming operations, the entity controlling the generation of the weight (i.e., O-DU or O-RU) shall ensure that

$$|\boldsymbol{w}|^2 \leq \mathbf{1}.$$

Note that this constraint is applicable to every beamforming weight, used in frequency or time domain. The constraint is applicable to every explicit or implied beamforming weight applied by O-RU regardless of the source of the beamforming weight (e.g. received from O-DU, predefined by O-RU, generated by O-RU from beam attributes or from channel information, etc.).

8.2 Elementary Procedures

8.2.1 IQ Data Transfer procedure

This procedure (an alternative procedure IQ data transfer is described in clause 8.2.2) is used to transfer frequency domain IQ data samples between the O-DU and O-RU. Data is transmitted symbol by symbol as U-Plane messages. The data-associated control information is typically sent every slot (or for LTE, TTI) in a different data flow with its own header and encapsulated payload (C-Plane messages). An overview of the logic for transmission of both the data-associated control information and IQ data is depicted in **Figure 8-4**.

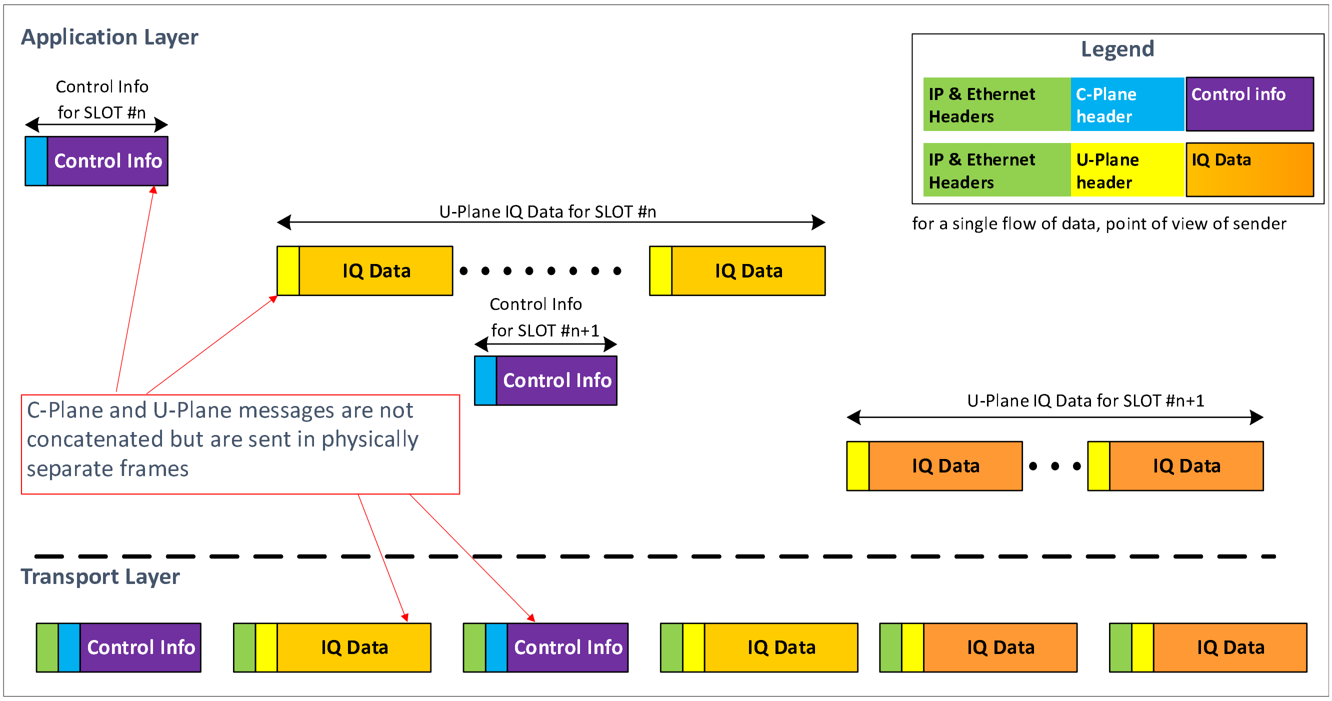


Figure 8-4 : DL IQ data transfer overview

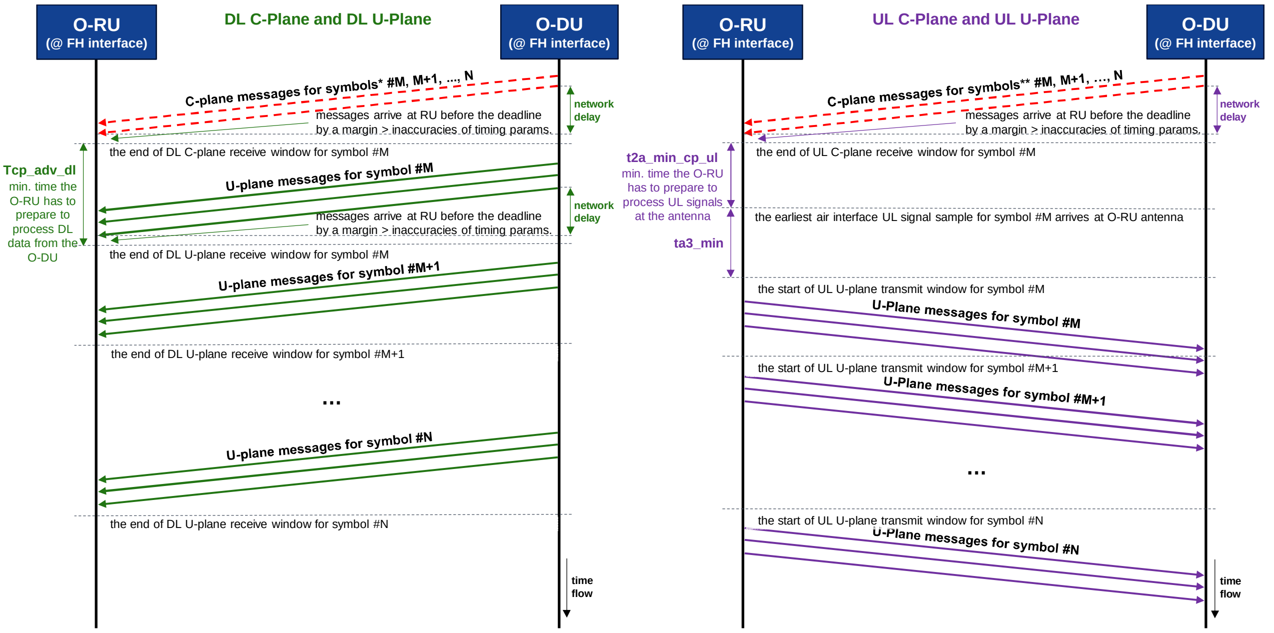
The IQ transfer procedure is comprised of two types of messages:

- Message containing scheduling and beamforming commands information (i.e. data-associated control)
- Message containing frequency domain IQ samples

The data-associated control information is bundled in messages within a single data flow (albeit possibly in multiple C-Plane messages). However, control information for UL and DL is sent separately (See **Figure 8-5**). As shown in the figure, the C-Plane messages for a given slot and given symbol(s) exchanged between the O-DU and O-RU are followed by the U-Plane data for that slot and symbol(s), one symbol at a time. U-plane messages are sent by the O-DU and O-RU in order of the symbol for which they carry IQ data. C-Plane and DL U-Plane messages are sent by the O-DU in advance (see 2.3 for details) such that they arrive at O-RU within a time window that is early enough to leave the O-RU time to process them. **Figure 6-5** shows the case where O-DU sends DL C-Plane message describing symbols

#M, M+1,…, N of slot S between “T1a_max_cp_dl” and “T1a_min_cp_dl” before the start of DL symbol #M (the earliest symbol described by the message). As shown in the figure, the end of receive time window for DL C-Plane messages describing symbol #M, M+1, …, N of slot S (where M is the earliest symbol described by each of messages) is by “Tcp_adv_dl” earlier than the end of receive time window for DL U-Plane messages carrying IQ data for symbol #M. The O-DU sends and O-RU receives DL U-Plane message as per the transmission window and reception window specified in clause 4.3. Similarly, the O-DU sends UL C-Plane messages describing symbols #M, M+1, …, N of slot S between “T1a_max_cp_ul” and “T1a_min_cp_ul” before the start of the UL symbol #M (the earliest symbol described by the message). The end of receive time window for UL C-Plane messages describing symbol #M, M+1, …, N (where M is the earliest symbol described by each of messages) is by “T2a_min_cp_ul” earlier than the start of UL symbol #M. The O-RU sends and O-DU receives UL U-Plane message as per the transmission window and reception window specified in clause 4.3. For non-PRACH channels the O-RU sends UL U-Plane message carrying IQ data for symbol #M between “Ta3_min” and “Ta3_max” after the reception of the earliest air interface sample of UL symbol #M (this corresponds to start of cyclic prefix). For PRACH channels the O-RU sends UL U-Plane messages carrying IQ data for a repeatable part of PRACH sequence between “Ta3_min” and “Ta3_max” after the reception of the earliest air interface sample that is necessary to generate the IQ data in the message (this corresponds to the start of the FFT sampling window).

Also shown in the figure is the fact that there is a certain period of time between the O-RU’s receipt of C-Plane messages for a symbol and the need for the O-RU to process U-Plane data for that symbol. In particular, in the DL there is a period of time, “Tcp_adv_dl” which provides for some number of microseconds for the O-RU to e.g. update beamforming weights prior to processing the DL data arriving from the O-DU. In the UL there is a period of time, “t2a_min_cp_ul” between the O-RU’s receiving the C-Plane messages governing the processing of UL data and the receipt of UL signals at the O-RU’s antennas. These time intervals, when combined with network delays and other processing latencies, result in the RAN’s HARQ loop to be closed allowing feedback in the air interface processing.


O-RU
(@ FH interface)
DL C-Plane and DL U-Plane
O-DU
(@ FH interface)
C-plane messages for symbols* #M, M+1, …, N
network delay
messages arrive at RU before the deadline by a margin > inaccuracies of timing params.
the end of DL C-plane receive window for symbol #M
Tcp_adv_dl
min. time the O-RU has to prepare to process DL data from the O-DU
U-plane messages for symbol #M
network delay
messages arrive at RU before the deadline by a margin > inaccuracies of timing params.
the end of DL U-plane receive window for symbol #M
U-plane messages for symbol #M+1
the end of DL U-plane receive window for symbol #M+1
…
U-plane messages for symbol #N
the end of DL U-plane receive window for symbol #N
time flow
O-RU
(@ FH interface)
UL C-Plane and UL U-Plane
O-DU
(@ FH interface)
C-plane messages for symbols** #M, M+1, …, N
network delay
messages arrive at RU before the deadline by a margin > inaccuracies of timing params.
the end of UL C-plane receive window for symbol #M
t2a_min_cp_ul
min. time the O-RU has to prepare to process UL signals at the antenna
the earliest air interface UL signal sample for symbol #M arrives at O-RU antenna
ta3_min
the start of UL U-plane transmit window for symbol #M
U-Plane messages for symbol #M
the start of UL U-plane transmit window for symbol #M+1
U-Plane messages for symbol #M+1
…
the start of UL U-plane transmit window for symbol #N
U-Plane messages for symbol #N
time flow

*) A DL C-plane message describing multiple symbols must arrive at O-RU at least **Tcp_adv_dl** before the end of DL U-plane receive window for symbol **startSymbolId** (the earliest symbol described by the message).

) A UL C-plane message describing multiple symbols must arrive at O-RU at least **t2a_min_cp_ul before the earliest air interface UL signal sample of symbol **startSymbolId** (the earliest symbol described by the message) arrives at O-RU antenna.

Figure 8-5 : C-Plane and U-Plane message transfer procedure (DL & UL shown)

Regardless of the level of compression, user-plane data may exceed the maximum packet size (e.g., MTU of 1500 Bytes for regular not Jumbo frames) even in case of one symbol of granularity. Thus, IQ data for each symbol may require packetization over multiple packets.

User plane messages are sent as resource blocks (“PRBs”) and the data for each PRB shall start on a byte boundary. If due to the IQ bit-width being used the natural end of the data in the PRB does not fall on a byte boundary, then zero bits shall be appended until a byte boundary is reached.

PRACH data is handled similarly, such that the PRACH REs are packaged into 12-RE blocks analogous with data PRBs. Each 12-RE block shall be padded with zero bits to end on a byte boundary. In cases when there is not an even factor of 12 REs in the PRACH data (e.g. 839 PRACH REs), added zero-value REs shall be added to fill out the block to 12 REs, and then if needed zero-padding will then be appended to reach a byte boundary.

8.2.2 IQ Data Transfer Procedure Without C-Plane

This is a procedure used to transfer frequency domain IQ data samples between the O-DU and O-RU without C-plane control. In respect to IQ data transfer this procedure follows the rules described in clause 8.2.1 with an exception: data-associated control information for selected eAxCs is not transferred in C-Plane messages but configured via M-Plane. This method is applicable to channels with predictable scheduling (like PRACH or SRS) which do not require changes of data-associated control information in real-time.

At present this procedure is applicable to eAxCs assigned to receive a subset of uplink signals and channels – specifically PRACH and SRS (see clause 8.3.5.3 and clause 8.3.5.4):

1) O-DU provides via M-Plane data-associated control information (this consists of scheduling and beamforming configuration) to O-RU and activates the eAxC. C-plane messages are not sent for this eAxC.
2) O-RU periodically receives uplink radio signal following the configured schedule and sends U-plane messages containing frequency domain IQ data (as in clause 8.2.1). Note values of fields in U-plane message headers and data section headers are derived from parameters configured via M-plane as per rules described in M-plane specification (see [7]).

For more details on parameters that convey scheduling and beamforming configuration and rules for mapping them to U-plane messages refer to M-Plane specification (see [7]).

8.3 Elements for the U-plane Protocol

8.3.1 General

U-Plane messages are encapsulated using a two-layered header approach. The first layer consists of an eCPRI or IEEE 1914.3 common header, including fields used to indicate the message type, while the second layer is an application layer including necessary fields for control and synchronization. When the optional “little endian byte order” is chosen via M-plane, the iSample/qSample fields shall use little endian byte order to transmit the complex numbers. Annex D.2 shows little endian byte order format for various I/Q data length.

8.3.2 DL/UL Data

A common frame format is used for U-Plane messages consisting of a transport layer and an application layer. The application layer is within the transport payload and consists of a common header for time reference, followed by information and parameters dependent and specific to the Section Type in use. Data from multiple sections of the same value can be lined up one after another within the payload. To minimize packet rate over the interface, transmitter should fill messages with as many subsequent sections as possible. Data from sections of different Section Types can be mixed within a single U-Plane message for a given eAxC when the Section Types do not have differences in format and values of user plane application header or format of user plane section header. NOTE: Previous versions of the specification mandated separate U-plane messages to be used with different Section Types, this restriction has been relaxed. Furthermore, whenever necessary, stuffing bits are to be added at the end of a sections after (possibly compressed) I and Q sample data to achieve 1-Byte alignment. Note that within a resource block, unless and sReSMask is included in the udCompParam field, 12 REs are always sent across the interface. If some REs are meant to be blank some data (perhaps zero value) shall be sent anyway because the U-Plane data parser is expecting exactly 12 complex RE values per resource block.

- Transport Layer – see clause 5.1.3
- Application Layer
- Section Type “0” Fields (used for indicating idle or guard periods from O-DU to O-RU):
    - No U-plane messages are associated with Section Type “0”
- Section Type “1” Fields (used for most Downlink and Uplink physical radio channels):
  - Common Header Fields
    - **dataDirection** (data direction (gNB Tx/Rx)) field: 1 bit
    - **payloadVersion** (payload version) field: 3 bits

- Value = “1” shall be set (1st protocol version for payload and time reference format)
- **filterIndex** (filter index) field: 4 bits
- **frameId** (frame identifier) field: 8 bits
- **subframeId** (subframe identifier) field: 4 bits
- **slotID** (slot identifier) field: 6 bits
- **symbolId** (symbol identifier) field: 6 bits

- <u>Section header fields</u>
  - **sectionID** (section identifier) field: 12 bits
  - **rb** (resource block indicator) field: 1 bit
  - **symInc** (symbol number increment command) field: 1 bit
  - **startPrbu** (starting PRB of user plane section) field: 10 bits
  - **numPrbu** (number of contiguous PRBs per data section) field: 8 bits
  - **udCompHdr** (user data compression header) field, not always present: 8 bits
  - **reserved** (reserved for future use) field, only present with udCompHdr: 1 byte
  - **udCompLen** (PRB field length) field, not always present: 16 bits

- <u>PRB fields</u>
  - **udCompParam** (user data compression parameter) field: 0, 8 or 16 bits
  - **iSample** (in-phase sample) field: 1-16 bits
  - **qSample** (quadrature sample) field: 1-16 bits

- Section Type “3” Fields (used for PRACH and mixed-numerology channels**):
  - Timing header, section header and PRB fields same as for Section Type “1”
- Section Type “5” Fields (used for UE scheduling information):
  - Timing header, section header and PRB fields same as for Section Type “1”
- Section Type “6” Fields (used for sending channel information for a specific UE ID):
  - No U-plane messages are associated with Section Type “6”

**NOTE: Section Type “1” message can also be used for PRACH channel. When Section Type “1” message is used for PRACH, the UL U-plane processing should apply same rules as a non-PRACH channel in O-RU.

Table 8-2 : IQ data frame format

<table>
<tr><th colspan="10">Section Type 1,3 : DL/UL IQ data msgs</th></tr>
<tr><th>0 (msb)</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7 (lsb)</th><th># of bytes</th><th></th></tr>
<tr><td colspan="8">transport header, see clause 5.1.3</td><td>8</td><td>Octet 1</td></tr>
<tr><td>dataDirection</td><td colspan="3">payloadVersion</td><td colspan="4">filterIndex</td><td>1</td><td>Octet 9</td></tr>
<tr><td colspan="8">frameId</td><td>1</td><td>Octet 10</td></tr>
<tr><td colspan="4">subframeId</td><td colspan="4">slotId</td><td>1</td><td>Octet 11</td></tr>
<tr><td colspan="2">slotId</td><td colspan="6">symbolid</td><td>1</td><td>Octet 12</td></tr>
<tr><td colspan="8">sectionId</td><td>1</td><td>Octet 13</td></tr>
<tr><td colspan="4">sectionId</td><td>rb</td><td>symInc</td><td colspan="2">startPrbu</td><td>1</td><td>Octet 14</td></tr>
<tr><td colspan="8">startPrbu</td><td>1</td><td>Octet 15</td></tr>
<tr><td colspan="8">numPrbu</td><td>1</td><td>Octet 16</td></tr>
<tr><td colspan="8">udCompHdr (not always present)</td><td>0/1</td><td>Octet 17</td></tr>
<tr><td colspan="8">reserved (not always present)</td><td>0/1</td><td>Octet 18</td></tr>
<tr><td colspan="8">udCompLen (not always present)</td><td>0/2</td><td>Octet 17/19</td></tr>
</table>

udCompParam (not always present)				0/1/2	Octet 17/19/21
iSample (1st RE in the PRB)				1*	K= 17/19/20/21/23
qSample (1st RE in the PRB)				1*	K+1*
…					
iSample (12th RE in the PRB)				1*	K+22*
qSample (12st RE in the PRB)				1*	K+23*
udCompParam (not always present)				0/1/2	K+24*
iSample (1st RE in the PRB)				1*	K+24/25/26*
qSample (1st RE in the PRB)				1*	K+25/26/27*
…					
iSample (12th RE in the PRB)				1*	K+46/47/48*
qSample (12st RE in the PRB)				1*	K+47/48/49*
…					
sectionId				1	Octet M
sectionId	rb	symInc	startPrbu	1	M+1
startPrbu				1	M+2
numPrbu				1	M+3
udCompHdr (not always present)				0/1	M+4
reserved (not always present)				0/1	M+5
udCompLen (not always present)				0/2	M+4/6
udCompParam (not always present)				0/1/2	M+4/6/8
iSample (1st RE in the PRB)				1*	K=M+4/6/7/8/10
qSample (1st RE in the PRB)				1*	K+1*
…					
iSample (12th RE in the PRB)				1*	K+22*
qSample (12st RE in the PRB)				1*	K+23*
udCompParam (not always present)				0/1/2	K+24*
iSample (1st RE in the PRB)				1*	K+24/25/26*
qSample (1st RE in the PRB)				1*	K+25/26/27*
…					
iSample (12th RE in the PRB)				1*	K+46/47/48*
qSample (12st RE in the PRB)				1*	K+47/48/49*

shading: yellow is transport header, pink is radio application header, others are repeated sections
*Octet count given if the iqWidth = 8 but other iqWidth values are possible; the octet count assumes all REs are present which may not be the case when field sReSMask is present in udCompParam.

8.3.3 UL/DL Data Coding of Information Elements

See clause 5.1.3 for transport header information element details.

8.3.3.1 dataDirection (data direction (gNB Tx/Rx))

See clause (7.4.4.1)

8.3.3.2 payloadVersion (payload version)

See clause (7.4.4.2)

8.3.3.3 filterIndex (filter index)

See clause (7.4.4.3)

8.3.3.4 frameId (frame identifier)

See clause (7.4.4.4)

frameId in U-plane message shall be set to the frameId value signaled in the corresponding C-plane message.

8.3.3.5 subframeId (subframe identifier)

See clause (7.4.4.5)

subframeId in U-plane message shall be set to the subframeId value signaled in the corresponding C-plane message.

8.3.3.6 slotId (slot identifier)

See clause (7.4.4.6)

slotId in U-plane message shall be set to the slotId value signaled in the corresponding C-plane message.

8.3.3.7 symbolId (symbol identifier)

Description: This parameter identifies a symbol number within a slot. When a C-plane message describes a single symbol (or in case of PRACH, a single PRACH repetition), symbolId in the U-plane message for that symbol (or PRACH repetition) shall be set to the startSymbolId value signaled in the C-plane message. When a C-plane message describes multiple symbols (or in case of PRACH, multiple PRACH repetitions), symbolId in the U-plane message for the first symbol (or PRACH repetition) shall be set to the startSymbolId value signaled in the C-plane message. symbolId in the U-plane message for the subsequent symbols (or PRACH repetitions) are incremented thereon.

Value range: {00 0000b-11 1111b}

Type: unsigned integer.

Field length: 6 bits.

8.3.3.8 sectionId (section identifier)

See clause (7.4.5.1)

8.3.3.9 rb (resource block indicator)

See clause (7.4.5.2)

8.3.3.10 symInc (symbol number increment command)

See clause (7.4.5.3)

8.3.3.11 startPrbu (startingPRB of user plane section)

Description: This parameter is the starting PRB (lowest frequency) of a user plane data section. Values of rb, startPrbu and numPrbu shall ensure that data sections shall never overlap: a single PRB (a block of 12 resource elements consecutive in frequency) may only exist within one data section for a given eAxC. For one section description in C-Plane message, there may be multiple U-Plane data sections associated with it and requiring defining from which PRB the contained IQ data are applicable. NOTE: freqOffset parameter conveyed in the corresponding section description affects the frequency span for specific range of PRB numbers (formulas in clause 7.4.5.6 apply with startPrbu replacing startPrbc). Therefore "shall never overlap" shall consider the value of freqOffset.

In general U-plane data section shall include only PRBs addressed by the corresponding C-plane section descriptions, but O-RU shall be prepared for possibility of lost C-plane or U-plane message. Depending on method of coupling of C-plane and U-plane more specific rules apply. If coupling of C-plane and U-plane via sectionId value is used then values of rb, startPrbu and numPrbu shall ensure only PRBs addressed by section descriptions with the same sectionId as the user plane data section are present in the user plane data section. If coupling of C-plane and U-plane via frequency and time (with or without priorities) is used then values of rb, startPrbu and numPrbu shall ensure only PRBs addressed by section descriptions are present in the user plane data section.

Note that due to fragmentation (an application level fragmentation to meet restriction of message size or a fragmentation of section in frequency, e.g. as a result of presence of Section Extension #6 of #12 in a corresponding section description) there may be multiple user plane data sections per a C-plane section description. Presence of rb = 1 (see 5.4.5.2) in C-plane section description requires presence of rb = 1 in user plane data section and does not require multiple user plane data sections.

Value range: {00 0000 0000b-11 1111 1111b}.

Type: unsigned integer.

Field length: 10 bits.

8.3.3.12 numPrbu (number of PRBs per user plane section)

Description: This parameter defines the number of PRBs (blocks of 12 resource elements consecutive in frequency) in the user plane data section. If the parameter rb (see 6.3.3.9) is zero, then the PRBs in the user plane data section shall be consecutive in the frequency. Otherwise the set of PRBs includes only every other PRB in the frequency. Values of rb, startPrbu and numPrbu shall ensure that data sections shall never overlap: a single PRB may only exist within one data section for a given eAxC. NOTE: freqOffset parameter conveyed in the corresponding section description affects the frequency span for specific range of PRB numbers (formulas in clause 7.4.5.6 apply with startPrbu replacing startPrbc). Therefore "shall never overlap" shall consider the value of freqOffset.

In general U-plane data section shall include only PRBs addressed by the corresponding C-plane section descriptions, but O-RU shall be prepared for possibility of lost C-plane or U-plane message. Depending on method of coupling of C-plane and U-plane more specific rules apply.

If coupling of C-plane and U-plane via sectionId value is used then values of rb, startPrbu and numPrbu shall ensure only PRBs addressed by C-plane section descriptions with the same sectionId as the user plane data section are present in the user plane data section. If coupling of C-plane and U-plane via frequency and time (with or without priorities) is used then values of rb, startPrbu and numPrbu shall ensure only PRBs addressed by C-plane section descriptions are present in the user plane data section.

Note that due to fragmentation (an application level fragmentation to meet restriction of message size or a fragmentation of section in frequency, e.g. as a result of presence of Section Extension #6 of #12 in a corresponding section description) there may be multiple user plane data sections per a C-plane section description. Presence of rb = 1 (see 5.4.5.2) in C-plane section description implies presence of rb = 1 in user plane data section and does not require multiple user plane data sections.

Value range: {0000 0001b-1111 1111b, 0000 0000b = all PRBs in the specified SCS and carrier bandwidth }.

Value 0000 0000b is reserved for NR cases wherein the total number of PRBs may be more than 255, for other cases a non-zero value of numPrbc shall be used. In this case, the sending node shall set the startPrbu value to zero and the receiving node shall ignore whatever startPrbu value is received and assume it is zero.

Type: unsigned integer.

Field length: 8 bits.

8.3.3.13 udCompHdr (user data compression header)

Description: This parameter defines the compression method and IQ bit width for the user data in a data section. This means that, in the DL at least, each data section can in principle have a different udCompHdr value. In the UL, the O-RU shall copy the received udCompHdr value in the C-plane message to the udCompHdr field in the UL U-plane message. This C-Plane instruction provides a single udCompHdr value for all data sections defined in the C-Plane message. This field is absent from U-Plane messages when the static IQ format and compression method is configured via the M-Plane. In this way a single compression method and IQ bit width is provided (per UL and Dl, per LTE and NR) without adding more overhead to U-Plane messages.

Value range: {0000 0000b-1111 1111b}

Bit allocations

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
udIqWidth				udCompMeth				1	Octet 1

Table 8-3 : udIqWidth definition

udIqWidth	Bit width of each I and each Q

0000-1111b	16 for udIqWidth=0, otherwise equals udIqWidth e.g. udIqWidth = 0000b means I and Q are each 16 bits wide;
	e.g. udIQWidth = 0001b means I and Q are each 1 bit wide;
	e.g. udIqWidth = 1111b means I and Q are each 15 bits wide

Table 8-4 : udCompMeth definition

udCompMeth	compression method	udIqWidth meaning
0000b	no compression	bitwidth of each uncompressed I and Q value
0001b	block floating point (BFP)	bitwidth of each I and Q mantissa value
0010b	block scaling	bitwidth of each I and Q scaled value
0011b	µ-law	bitwidth of each compressed I and Q value
0100b	modulation compression	bitwidth of each compressed I and Q value
0101b	BFP + selective RE sending	bitwidth of each compressed I and Q value
0110b	mod-compr + selective RE sending	bitwidth of each compressed I and Q value
0111b – 1111b	reserved for future methods	depends on the specific compression method

Type: unsigned integer (concatenated bit fields).

Field length: 8 bits.

8.3.3.14 reserved (reserved for future use)

Description: This parameter provides 1 byte for future definition, should be set to all zeros by the sender and ignored by the receiver. This field is only present when udCompHdr is present, and is absent when the static IQ format and compression method is configured via the M-Plane.

Value range: {0000 0000b-1111 1111b}, but shall be set to all zeros.

Type: unsigned integer.

Field length: 8 bits.

8.3.3.15 udCompParam (user data compression parameter)

Description: This parameter applies to whatever compression method is used for the PRB (configured statically via M-Plane or specified in udCompHdr of the data section containing the PRB).

Value range: {0000 0000b-1111 1111b}.

Bit allocations

udCompMeth	0 (msb)	1	2	3	4	5	6	7 (lsb)	compParam size
0000b = no compression	absent								0 octets
0001b = block fl. Point (BFP)	reserved (set to all zeros)				exponent (unsigned)				1 octet
0010b = block scaling	sblockScaler (unsigned, 1 integer bit, 7 fractional bits)								1 octet
0011b = µ-law	compBitWidth				compShift				1 octet
0100b = modulation compr.	absent								0 octets
0101 = BFP + selective RE sending	sReSMask[11:8]				exponent (unsigned)				1 octet
	sReSMask[7:0]								1 octet
0110 = mod-compr + selective RE sending	sReSMask[11:8]				reserved				1 octet
	sReSMask[7:0]								1 octet
0111b – 1111b	reserved (set to all zeros)								? octets

Type: variable.

Field length: zero for udCompMeth values 0000b and 0100b, 8 bits for udCompMeth values 0001b, 0010b and 0011b, 16 bits for udCompMeth values 0101b and 0110b; other udCompMeth values may imply other lengths but will always be an integer number of bytes.

8.3.3.16 iSample (in-phase sample)

Description: This parameter is the In-phase sample value. When the optional “little endian byte order” is chosen via M-plane, refer to Annex D.2 for detail byte order, otherwise see Annex D.1 for example sample formatting.

Value range: {all zeros – all ones}.

Type: signed integer.

Field length: 1-16 bits.

8.3.3.17 qSample (quadrature sample)

Description: This parameter is the Quadrature sample value. When the optional “little endian byte order” is chosen via M-plane, refer to Annex D.2 for detail byte order, otherwise see Annex D.1 for example sample formatting.

Value range: {all zeros – all ones}.

Type: signed integer.

Field length: 1-16 bits.

8.3.3.18 sReSMask (Selective RE Sending Mask)

Description: This parameter defines the IQ usage mask when using the compression methods block floating point + Selective RE sending or modulation compression + Selective RE sending. IQ-samples in corresponding PRB that are included in the U-plane message are indicated with bit-value 1b in the mask. IQ-samples that are not included in U-plane message are indicated with the bit-value 0b. The notation of the IQ-samples 1st, 2nd … is according to the same order as shown in Table 6-2. The most significant bit in this mask is defined as the bit for the highest frequency RE in the PRB.

Value range: {rrrr 0000 0000 0000b–rrrr 1111 1111 1111b}

Bit allocations:

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
reserved (set to all zeros)				12th	11th	10th	9th	1	Octet 1
8th	7th	6th	5th	4th	3rd	2nd	1st	1	Octet 2

Type: unsigned integer

Field length: 2 bytes

8.3.3.19 udCompLen (PRB field length)

Description: This parameter specifies the total number of octets including padding in the PRB fields up to the end of current section. The maximum supported PRB field length is 2^{16}-1, but the actual size may be further limited by the maximum payload size of the underlying transport network. This field is only present for the following compression methods:

udCompMeth 0101b = BFP + selective RE sending udCompMeth 0110b = Modulation compression + selective RE sending

Value range: {0000 0000 0000 0010b - 1111 1111 1111 1111b},

{0000 0000 0000 0000b, 0000 0000 0000 0001b} Reserved

Type: unsigned integer.

Field length: 16 bits.

NOTE: if an O-RU declares support of the field then udCompLen MUST be included in the U-Plane message (DL and UL). If the O-RU does not declare support of this field then this field MUST NOT be included in the U-Plane message. This ensures backward-compatibility with v03.00 of this Specification.

8.3.4 DL Data Precoding

- Section Extension ‘3’ is used for C-plane and associated sectionID for U-plane
- O-RU shall understand that for this Section Extension, O-RU should read 12 REs which have CRS reference signals in that PRB.
- O-RU shall understand the crsShift and crsReMask field to map appropriately the CRS REs to each antenna port

Table 8-5 : DL Data Precoding Example

<table>
<tr><th>C-plane example
DL Precoding configuration parameters and indications (Section ext “3”)</th><th>O-RU Outcome</th></tr>
<tr><td>• Section ext = 3
• txScheme=’txD’
• codeBookIndex= ‘00000000’ (invalid)
• numLayer = 4
• layerId = ‘0000’ (TxD)
• crsReMask= 001001001001 (Assuming MSB..LSB)
• crsSymNum=0000
• crsShift=0
• beamIdAP1
• beamIdAP2
• beamIdAP3</td><td>• Based on numLayers, crsShift, crsSymNum and crsReMask bit positions, CRS ports are mapped to the appropriate RE position and rest are left blank.
<table>
<tr><td>CRS (4 Layer)</td><td>If crsSymNum</td><td>crsReMask Bit Position</td></tr>
<tr><td>(crs_Ant0)</td><td>0</td><td>{0,6} + vShift</td></tr>
<tr><td>(crs_Ant1)</td><td>0</td><td>{3,9} + vShift</td></tr>
</table></td></tr>
</table>

8.3.5 Data Transfer for Special Cases

8.3.5.1 Data Message Mapping and Packetization

See clause 8.3.2.

8.3.5.2 Uplink Data Transfer

Uplink IQ data transfer is covered in clause 8.3.2. In particular, PRACH and other common channels as well as SRS and other reference signal channels use the same frequency domain IQ data packetization as with user data channels (PDSCH, PUSCH) following the transfer procedure described in clause 8.2.1. Alternatively, PRACH and SRS related IQ data may be transferred as described in clause 8.3.5.3 and clause 8.3.5.4.

8.3.5.3 PRACH Data Transfer Without C-Plane

In addition to general uplink data transfer (see 6.3.5.2), PRACH related IQ data may be transferred using the IQ data transfer procedure without C-plane (see clause 8.2.2). In this case parameters controlling signal reception and sending

U-plane messages are provided via M-plane. Refer to “Static PRACH” in M-Plane specification (see [7]) for description of M-plane parameters and rules for sending U-plane messages and interpretation of fields in U-plane messages.

8.3.5.4 SRS Data Transfer Without C-Plane

In addition to the general uplink data transfer (see 6.3.5.2), SRS related IQ data may be transferred with IQ data transfer procedure without C-plane (see clause 8.2.2). In this case parameters controlling signal reception and sending U-plane messages are provided via M-plane. Refer to “Static SRS” in M-Plane specification (see [7]) for description of M-plane parameters and rules for sending U-plane messages.

8.4 U-Plane Optimizations

8.4.1 Coupling via Frequency and Time

Coupling between C-Plane section descriptions and U-Plane data sections via frequency and time (for more details see clause 7.4.1.2.2) is mode of operation enabled by eAxC via M-plane. It allows to optimize U-Plane message size by combining data sections that are continuous in frequency and are within the same symbol. Combined data sections share the same data section header (sectionId, rb, symInc, startPrbu, numPrbu and optionally udCompHdr) resulting in reduced message size. Value of **sectionId** is the same for all sections (predefined value) and does not prevent combining. If **udCompHdr** is present, then data sections can be combined only if value of **udCompHdr** is the same. Details of resource allocation (e.g. discontinuity in frequency, differences in compression) may restrict combining but resulting U-plane message size is always less or equal the size without combining.

This U-plane optimization method is compatible with all Section Types and Section Extensions with an exception for Section Type 3. With this coupling method, when multiple Section Type 3 messages are sent each with two (or more) different “freqOffset” values and/or “frameStrcutures”, but same startPrbc and numPrbc, each C-plane messages can be uniquely identified by “freqOffset” and/or “frameStructure” fields, however associated U-Plane message with sectionId as 4095 and with no “freqOffset” or “frameStructure” cannot be uniquely identified. Hence, Section Type 3 shall not be used in conjunction with Time-Frequency Coupling. Note that frequency discontinuity of beamforming configuration in C-plane does not imply discontinuity of data sections in U-plane.

Example 1: Let’s assume before combining in symbol s there are two data sections: A and B such that A.udCompHdr = B.udCompHdr and A.startPrbu < B.startPrbu and A.rb = B.rb = 0. If B.startPrbu = A.startPrbu + A.numPrbu then A and B can be combined into data section C such that C.rb = 0, C.startPrbu = A.startPrbu, C.numPrbu = A.numPrbu + B.numPrbu.

Example 2: Let’s assume before combining in symbol s there are two data sections: A and B such that A.udCompHdr = B.udCompHdr and A.startPrbu < B.startPrbu and A.rb = B.rb = 1. If B.startPrbu = A.startPrbu + 2•A.numPrbu then A and B can be combined into data section C such that C.rb = 1, C.startPrbu = A.startPrbu, C.numPrbu = A.numPrbu + B.numPrbu.

Example 3: Let’s assume before combining in symbol s there are two data sections: A and B such that A.udCompHdr = B.udCompHdr and A.startPrbu < B.startPrbu and A.rb = B.rb = 1. If B.startPrbu = A.startPrbu + 1 and A.numPrbu = B.numPrbu then A and B can be combined into data section C such that C.rb = 0, C.startPrbu = A.startPrbu, C.numPrbu = A.numPrbu + B.numPrbu.

Note the above examples are non-exhaustive.

9 Counters and KPIs

9.1 Counters

This clause defines recommended ORAN CU-Plane specific performance counters for the fronthaul interface. Table 9-1 defines the set of mandatory and optional ORAN performance counters to be implemented within an O-RU or O-DU. These counters are defined from the perspective of the O-RU’s or O-DU’s Ethernet interfaces. These counters increment on eCPRI/IEEE1914.3 message-based events. A message is defined as a complete eCPRI, or IEEE1914.3 message including header and payload. Unless otherwise noted in the table, these counters may be implemented to count events with a granularity of per RU, per Transport Flow, or per eAxC as indicated in the fronthaul M-Plane specification [7]. Higher granularity counters provide more detailed information and can simplify managing and troubleshooting the front haul network.

It is recommended that the counters defined in Table 9-1 shall only increment if the message is contained in a valid (error free) layer 2 or 3 packet. In other words, if the packet has no layer 2 CRC or no layer 3 checksum errors it may be counted. All counters defined in this specification are assumed to be wrap-around counters. Wrap-around counters are counters that automatically go from their maximum/final value to zero and continue to operate. These are unsigned counters. These counters shall provide for any explicit means to return them to their minimum/zero, i.e., reset during normal operation. Due to their nature, wrap-around counters should be read frequently enough to avoid loss of information.

Only the transport related counters are listed herein; other O-RU counters may also be relayed across the fronthaul interface and these are described in Annex B of the M-Plane Specification [7].

Table 9-1 : O-RU and O-DU Performance Counter Definitions

Counter name	Size (bits)	Mandatory or Optional	Counter Definition
RX_TOTAL	64b	Mandatory	The total number of control/user plane eCPRI or 1914.3 messages received. This counter is the sum of all valid and errored messages received. (previously ORAN-WG4.CUS.0-v02.00 named this Total_msgs_rcvd)
RX_ON_TIME	64b	Mandatory	The number of inbound user plane (ecpri type 0) messages that arrived within the specified time window. Some "on time" messages may have sequence number errors or corruption errors but as long as they arrived within specified window time, this counter should include them. If the received message has been transport-fragmented, the full message shall be reassembled before checking its arrival window. • Implementation of a single global counter is required. • Implementation of per transport counters are optional. • Implementation of per eAxC counters are optional.
RX_EARLY	64b	Mandatory	The number of inbound user plane messages which were detected to have arrived before the start of their designated receive window time. This counter increments whether the message is subsequently processed or dropped. • Implementation of a single global counter is required. • Implementation of per transport counters are optional. • Implementation of per eAxC counters are optional.
RX_LATE	64b	Mandatory	The number of inbound user plane messages which were detected to have arrived after the end of their designated receive window time. This counter increments whether the message is subsequently processed or dropped. • Implementation of a single global counter is required. • Implementation of per transport counters are optional. • Implementation of per eAxC counters are optional.
RX_SEQID_ERR	64b	Optional	The number of inbound on time user plane messages in which a sequence identifier number error is detected. When this counter is implemented with an eAxC granularity, separate uplink and downlink counters shall be implemented. This error occurs when the ecpriSeqId field does not increment as specified in Clause 5.1.3. Both the Sequence ID, and Subsequence ID fields shall be checked if transport fragmentation is supported otherwise only the Sequence ID field may be checked. In addition to identifying a sending equipment sequencing error, this counter can increment when packets are dropped prior to reception by the RU, or when packets reordered by the network exceed the receiving device's capabilities.
RX_ON_TIME_C	64b	Mandatory	The number of valid inbound control plane (ecpri type 2) messages that arrived within the specified time window. Some "on time" messages may have sequence number errors or corruption errors but as long as they arrived within specified window time, this counter should count them.

Counter name	Size (bits)	Mandatory or Optional	Counter Definition
			• Implementation of a single global counter is required. • Implementation of per eAxC counters are optional. • Implementation of per transport counters are optional. • Implementation of counters for each combination of eAxC and dataDirection fields are optional.
RX_EARLY_C	64b	Mandatory	The number of inbound control plane messages which were detected to have arrived before the start of their designated receive window time. This counter increments whether the message is subsequently processed or dropped. • Implementation of a single global counter is required. • Implementation of per eAxC counters are optional. • Implementation of per transport counters are optional. • Implementation of counters for each combination of eAxC and dataDirection fields are optional.
RX_LATE_C	64b	Mandatory	The number of inbound control plane messages which were detected to have arrived after the end of their designated receive window time. This counter increments whether the message is subsequently processed or dropped. • Implementation of a single global counter is required. • Implementation of per eAxC counters are optional. • Implementation of per transport counters are optional. • Implementation of counters for each combination of eAxC and dataDirection fields are optional.
RX_SEQID_ERR_C	64b	Optional	The number of inbound on time control messages in which a sequence identifier number error is detected. This counter increments under the same conditions as the rx_seqid_num_err except for control plane messages. • Implementation of a global counter is optional. • Implementation of eAxC counters is optional. • Implementation of per transport counters are optional. • Implementation of counters for each combination of eAxC and dataDirection fields are optional.
RX_CORRUPT	64b	Optional	The number of inbound on time messages with a correct ecpriSeqId (no sequence number error) which are dropped by the terminating entity due to the message containing one or more eCPRI/1914.3 or ORAN protocol errors. This counter's granularity is limited to per O-RU or per transport flow implementations. Protocol errors are defined as when eCPRI/1914.3/ORAN defined fields contain invalid values or indicate unsupported capabilities. Some examples of this are: 1. PcId or section Id number which has not been configured. 2. Unexpected use of C bit, 3. Unconfigured or Unsupported udCompHdr setting. 4. Unsupported Section Extension. 5. Wrong ecpriVersion and/or payloadVersion information 6. ecpriMessage field does not contain 0, 2, or 5.
RX_ERR_DROP		Optional	The total number of inbound messages which are discarded by the receiving O-RAN entity for any reason.
RX_PKT_DUPL	64b	Optional	Duplicated packet. This counter is **deprecated**.
TX_TOTAL	64b	Mandatory	The number of valid outbound control/user plane messages.
TX_TOTAL_C	64b	Mandatory	The number of valid outbound control plane messages. This counter is required on an O-RU only if O-RU supports LAA/LBT capabilities.
NOTE: A counter is listed as mandatory if it is considered important enough to be necessary for the O-RU or O-DU to be compliant to this specification. A counter is defined as optional if it is of lessor importance and therefore is not necessary to implement for compliance to this standard.			

10 Specification Mandatory and Optional Capabilities

10.1 General

This clause provides details regarding which capabilities within the specification are mandatory and which are optional. The list will in general be different for the O-DU versus the O-RU because in many cases, the O-DU will need to implement multiple options as mandatory to ensure interoperability with O-RUs that have optional capabilities. For example, the ability to support many compression methods may be mandatory in the O-DU while in O-RUs there may be only a single mandatory compression method to allow simplicity in O-RU design (while vendors may enhance their O-RU product offering by implementing some of the optional compression methods).

Table 10-1 : O-RAN-Supported LTE and NR Channels

Physical Channel Support	
LTE DL Channels	PDSCH, PBCH, PCFICH, PDCCH, ePDCCH, MPDCCH, PHICH, CRS, MBSFN RS, UE-RS, DMRS for ePDCCH/MPDCCH, PRS, CSI-RS, PSS, SSS, Discovery RS
LTE UL Channels	PUSCH, PUCCH, DMRS-PUSCH, DMRS-PUCCH, SRS, PRACH (incl. eMTC)
Narrow band IoT DL Channels	NB-DMRS, NB-PDSCH, NB-PBCH, NB-PDCCH, NB-RS, NB-PRS, NB-PSS, NB-SSS
Narrow band IoT UL Channels	NB-PUSCH, NB-PRACH
NR DL Channels	PDSCH, PDCCH, DMRS-PDSCH, PTRS-PDSCH, DMRS-PDCCH, DRMS-PBCH, CSI-RS, PSS, SSS, SS Block/PBCH
NR UL Channels	PUSCH, PUCCH, PRACH, DMRS-PUSCH, PTRS-PUSCH, DMRS-PUCCH, SRS
System Capability Support	
Technologies	LTE TDD, FDD (normal and extended CP) NR TDD, FDD
Channel Bandwidth	LTE: 1.4, 3, 5, 10, 15, 20 MHz
	NR: up to 400MHz
Subcarrier Spacing	LTE: 15kHz, 7.5kHz, 1.25kHz LTE PRACH: 1.25kHz, 7.5kHz NB-IoT PRACH: 3.75kHz
	NR: 15, 30, 60, 120, 240 kHz
	NR Multi Numerology
	NR PRACH: 1.25, 5, 15, 30, 60, 120 kHz
LTE Specific Features	DL Transmission Modes : TM1 - TM10
	UL Transmission Modes : TM1, TM2
	Carrier Aggregation
	eMBMS
	TTI-Bundling
	Semi-Persistent Scheduling (SPS)
	MIMO (SU/MU-MIMO)
	UE TAS (Tx Antenna Selection)
	FeICIC (ABS)
	CoMP (DL/UL), JT
	Short TTI
	eMTC
	NB-IOT (in band/guard band/standalone)
	License Assisted Access (LAA)

	Sidelink (Proximity Services)
	Dynamic TDD (eIMTA)
	Mission Critical PS-LTE Features (MCPTT, ..)
	Positioning (PRS, OTDOA etc)
	V2X
	Distributed Antenna System Support
	CBRS Support
NR Specific Features	EN-DC
	SSBlock
	BW Part
	Supplementary UL
	Mini-slot
	LTE-NR Co-existence
Beamforming	Analog Beamforming
	Digital Beamforming
	Hybrid Beamforming
	O-RU Support for 64 TRX
Transport	L2 : Ethernet
	L3 : IPv4, IPv6
	QoS over Fronthaul

Table 10-2 describes the capabilities required of O-DU and O-RU units. There are three choices:

Mandatory: the unit shall support the described capability to be O-RAN compliant.

Conditional Mandatory: the unit shall support the described capability to be O-RAN compliant, but the Notes column described the conditions under which the capability is mandatory.

Optional: the unit may not support the capability and still be O-RAN compliant, but if the unit does support the described capability it shall support it in the way described within the O-RAN specifications**.**

Table 10-2 : O-RAN Mandatory and Optional Features

Category	Feature of O-DU or O-RU	O-DU Support	O-RU Support	Note
O-RU Category Support	**Support for CAT-A O-RU (up to 8 spatial streams)**	**Mandatory**	NA	The O-DU may only support fewer than 8 spatial streams; that number of spatial streams shall however be supported for CAT-A O-RUs.
	Support for CAT-A O-RU (> 8 spatial streams)	Optional	NA	
	Support for CAT-B O-RU (precoding in O-RU)	**Mandatory**	NA	
Beam-forming	**Beam Index based**	**Mandatory**	**Conditional Mandatory**	Condition applies to UE-specific BF for any O-RU capable of BF; a non-BF radio shall be supplied a zero beamId if a C-Plane message containing a beamId will be sent at all (FFS).
	Real-time BF Weights	**Conditional Mandatory**	**Conditional Mandatory**	Condition for O-DU: Mandatory only for O-DUs designed to support any kind of BF that involves the

Category	Feature of O-DU or O-RU	O-DU Support	O-RU Support	Note
				operation of updating BF weights in real-time; Condition for O-RU: for any O-RU internally using BF weights, the ability to update the weights in real-time via C-Plane messages (using "Real-time BF Weights") shall be mandatory.
	Real-Time Beamforming Attributes	Optional	Optional	This is considered to not internally use BF weights.
	Real-Time UE Channel Info	Optional	Optional	This is considered to internally use BF weights.
	Predefined beam tilt for Beam Index based Beamforming	Optional	Optional	
	Antenna Calibration support	Optional	Optional	
	Null-layer Info. for ueId-based beamforming using Section Extension 14	Optional	Optional	
	User port group indication for beamforming based on UE channel (Section Extension = 17)	Optional	Optional	
Bandwidth Saving	**IQ Data Formats**			
	- **Fixed point** (no compression)	**Mandatory**	**Mandatory**	**16-bit mandatory** (others optional).
	- Block floating point compression	**Conditional Mandatory**	**Conditional Mandatory**	**9, 12 & 14-bit mantissa mandatory if this compression method is supported** (others optional)**.** Condition: if an O-DU or O-RU supports any IQ compression it shall support this one.
	- Block scaling compression	Optional	Optional	**9 & 14-bit scaler mandatory if this compression method is supported** (others optional).
	- µ-law compression	Optional	Optional	**9 & 14-bit width mandatory if this compression method is supported** (others optional).
	- Modulation compression	Optional	Optional	**4-bit width mandatory if this compression method is supported** (others optional).
	- Block floating point compression + selective RE sending	Optional	Optional	**9, 12 & 14-bit mantissa mandatory if this compression method is supported** (others optional).
	- Modulation compression + selective RE sending	Optional	Optional	**4-bit width mandatory if this compression method is supported** (others optional).
	Presence of udCompLen	**Conditional Mandatory**	Optional	If the O-RU declares support of udCompLen then it shall be used, otherwise it shall be omitted (in DL and UL U-Plane messages); this is only relevant for block floating point with selective RE sending or modulation compression, with selective RE sending.
	Real-time variable bit-width	Optional	Optional	IQ data format is determined by value of udCompHdr. This implies

Category	Feature of O-DU or O-RU	O-DU Support	O-RU Support	Note
				presence of udCompHdr in U-plane messages.
	Real-time variable bit-width per Channel (per data section)	Optional	Optional	IQ data format is determined by value of udCompHdr. This implies presence of udCompHdr in U-plane messages. Values of udCompHdr can be different in different sections in the U-plane messages.
	Static configuration of U-Plane IQ format and compression header	**Conditional Mandatory**	**Conditional Mandatory**	IQ data format is determined by M-plane configuration. This implies absence of udCompHdr in U-plane messages. Mandatory for supported IQ formats listed in 7 rows under "IQ Data Formats" in this table.
	Beamspace compression	Optional	Optional	This compression algorithm is specific to beamforming weights.
	Use of "symInc" flag to allow multiple symbols in a C-Plane section	Optional	Optional	
	Coupling via sectionId Value	**Mandatory**	**Mandatory**	
	Coupling via Frequency and Time	**Conditional Mandatory**	**Conditional Mandatory**	If O-RU or O-DU supports "Coupling via Frequency and Time with Priorities" it shall also support this one.
	Coupling via Frequency and Time with Priorities	Optional	Optional	
	Coupling via Frequency and Time with Priorities (Optimized)	Optional	Optional	Refer clause 7.4.1.2.4 and clause 7.5.7.1.
	PRACH data transfer without C-Plane	Optional	Optional	
	SRS data transfer without C-Plane	Optional	Optional	
Energy Savings	Transmission blanking	Optional	Optional	
O-DU – O-RU Timing	**Defined Transport Method**	**Mandatory**	**Mandatory**	
	Measured Transport Method (eCPRI Msg 5)	Optional	Optional	If O-RU supports Measured Transport Method it shall support both 1-Step and 2-step version of T12 measurement and at least one of 1-Step or 2-Step version of T34 measurement. If the O-DU supports T34 measurement it shall support both 1-Step and 2-Step version. Refer clause 4.3.3.3 for more detailed information.
	External Antenna Delay handling using Minimal O-DU Impact Method	Optional	Optional	Using Tda and Tau parameters as defined in clause 4.6.1.
Synchroni-zation	**G.8275.1**	**NA**	**Conditional Mandatory**	- All O-RUs shall support the multicast PTP profile of G.8275.1 except for O-RUs targeting only config LLS-C4. - O-DU and O-RU end applications may optionally use PLFS input within G.8275.1.

Category	Feature of O-DU or O-RU	O-DU Support	O-RU Support	Note
				- in LLS-C1 config, the emission of PLFS by O-DU is only mandatory if O-RU requires it. - in LLS-C2 config, the emission of PLFS by O-DU is mandatory. - in LLS-C3/C4, the O-DU does not transmit synchronization signals to the O-RU, so the emission of PLFS is optional. - in LLS-C2 and LLS-C3 topologies supporting Shared cell, the emission of PLFS is mandatory for network elements in the Fronthaul synchronization chain (FHM or cascaded O-RUs that are on the path to other network elements in the synchronization chain).
	G.8275.2	Optional	Optional	- O-DU (in LLS-C1/C2/C3/C4) or O-RU (in LLS-C1/C2/C3) may optionally be synchronized from a G.8275.2 PTP source. - in LLS-C1/C2, O-DU may optionally act as G.8275.2 PTP master on the fronthaul for the O-RU.
	Local PRTC	Optional	Optional	O-DU (in LLS-C1/C2/C3/C4) or O-RU (in LLS-C4) may optionally be synchronized from a local time source, for example GNSS-based.
Transport Features	**L2 : Ethernet**	**Mandatory**	**Mandatory**	
	L3 : IPv4, IPv6 (CUS Plane)	Optional	Optional	
	QoS over Fronthaul	**Mandatory**	**Mandatory**	
	Prioritization of different U-Plane traffic types	Optional	Optional	
	Support of Jumbo Ethernet frames	Optional	Optional	
	eCPRI	**Mandatory**	**Mandatory**	
	support of eCPRI concatenation	Optional	Optional	
	IEEE 1914.3	Optional	Optional	
	Application layer fragmentation	**Mandatory**	**Mandatory**	C-Plane and U-Plane.
	Radio Transport layer fragmentation	Optional	Optional	U-Plane (see clause 5.1.3.1.7).
Section Types / extensions	Section Type 0	Optional	**Mandatory**	O-RU may ignore message if blanking or other Section Type 0 utility is not supported.
	Section Type 1	**Mandatory**	**Mandatory**	
	Section Type 3	**Mandatory**	**Mandatory**	
	Section Type 5	Optional	Optional	Specific to Channel-Info beamforming.
	Section Type 6	Optional	Optional	Specific to Channel-Info beamforming.

Category	Feature of O-DU or O-RU	O-DU Support	O-RU Support	Note
	Section Type 7	Optional	Optional	Specific to LAA which is an optional capability.
	Beamforming weight transfer using Section Extension = 1	**Conditional Mandatory**	**Conditional Mandatory**	**Condition for O-DU**: Mandatory only for O-DUs designed to support any kind of BF that involves the operation of updating BF weights in real-time; **Condition for O-RU**: for any O-RU internally using BF weights, the ability to update the weights in real-time via C-Plane messages (using “Real-time BF Weights”) shall be mandatory.
	Beamforming attribute transfer using Section Extension = 2	Optional	Optional	Attribute-based beamforming is optional.
	DL precoding configuration using Section Extension = 3	Optional	Optional	While CAT B is mandatory, it is possible to precode using beamforming so use of this extension is optional.
	Modulation compr. Parameters using Section Extension =4	Optional	Optional	Modulation compression is optional.
	Modulation compr. Parameters using Section Extension =5	Optional	Optional	Modulation compression is optional.
	Non-contiguous PRB allocation using Section Extension = 6	Optional	Optional	Use of non-contiguous PRBs is optional.
	eAxC masking using Section Extension =7	Optional	Optional	Use of eAxC masking is optional.

	Provide MMSE parameters using Section Extension =8	**Conditional Mandatory**	**Conditional Mandatory**	Specific to Channel-Info beamforming;
				O-DU condition: if O-RU and O-DU both support Channel-Info BF, then the O-DU shall use Section Extension =8 to convey MMSE parameters;
				O-RU condition: if the O-RU supports Channel-Info BF, then the O-RU shall accept Section Extension =8 MMSE parameters from the O-DU.

	Add LTE/NR indicator using Section Extension =9	Optional	Optional	DSS using overlapping carriers is possible; DSS using this Section Extension is optional.
	Group configuring of multiple ports using Section Extension =10	Optional	Optional	Use of multiple port (multiple eAxC) grouping is optional.
	Flexible Beamforming Weights using Section Extension =11	Optional	Optional	Use of flexible beamforming weights Section Extension is optional.
	Non-contiguous PRB allocation with frequency ranges using Section Extension =12	Optional	Optional	
	PRB allocation with frequency hopping using Section Extension =13	Optional	Optional	
	Nulling-layer Info. for ueId-based beamforming using Section Extension = 14	Optional	Optional	
	Mixed-numerology Info. for ueId-based beamforming using Section Extension =15	Optional	Optional	

Category	Feature of O-DU or O-RU	O-DU Support	O-RU Support	Note
	Antenna mapping in UE channel information based UL beamforming using Section Extension = 16	Optional	Optional	Use of antenna mapping in UE channel information based UL beamforming is optional.
	Description for indication of user port group using Section Extension = 17	Optional	Optional	
	Uplink traffic management using Section Extension = 18	Conditional Mandatory	Conditional Mandatory	Mandatory if uplink traffic management using C-Plane is supported. Not permitted if uplink traffic management using C-Plane is not supported.
	Compact multiple port beamforming information using Extension = 19	Optional	Optional	See clause 7.4.7.19 and 7.5.10.
	Dedicated puncturing Section using Extension = 20	Optional	Optional	See clause 7.4.7.20 and 7.5.11.
Other features	LAA LBT O-DU Congestion Window mgmt	**Conditional Mandatory**	**Conditional Mandatory**	Mandatory only for O-DUs and O-RUS supporting LAA.
	LAA LBT O-RU Congestion Window mgmt	Optional	Optional	
	UL gain correction per eAxC	Optional	Optional	See clause 8.1.3.2.2.
	DL reference level adjustment	Optional	Optional	See Reference_Level in clause 8.1.3.3.
	FS adjustment	Optional	Optional	See FS_Offset in clause 8.1.3.
	Ordered transmission	Optional	Optional	See clause 4.5.3.
	Uplink traffic management using M-Plane	Optional	Optional	See clause 4.5.4.
	Uplink traffic management using C-Plane	Optional	Optional	See clause 4.5.4.
	Uniformly distributed transmission	Optional	Optional	See clause 4.5.2, Requires support of uplink traffic management (using M-plane or C-plane).
	Independent U-plane transmission window control	Optional	Optional	See clause 4.5.4. Requires support of uplink traffic management (using M-plane or C-plane).
	C-Plane Message processing O-RU limits	Optional	Optional	Refer clause 7.6.2 CUS-Plane and clause 15.8 M-Plane.

NOTE: when a capability that is "per endpoint" is cited as MANDATORY in the O-RU, it means at least one endpoint shall support it (a minimum number meaningful for the functionality of the O-RU), not that ALL endpoints shall support it.

Note also that for some Mandatory capabilities, only a subset of the full set of parameter values is really mandatory, this is marked in the “Note” column. For some Optional capabilities, there are parameter values that shall be supported (if the optional capability is present) and other parameter values may be supported but are not mandatory for that optional feature. These are marked in the Note column.

11 S-Plane Protocol

11.1 General

11.1.1 Overview

Time and frequency synchronization can be distributed to the O-DU and O-RU in different manner. However, synchronization accuracy is mostly impacted by implementation (e.g., timestamping near the interfaces, number of hops) than by the technology itself. The following synchronization options are available over an Ethernet network:

- Frequency synchronization where clocks are aligned in frequency
- Phase synchronization where clocks are aligned in phase
- Time synchronization where clocks are aligned to a common base time

Together the above parameters define a profile for the network, requiring a set of features and option selections for bridges and end stations operation. Further, the profile also states the conformance requirements for supporting equipment and user applications.

This edition of the document considers frequency, phase and time synchronization of all the network elements (O-DUs, intermediate switches and O-RUs) for TDD and FDD features requiring specific TAE. Frequency-only configurations (like LTE FDD or 5G FDD) are For Further Study.

This edition of the document considers macro BTS O-DUs and not small cells that have O-DU and O-RU in the same box and therefore no need for fronthaul link.

11.2 Synchronization Baseline

11.2.1 List of Reference Documents

See clause 2.

11.2.2 Clock Model and Synchronization Topology

Different O-RAN synchronization topologies are necessary to address different deployment market need. The following 4 topology configurations are considered by O-RAN as compliant topologies for supporting the O-RU synchronization needs. A configuration label is used for easier reference through this specification:

- Configuration LLS-C1: with this topology, the O-DU is part of the synchronization chain towards the O-RU. Network timing is distributed from O-DU to O-RU via direct connection between O-DU site and O-RU site

- Configuration LLS-C2: with this topology, the O-DU is part of the synchronization chain towards the O-RU. Network timing is distributed from O-DU to O-RU between O-DU sites and O-RU sites. One or more Ethernet switches are allowed in the fronthaul network. Interconnection among switches and fabric topology (for example mesh, ring, tree, spur etc.) are deployment decisions which are out of the scope of this O-RAN specification.

- Configuration LLS-C3: with this topology, the O-DU is not part of the synchronization chain towards the O-RU. Network timing is distributed from PRTC/T-GM to O-RU typically between central sites (or aggregation sites) and O-RU sites. One or more Ethernet switches are allowed in the fronthaul network. Interconnection among switches and fabric topology (for example mesh, ring, tree, spur etc.) are deployment decisions which are out of the scope of this O-RAN specification

- Configuration LLS-C4: This topology concerns the case where the synchronization reference is provided to the O-RU with no involvement of the transport network (typically with a local GNSS receiver).

Note applying to all LLS-C1 to LLS-C4:

- With exception of the O-DU and O-RU, only clock types and classes (PRTC/T-GM, T-BC, T-TC) under G.8271.1 are covered by this document.

- How O-DU is synchronized is not in the scope of this classification of the synchronization topologies. O-DU may be synchronized from either a local or remote PRTC.

11.2.2.1 Topology configuration LLS-C1 and LLS-C2 Synchronization

Configuration LLS-C1 is based on point-to-point connection between O-DU and O-RU using network timing option. As shown in **Figure 11-1** below, it is basically the simplest topology for network timing option, where O-DU directly synchronizes O-RU.

Configuration LLS-C2 is similar to LLS-C1 with O-DU acting as master to distribute network timing toward O-RU. One or more Ethernet switches are allowed between the central site (hosting O-DUs) and the remote sites (hosting O-RUs). The allowed number of switches in the synchronization path is limited by frequency and time error contributions by all clocks in the chain.

With Full Timing Support, the allowed network noise limit in the budget can be met by a certain number (refer to Annex H for more details) of Class B or class C T-BC switches as shown in **Figure 11-1** below. Additional T-BC switches may be allowed if total noise limit can be met. The synchronization master is located at the O-DU. Further, all Ethernet switches in the fronthaul function as Telecom Boundary Clocks as specified by ITU-T G.8273.2. T-TC switch is also allowed as T-BC replacement with the same expectation based on G.8271.1.

With Partial Timing Support, non-T-BC switches may also be deployed. Further investigation is required to specify appropriate frequency and timing budgets and network configuration to ensure 4G and NR TAE requirements as described in **Table 9-3** and 3GPP frequency accuracy requirements can be met.

Interconnection among switches and fabric topology (for example mesh, ring, tree, spur etc.) are deployment decisions which are out of the scope of this O-RAN spec.

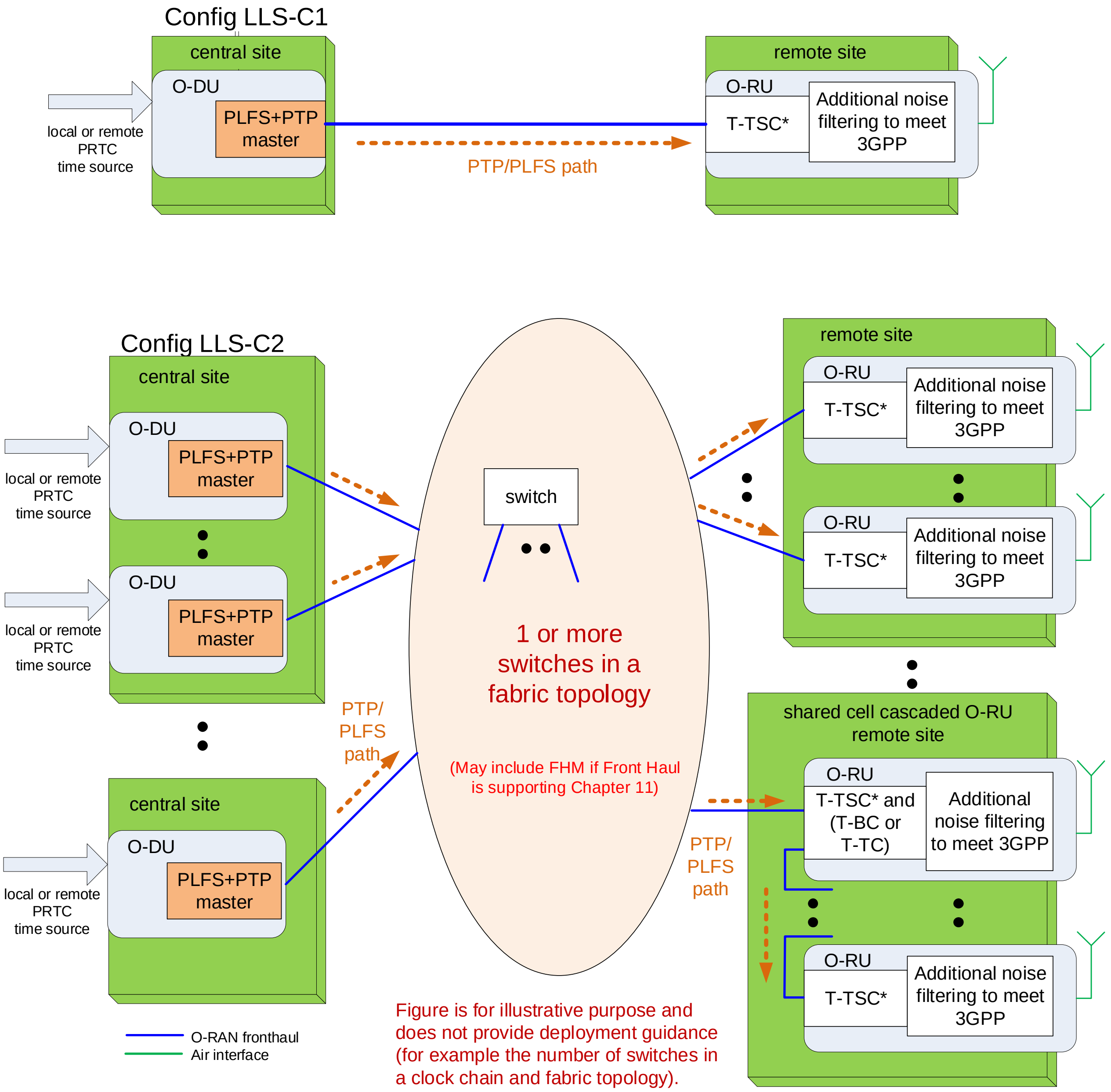

Config LLS-C1
central site
O-DU
PLFS+PTP master
local or remote PRTC time source
PTP/PLFS path
remote site
O-RU
T-TSC*
Additional noise filtering to meet 3GPP
Config LLS-C2
central site
O-DU
PLFS+PTP master
local or remote PRTC time source
O-DU
PLFS+PTP master
local or remote PRTC time source
central site
O-DU
PLFS+PTP master
local or remote PRTC time source
PTP/ PLFS path
switch
1 or more switches in a fabric topology
(May include FHM if Front Haul is supporting Chapter 11)
remote site
O-RU
T-TSC*
Additional noise filtering to meet 3GPP
O-RU
T-TSC*
Additional noise filtering to meet 3GPP
shared cell cascaded O-RU remote site
O-RU
T-TSC* and (T-BC or T-TC)
Additional noise filtering to meet 3GPP
PTP/ PLFS path
O-RU
T-TSC*
Additional noise filtering to meet 3GPP
O-RAN fronthaul
Air interface
Figure is for illustrative purpose and does not provide deployment guidance (for example the number of switches in a clock chain and fabric topology).

Figure 11-1 : Configurations LLS-C1 and LLS-C2 synchronization (see notes below for additional information on O-DU, O-RU, Switch functionality).

Notes for O-DU:

O-DU acts as PLFS + PTP synchronization master towards the fronthaul interface, but there are different possible sub-configurations based on the O-DU sync source. Two main cases are possible: O-DU is itself the master at the top of the synchronization chain (e.g., PRTC directly connected to the O-DU) or it gets synchronization from the network. In particular:

- if O-DU synchronization source is from a local PRTC (typically a GNSS receiver), it may act as a T-GM, or as a specific PLL with higher jitter and wander filtering capability.
- if O-DU synchronization source is from a remote PRTC through a network (typically PTP, with or without PLFS, which grandmaster can be located anywhere in the network), O-DU acts as synchronization subordinate toward the upstream network. Two sub- configurations are possible:
    - If the PTP profile used in the upstream network is different (typically ITU-T G.8275.2) from the fronthaul one (which is typically ITU-T G.8275.1), then the O-DU acts as an InterWorkingFunction clock to bridge between the profiles, as defined in ITU-T G.8271.2 (but no clock specification exists for such IWF). This is For Further Study.

- If the PTP profile used in the upstream network is same as the fronthaul one (typically ITU-T G.8275.1), then the O-DU may act as a combined PTP Subordinate and Master with higher jitter and wander filtering capability. NOTE: O-DU acting as a ITU-T G.8273.2 T-BC clock does not provide enough wander cleaning to guarantee the 15 ppb limit and is therefore outside the scope of LLS-C1/LLS-C2.

When multiple O-DU are directly connected to the central site aggregation switch supporting T-BC in configuration LLS-C2, only one active Master will be in the clock chain. One O-DU will serve as active GrandMaster to all O-RUs, including the ones controlled by other O-DUs over the M-plane. If Master redundancy is needed, another O-DU shall serve as backup Master.

Notes for Switches:

- ITU-T “Full Timing Support” Clock: ITU-T G.8273.2 T-BC (or ITU-T G.8273.3 T-TC).
- In the case of FHM or cascaded O-RU used in shared cell topologies described in clause 13, this equipment also behaves as an ITU-T “Full Timing Support” Clock from its subordinate to master port from S-plane point of view.
- Partial timing support using non-T-BC switches may also be allowed. G.8275.2 Telecom Profile is required for partial timing support. Performance aspects and budgets associated with this mode require further investigation.

Notes for O-RU:

- As an end application, the O-RU includes an application-specific PTP (T-TSC*) subordinate clock that may be compliant to G.8273.2 T-TSC. The combination of the T-TSC* and the local RF end application clock needs to fulfill additional requirements to ensure 3GPP air interface compliance on the RF interface.
  T-TSC is per IEEE 802.1CM interface condition Case 1.
- In the case of Shared Cell using cascade mode with direct chaining of the O-RUs, the O-RU also implements either ITU-T G.8273.2 T-BC or ITU-T G.8273.3 T-TC on the path between the S-Plane subordinate and S-Plane master ports (as per above Notes for Switches).
  If the O-RU is not on the synchronization path to other O-RUs, the O-RU is not required to support T-BC or T-TC functions. These may be supported by a collocated switch.
- An end application connected to an external T-TSC per IEEE 802.1CM interface condition Case 2. This case may be excluded by O-RAN since the O-RU generally does not provide a separate PPS/ToD interface for external T-TSC connection, and there may be some performance concerns about the 1pps distribution interface compared to PTP over Ethernet.

11.2.2.2 Topology configuration LLS-C3 Synchronization

Configuration LLS-C3 is similar to LLS-C2 except frequency and time distribution is made by the fronthaul network itself (not by the O-DU). That means that one or more PRTC/T-GM are implemented in the fronthaul network to distribute network timing toward O-DU and O-RU. One or more Ethernet switches are allowed between the central site (hosting O-DUs) and the remote sites (hosting O-RUs). The permitted number switches in the synchronization path is limited by frequency and time error contributions by all clocks in the chain.

With Full Timing Support, all Ethernet switches in the fronthaul function as T-BC as specified by ITU-T G.8273.2. T-TC switch is also allowed as T-BC replacement with the same expectation based on G.8271.1.

Partial Timing Support using non T-BC switches may also be allowed. G.8275.2 Telecom Profile is required for Partial Timing Support. Performance aspects and budgets associated with this mode requires further investigation.

Interconnection among switches and fabric topology (for example mesh, ring, tree, spur etc.) are deployment decisions which are out of the scope of this O-RAN spec.

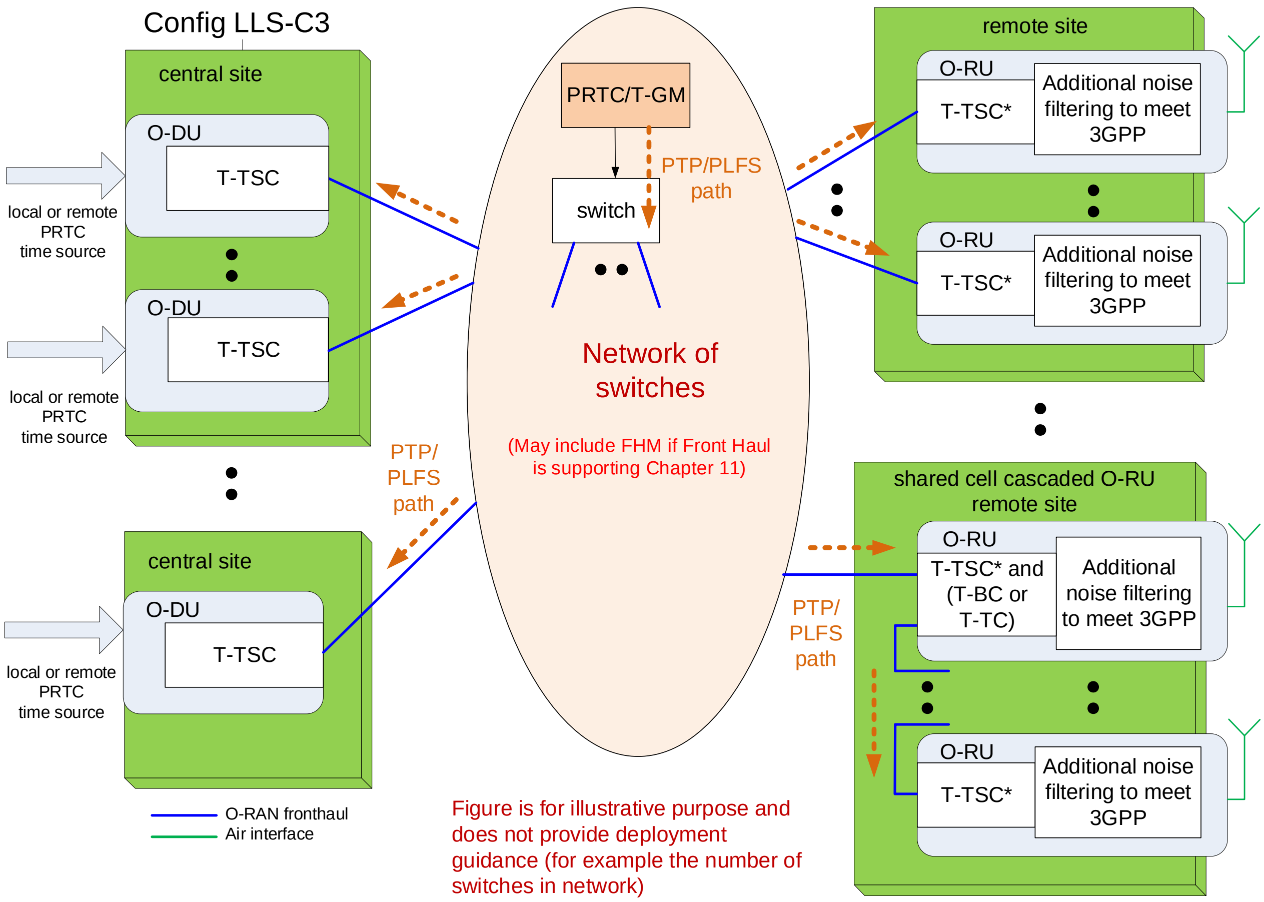

Config LLS-C3
central site
O-DU
T-TSC
local or remote PRTC time source
O-DU
T-TSC
local or remote PRTC time source
central site
O-DU
T-TSC
local or remote PRTC time source
O-RAN fronthaul
Air interface
PRTC/T-GM
switch
PTP/PLFS path
Network of switches
(May include FHM if Front Haul is supporting Chapter 11)
PTP/ PLFS path
remote site
O-RU
T-TSC*
Additional noise filtering to meet 3GPP
O-RU
T-TSC*
Additional noise filtering to meet 3GPP
shared cell cascaded O-RU remote site
O-RU
T-TSC* and (T-BC or T-TC)
Additional noise filtering to meet 3GPP
PTP/ PLFS path
O-RU
T-TSC*
Additional noise filtering to meet 3GPP
Figure is for illustrative purpose and does not provide deployment guidance (for example the number of switches in network)

Figure 11-2 : Configuration LLS-C3 synchronization (see notes below for additional information on O-DU, O-RU, Switch functionality).

Notes for O-DU:

Unlike LLS-C1 & LLS-C2, O-DU does not act as synchronization master towards the fronthaul interface. It can select its own synchronization from local or remote PRTC like in LLS-C1/LLS-C2, but can also select the same synchronization master from the fronthaul as the O-RU and act as an ITU-T G.8273.2 Telecom Time Slave Clock (T-TSC).

One possible LLS-C3 implementation consists in having one of the ITU-T G.8273.2-compatible T-BC of the chain being the O-DU.

Whether O-DU is or not involved in the synchronization of the O-RU, it shall meet the time error limit for latency management as specified in clause 4.3.

Notes for Switches:

Same as for LLS-C2.

Notes for O-RU:

Same as LLS-C1 & LLS-C2.

Notes for PRTC/T-GM (as operator choice in deployment):

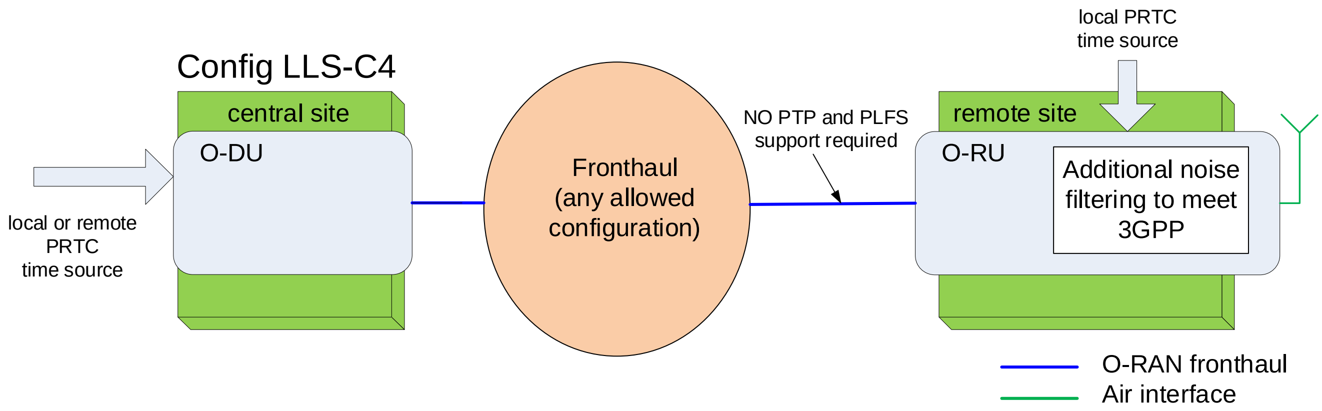
-
- The PRTC (typically a GNSS receiver) can be embedded either in an external T-GM connected to any T-BC, or an embedded function inside any T-BC in the network,
- Multiple PRTC/T-GMs can offer redundancy. ITU-T G.8275.1 BMCA is used to optimize the time distribution through the clock chain.

11.2.2.3 Topology configuration LLS-C4 Synchronization

O-RAN maintains network timing distribution as the preferred approach within the fronthaul network, however, there could be some deployment use cases that prevent the fronthaul network (or only a section of the network) from being upgraded to G.8271.1 compliance and meeting the target performance at the O-RU. To cover these use cases, the O-RAN synchronization strategy shall allow local PRTC (typically GNSS) timing option at the O-RU.

It should be noted that such timing support at O-RU requires extra timing interface or embedded function. Examples of implementation include:

- a local PRTC that provides time synchronization to the O-RU via a 1PPS/ToD interface
- a local PRTC/T-GM that provides synchronization to the O-RU via a local PTP interface
- a local PRTC embedded in the O-RU


Config LLS-C4
central site
O-DU
local or remote PRTC time source
Fronthaul (any allowed configuration)
NO PTP and PLFS support required
local PRTC time source
remote site
O-RU
Additional noise filtering to meet 3GPP
O-RAN fronthaul
Air interface

Note : Additional O-RU noise filtering is needed to filter sync source dynamic noise to meet 3GPP frequency accuracy requirement and satisfy target |TAE|.

Figure 11-3 : Configuration LLS-C4 synchronization

Notes for O-DU:

Unlike within LLS-C1 & LLS-C2 configurations, O-DU does not act as synchronization master towards the fronthaul interface. It can select its own synchronization from local or remote PRTC like in LLS-C1/LLS-C2, but can also select the same synchronization master from the fronthaul.

Although O-DU is not involved in the synchronization of the O-RU, it shall meet the time error limit for latency management as specified in clause 4.3.

11.2.3 Clock Synchronization

The following requirements are provided for clock synchronization:

- For Full Timing Support networks (either upstream O-DU or fronthaul between O-DU and O-RU), PLFS (typically SyncE) shall be used within the fronthaul network distribution per ITU-T G.8271.1, G.8275.1 and G.8273.2. In an LLS-C1 configuration, the O-DU shall drive the PLFS regardless of its own selected time source. A derogation applies under the operator’s responsibility in the specific case where O-RU does not make use of PLFS, in which case PLFS emission is optional for the O-DU. In an LLS-C2 configuration, the O-DU shall drive the PLFS regardless of its own selected time source. In an LLS-C3 configuration, the network shall deliver the PLFS to all switches and O-RUs. However, an O-DU or O-RU end application is not required to use a PLFS to achieve clock (frequency) synchronization. They can use PTP alone to achieve both frequency and phase/time synchronization. The ITU recommendations listed above refer to any PLFS and not just SyncE (for example SDH). However only SyncE has been fully studied for the T-BC so far (ITU-T G.8273.2), and SyncE is the most common implementation as it is in-band over Ethernet.
- For Partial Timing Support networks (either upstream of the O-DU or via the fronthaul between the O-DU and O-RU), PLFS may optionally be used, as per ITU-T G.8271.2 and G.8275.2
- When used for clock synchronization, SyncE shall comply with the following ITU-T specifications:

  - Definitions: ITU-T G.8260
  - Architecture: ITU-T G.8261
  - SSM transport channel and format: ITU-T G.8264
  - Clock specifications: ITU-T G.8262 (EEC) for the original, non-enhanced SyncE
  - Clock specifications: ITU-T G.8262.1 (eEEC) for the enhanced SyncE
  - Functional model and SSM processing: ITU-T G.781

- PLFS implementations other than SyncE are for further study.
  All network elements in the fronthaul network need to use the same PLFS method for optimal interoperability and performance.
- ITU-T G.8271.1 shall be consulted for guidance on balancing the need to meet a target air interface performance at the O-RU with the maximum number of hops allowed. The network operator shall plan for specific fronthaul network deployment based on this guidance.
- The use of eEECs is recommended as they generate less noise (Time error, FFO) in normal operation and during network rearrangements than EECs, allowing a larger number of hops while meeting a given time error budget.

11.2.4 Profiles

11.2.4.1 Physical Layer Frequency Signals (PLFS)

An implementation providing SyncE shall:

- Support ITU-T G.781 Option 1 Quality Level (per clause 5.4 Quality Code of ITU-T G.781)
  - Support ITU-T G.781 other options, is for further study. ITU-T G.8271.1 analysis has only been done with a synchronous Ethernet network based on option 1 EECs or eEECs.
- Support ITU-T G.8264 message types, format, transmission and reception (per Table 11.3 of ITU-T G.8264) if SyncE is used.
  - Support of extended SSM TLV (per the 2017 edition of ITU-T G.8264) is optional, and For Further Study

Alternate PLFS implementations are for further study.

NOTE: The accepted QL may be limited to PRC or better value for optimal operation, while other values may be used under at the operator’s discretion considering the holdover capability of the SyncE source and the way the end applications use SyncE.

11.2.4.2 PTP

11.2.4.2.1 Full Timing Support

Use of IEEE 1588 or PTP for time/phase synchronization shall be according to its clauses referred by ITU-T G.8275.1 (Full Timing Support).

Notes:

- The T-TSC inside the O-RU and O-DU are considered as T-TSC inside 3GPP end application modules. Such T-TSC may not provide a 1PPS measurement interface, and ITU-T G.8273.2 Appendix IV applies: the combined performance within each module may not behave as a stand-alone T-TSC described in the normative section of the recommendation.
- PLFS and PTP distribution shall be supported (with a few derogations listed in Table 8-2), either using congruent or non-congruent topologies, as per ITU-T G.8271.1.
- O-RUs shall support L2 (Ethernet) and multicast communication when acting as an ITU-T G.8275.1 PTP subordinate in a fronthaul network, which is considered “mandatory” in this version of the CUS-Plane specification.
- O-RUs and O-DUs supporting LLS-C1, LLS-C2, and LLS-C3 shall follow ITU-T G.8275.1 clause 6, including requirements related to one-step and two-step clock mode.

- For configurations LLS-C1 and LLS-C2, the O-DU shall support L2 (Ethernet) and multicast communication when acting as an ITU-T G.8275.1 PTP master in a fronthaul network to synchronize the O-RUs, which is considered “mandatory” in this version of the CUS-Plane specification.
- For all configurations, O-DU shall support L2 (Ethernet) and multicast communication when acting as an ITU-T G.8275.1 subordinate in a fronthaul or other network, which is considered “optional” in this version of the CUS-Plane specification.

11.2.4.2.2 Partial Timing Support

Support of Partial Timing Support using ITU-T G.8275.2 Telecom Profile is currently considered as permissible but requires additional considerations:

- Partial Timing Support allows switches with no T-BC or T-TC, hence there is no guarantee of synchronization performance based on ITU-T standard specification such as G.8273.2. As a result, the system operator shall ensure the network components will have adequate performance to meet frequency and phase error budgets to allow an accurate detection of frequency accuracy and phase for proper network operation. Such budgets and implications on performance require further investigation.
- O-RUs may support L3 (UDP/IP) and unicast communication when acting as an ITU-T G.8275.2 PTP subordinate in a fronthaul network, which is considered “optional” in this version of the CUS-Plane specification.
- O-RUs and O-DUs supporting LLS-C1, LLS-C2, and LLS-C3 shall follow ITU-T G.8275.2 clause 6, including requirements related to one-step and two-step clock mode.
- For configurations LLS-C1 and LLS-C2, the O-DU may support L3 (UDP/IP) and unicast communication when acting as an ITU-T G.8275.2 PTP master in a fronthaul network to synchronize the O-RUs, which is considered “optional” in this version of the CUS-Plane specification.
- For all configurations, the O-DU may support L3 (udp/ip) and unicast communication when acting as an ITU-T G.8275.2 PTP subordinate, which is considered “optional” in this version of the CUS-Plane specification.

Note finally that Partial Timing Support is not finalized in the ITU, which has considered this timing method only for relatively coarse timing accuracy (1.5 µsec). Using Partial Timing Support for S-plane fronthaul distribution to the O-RU to satisfy tighter requirements than levels of accuracy 4 and 4A (as per ITU-T G.8271 Table 1) is for further study.

11.2.5 Synchronization Accuracy

The parameters Time Error and other derived metrics are used in the subsequent clauses of this document. The definition of Time Error function, TE(*t*) is given in clause 3.1. For a synchronized clock or timing signal, the Time Error function is composed of several different error components which contribute to the total, and these individual components of time error have limits which are specified in subsequent clauses. To clarify the specification tables later in this clause, definitions of these additional time error parameters are provided here.

- **Max|TE|**: The maximum absolute value of the time error function, TE(*t*) (i.e. the furthest point away from zero, either positive or negative). See ITU-T G.8260, clause 3.1.20.
- **Constant Time Error, cTE:** The mean value of the time error function, TE(*t*), over a measurement period. See ITU-T G.8260, clause 3.1.20.
- **Dynamic Time Error, dTE(*t*):** The change in the time error function, TE(*t*), over a measurement period. See ITU-T G.8260, clause 3.1.20.
  This is divided into high and low frequency components by filtering
  - **$dTE_L(t)$:** The dynamic time error after low-pass filtering.
    For packet timing signals (e.g. PTP), the measurement filter bandwidth is typically 0.1Hz, and $dTE_L(t)$ is generally further characterized using MTIE or TDEV.
  - **$dTE_H(t)$:** The dynamic time error after high-pass filtering.
    For packet timing signals (e.g. PTP), the measurement bandwidth is typically 0.1Hz, and $dTE_H(t)$ is generally reported as a peak-to-peak or zero-to-peak value. This document specifies these parameters using zero-to-peak values.

- **TE_L(*t*):** The slow changes in time error after low-pass filtering.
  For packet timing signals (e.g. PTP), the measurement bandwidth is typically 0.1Hz.
  - **Max|TE_L|:** The maximum absolute value of the low pass filtered time error function

Further information about the accumulation of time error through a synchronization network can be found in Appendix IV of G.8271.1.

11.2.5.1 Jitter

Within the O-RAN fronthaul network, all network equipment (NE) supporting SyncE transport across the network shall comply with input and output jitter requirements specified in ITU-T G.8262 (for EEC) or ITU-T G.8262.1 (for eEEC).

Alternate PLFS implementations are for further study.

11.2.5.2 Wander

Within the O-RAN fronthaul network, all network equipment (NE) supporting SyncE transport across the network shall comply with input and output wander requirements specified in ITU-T G.8262 (for EEC) or ITU-T G.8262.1 (for eEEC).

Alternate PLFS implementations are for further study.

11.2.5.3 Air interface frequency error

The O-RAN fronthaul network shall ensure O-RU meeting a +/-50ppb air interface frequency error requirement. 3GPP TS 36.104 (for LTE macro cells) and TS 38.104 (for 5G macro ceO) specify +/-50ppb as the short-term average error in 1ms duration applicable to both LTE and 5G technologies. Refer to clause 11.3.2 for more detail information.

11.2.5.4 Air interface maximum time error

The O-RAN fronthaul network shall ensure O-RU meeting the following air interface time alignment error (|TAE| absolute or relative) requirements based on different features in LTE and 5G technologies. For features covered by 3GPP, they are specified in TS 36.104 (for LTE) and TS38.104 (for 5G).

The following figure shows the reference points to define the network time error |TE| vs air interface time alignment error |TAE| and the concept of relative vs absolute.

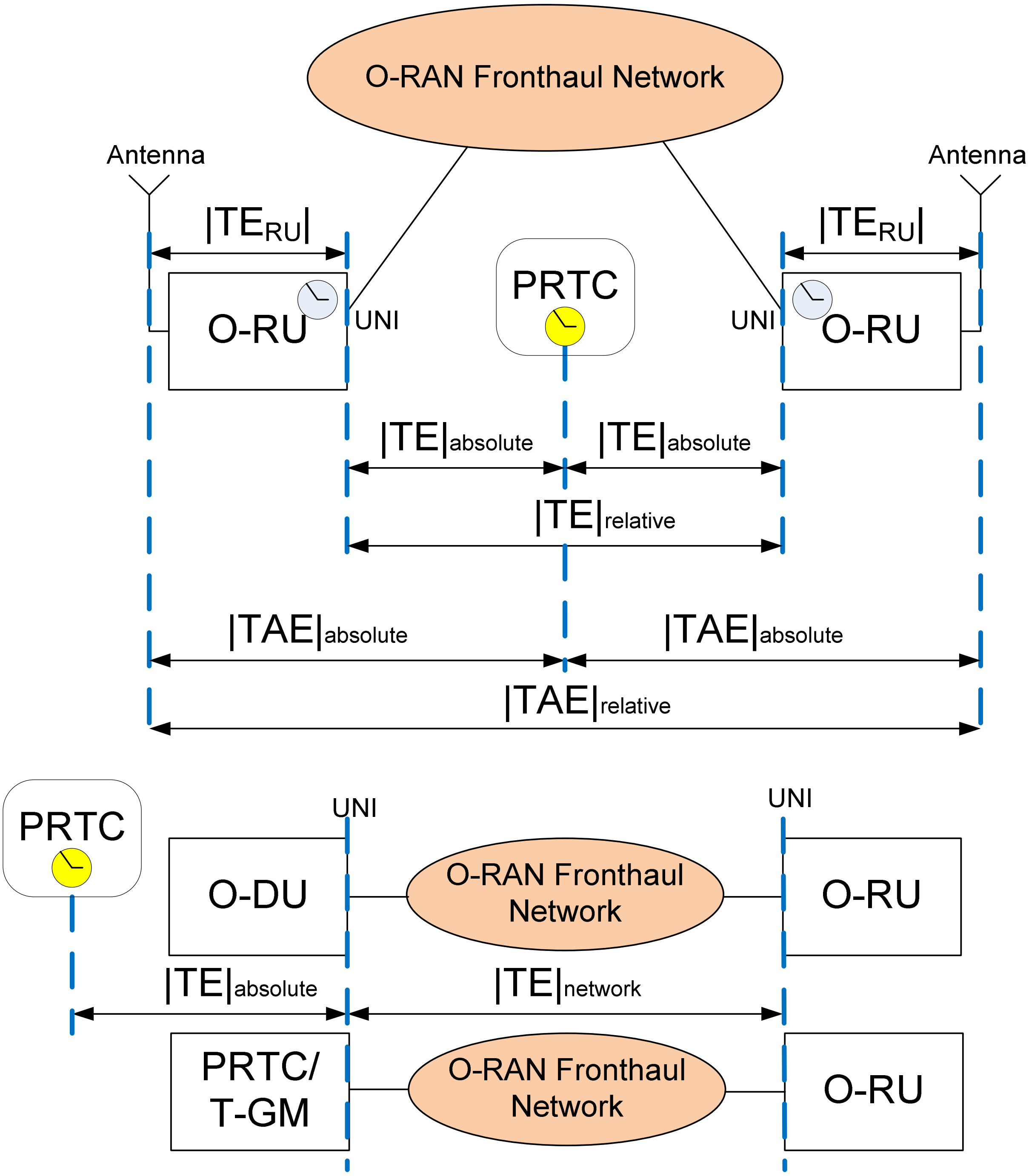


Figure 11-4 : Definition of |TE| and |TAE| and UNI

For LTE features:

Table 11-1 : LTE features with time alignment error requirement at the air interface

LTE Features	|TAE| relative or absolute	Corresponding Timing Category in 802.1CM/eCPRI standard (Informative)
TDD	Relative |TAE| ≤ 3µs (cell radius < 3km) Relative |TAE| ≤ 10us (cell radius > 3km) (note 1) TS36.133	Timing Category C
Dual connectivity	Relative |TAE| ≤ 3µs (note 1) TS36.133	Timing Category C
MIMO or Tx Diversity	Relative |TAE| ≤ 65ns (note 2, note 4) TS36.104	
CA (intraband contiguous)	Relative |TAE| ≤ 130ns (note 2, note 5) TS36.104	Timing Category A (note 3)
CA (interband or intraband non-contiguous)	Relative |TAE| ≤ 260ns (note 2) TS36.104	Timing Category B (note 3)
OTDOA	Absolute |TAE| at O-RU antenna << 1.5us, ~100-200 ns (not defined by 3GPP)	Not covered since it is not defined by 3GPP

NOTE 1: For TDD and dual connectivity features, relative |TAE| requirement is applied to any pair of O-RUs within RAN without any cluster boundary limit. Hence relative |TAE| spec indirectly leads to a per O-RU requirement of absolute |TAE| at O-RU antenna = 1.5µs.

NOTE 2: When these features are supported within 1 O-RU, relative |TAE| is impacted by O-RU internal |TE| only.

NOTE 3: When these features are supported by multiple cooperating O-RUs, relative |TAE| is also impacted by network relative |TE| where 802.1CM Timing Category is applicable to limit the network error contribution.

NOTE 4: In the current specification, it is assumed that “MIMO or Tx Diversity” is applicable within a single O-RU. MIMO or Tx Diversity with multiple cooperating O-RUs is for further study.

NOTE 5: Applicable when the aggregated carriers are generated by co-located O-RUs (i.e, where the distance between the O-RUs does not create significant differential delay between the O-RUs and the UE, when compared with the timing requirement).

For 5G features:

Table 11-2 : 5G features with time alignment error requirement

5G Features	|TAE| relative or absolute	Corresponding Timing Category in 802.1CM/eCPRI standard (Informative)
TDD	Relative |TAE| ≤ 3µs (note 1) TS38.133	Timing Category C
Dual connectivity	Relative |TAE| ≤ 3µs (note 1) TS38.133	Timing Category C
MIMO and Tx Diversity	Relative |TAE| ≤ 65ns (note 2, note 4) TS38.104 [8]	

CA (intraband contiguous per base station type)	Relative |TAE| ≤ 130ns (note 2, note 5) (FR2 as defined in 3GPP) TS38.104 [8]	Timing Category A (note 3)
CA (intraband contiguous per base station type)	Relative |TAE| ≤ 260ns (note 2, note 5) (FR1 as defined in 3GPP) TS38.104 [8]	Timing Category B (note 3)
CA (interband or intraband non-contiguous)	Relative |TAE| ≤ 3µs (note 2) TS-38.104 [8]	Timing Category C (note 3)
OTDOA	Absolute |TAE| at eRE antenna << 1.5us (not defined by 3GPP)	Not covered since it is not defined by 3GPP

NOTE 1: For TDD and dual connectivity features, relative |TAE| requirement is applied to any pair of O-RUs within RAN without any cluster boundary limit. Hence relative |TAE| spec indirectly leads to a per O-RU requirement of absolute |TAE| at O-RU antenna = 1.5µs.

NOTE 2: When these features are supported within 1 O-RU, relative |TAE| is impacted by O-RU internal |TE| only.

NOTE 3: When these features are supported by multiple cooperating O-RUs, relative |TAE| is also impacted by network relative |TE| where 802.1CM Timing Category is applicable to limit the network error contribution.

NOTE 4: In the current specification, it is assumed that “MIMO or Tx Diversity” is applicable within a single O-RU. MIMO or Tx Diversity with multiple cooperating O-RUs is for further study.

NOTE 5: Applicable when the aggregated carriers are generated by co-located O-RUs (i.e, where the distance between the O-RUs does not create significant differential delay between the O-RUs and the UE, when compared with the timing requirement).

11.3 Time and Frequency Synchronization Requirements

11.3.1 Allowed PTP and PLFS clock types and clock classes

A network element (NE) may use the following clock types and classes to support PTP and PLFS, and can be used among other such NEs to build an O-RAN-compliant fronthaul network meeting end-to-end frequency synchronization requirements as well as time synchronization requirements at the air interface.

- EEC (per ITU-T G.8262)
- eEEC (per ITU-T G.8262.1)
- PRC (per ITU-T G.811)
- PRTC* (per ITU-T G.8272 and G.8272.1))
- T-BC (per ITU-T G.8273.2)
- T-TSC (per ITU-T G.8273.2)
- T-TC** (per ITU-T G.8273.3)

*G.8272 specifies 2 types of PRTC: PRTC-A and PRTC-B. The time output of a PRTC-B is more accurate than that of a PRTC-A. PRTC-B is suitable for locations where it is possible to guarantee optimized environmental conditions (e.g., controlled temperature variation in indoor deployments). Typical examples are central location and large aggregation sites. This means that PRTC-B may be challenging and impractical for LLS-C4 deployments (e.g., due to generally outdoor deployments). G.8272.1 specifies 2 types of ePRTC, ePRTC-A and ePRTC-B. The main difference between the PRTCs and the ePRTCs is the input from an external autonomous primary reference clock (e.g., cesium clock) required in case of ePRTCs. This means that ePRTCs are only suitable for deployment in centralized location (therefore not applicable as a solution for LLS-C4).

**According to ITU-T G.8271.1 and IEEE 802.1CM, T-TC is allowed as T-BC replacement and T-TC/T-BC accuracy performance is equivalent. O-RAN allows T-TC as accepted clock type. However, using T-TC has potential issue with broadcast storms in a bridging network as stated in G.8271.1 and hence guidance by Appendix I in G.8275.1 should be followed.

The time error analysis in Annex H only covers T-BC deployments, and not the T-TC one (which is For Further Study)

O-DU and O-RU are considered end points in an O-RAN-compliant fronthaul network. O-DU and O-RU can support specific PTP clock and specific classes based on O-DU and O-RU synchronization capability options.

Alternate PLFS implementations are for further study.

O-DU synchronization capability options:

For configurations LLS-C1 and LLS-C2: PLFS and PTP Master.

- In case of local PRC/PRTC (typically a local GNSS receiver) used as frequency and time source: O-DU acts a PLFS and PTP master. The performance requirements are specified in clause 11.3.2.
- In case of a remote PRC/PRTC used as frequency and time source via a Full Timing Support packet network (defined by ITU-T G.8271.1): O-DU shall act as an embedded end application with better wander and jitter filtering capability, as per Appendix IV of ITU-T G.8273.2 (thus acting more like a combined PTP Subordinate /Master than a T-BC). Note that acting as a true ITU-T G.8273.2 T-BC does not guarantee the frequency accuracy required in 9.3.2, and is therefore For Further Study.
- In the case of a remote PRC/PRTC used as frequency and time source via a Partial Timing Support packet network (defined by ITU-T G.8271.2): O-DU may act as an IWF P-F InterWorking Function to drive Full Timing Support (defined by ITU-T G.8271.1) to the Fronthaul.

For both Full Timing Support and Partial Timing support, the remote PRC/PRTC used as frequency and time source can be located anywhere in the network.

For configuration LLS-C3 and LLS-C4:

- O-DU can be synchronized from any possible time source (either local or remote PRC/PRTC-traceable one, like for LLS-C1 and LLS-C2).
- O-DU does not need to meet the 3GPP frequency and TAE target specification as required in the O-RU. However, a more relaxed phase alignment between O-DU and O-RU timing should be kept to avoid data buffer overflow/underflow (impact to delay management topic). Whether O-DU is or is not involved in the synchronization of the O-RU, it shall meet the time error limit for latency management as specified in clause 4.3.

O-RU synchronization capability options:

- For configuration LLS-C1, LLS-C2 and LLS-C3 with full timing support: (T-TSC is either embedded or external, as per ITU-T G.8271.1 and IEEE 802.1 CM). In the case where O-RU includes a T-TSC, this one is “embedded in end application” as specified in G.8273.2 Appendix IV and include additional necessary filtering function to ensure 3GPP air interface compliance. Under investigation is what is needed for Partial Timing Support because there is not yet any available ITU recommendation.
- For configuration LLS-C4, O-RU is simply synchronized from the local PRC/PRTC using proprietary timing interface.

O-RU shall support network timing as mandatory synchronization capability to cover configuration LLS-C1, LLS-C2 and LLS-C3. Local PRC/PRTC (typically a GNSS receiver) shall be optional synchronization capability to cover configuration LLS-C4.

11.3.2 Frequency and Time Synchronization Requirements across fronthaul network elements

11.3.2.1 Configurations LLS-C1 and LLS-C2

Based on IEEE 802.1CM and ITU-T G.8271.1 guidance, the following table summarizes the frequency and time error budgets across different elements of an O-RAN-compliant fronthaul network.

- O-DU : shall not exceed allocated frequency error budget and time error budget (for chosen air interface target)
- O-RU : shall not exceed allocated frequency error budget and time error budget (for chosen air interface target). Note that the two O-RU classes described by IEEE 802.1 CM are deployment examples. The requirement shall be met only for O-RUs compliant with the limits of these classes.

- O-RAN fronthaul network: shall not exceed network limit to satisfy both frequency error budget and time error budget (for chosen air interface target). Allowed number of switches in a deployment can be derived based on allowed network limit vs chosen switch specification. Annex H shows the analysis of number of switches based on T-BC Class B and C switches.

The following table covers budget allocation for configuration LLS-C1 and LLS-C2 (Refer to **Figure 11-4** for reference point definition). **Requirements are in BOLD** :

Table 11-3 : Frequency and time error budget allocation (for topology configuration LLS-C1 and LLS-C2)

Frequency error budget allocation			
Timing Reference	**O-RAN fronthaul network contribution limit**	**O-RU**	**Air interface target**
O-DU PTP/PLFS master class A, with |freq error| ≤ 15ppb • @ O-DU UNI (see NOTES 1, 8, 9)	**total $\|dTE_{L+H}\| \le 45$ns** • Between O-DU UNI and O-RU UNI • For LLS-C2: Allowed number of hops (see NOTE 2) • For LLS-C1: single hop by definition	**O-RU |freq error| ≤ 35ppb** including both • FFO after O-RU filtering of dTE_{L+H} @ O-RU UNI • O-RU internal additive frequency error (see NOTE 3)	±50ppb per 3GPP spec
O-DU PTP/PLFS master class B, with |freq error| ≤ 5ppb • @ O-DU UNI (see NOTES 1, 8, 9)	**total $\|dTE_{L+H}\| \le 57$ns** • Between O-DU UNI and O-RU UNI • For LLS-C2: Allowed number of hops (see NOTE 2) • For LLS-C1: single hop by definition	**O-RU |freq error| ≤ 45ppb** including both • FFO after O-RU filtering of dTE_{L+H} @ O-RU UNI • O-RU internal additive frequency error (see NOTE 3)	±50ppb per 3GPP spec
Time error budget allocation			
Timing Reference	**O-RAN fronthaul network contribution limit**	**O-RU (All allowed class options are shown) (see NOTE 6)**	**Air interface target**
No relative $\|TE_L\|$ contribution by O-DU since O-DU is common PTP and PLFS master to all co-operated O-RU (NOTE 1)	**Relative $\|TE_L\| \le 60$ns** • Between 2 O-RUs UNI • per IEEE 802.1CM • For LLS-C2: Allowed number of hops (see NOTE 4) • For LLS-C1: single hop by definition	**O-RU |TE|** includes as per IEEE 802.1CM (see NOTE 5)**:** • **(enhanced O-RU can be used) O-RU |TE| ≤ 35ns**	Per IEEE 802.1CM Category A
	Relative $\|TE_L\| \le 100$ns (using regular O-RU) **Relative $\|TE_L\| \le 190$ns (using enhanced O-RU)** • Between 2 O-RUs UNI • per IEEE 802.1CM • For LLS-C2: Allowed number of hops (see NOTE 4) • For LLS-C1: single hop by definition	**O-RU |TE|** includes as per IEEE 802.1CM (see NOTE 5)**:** • **Either regular O-RU O-RU |TE| ≤ 80ns** • **Or enhanced O-RU O-RU|TE| ≤ 35ns**	Per IEEE 802.1CM Category B
For LLS-C2: absolute $\|TE_L\| \le 1.325$us **For LLS-C1: absolute $\|TE_L\| \le 1.420$us** • @ O-DU UNI	**Network $\|TE_L\| \le 95$ns (using regular O-RU)** **Network $\|TE_L\| \le 140$ns (using enhanced O-RU)**	**O-RU |TE|** includes as per IEEE 802.1CM (see NOTE 5 and 7)**:** • **Either regular O-RU O-RU |TE| ≤ 80ns** • **Or enhanced O-RU O-RU**	Absolute |TAE| at O-RU antenna = 1.5µs derived from 3GPP

• includes holdover budget • see NOTES 8, 9	• Between O-DU UNI and O-RU UNI • For LLS-C2: Allowed number of hops (see NOTE 4) • For LLS-C1: single hop by definition	**|TE| ≤ 35ns**	(Same as IEEE 802.1CM Category C)

NOTE 1: O-DU implements PTP and PLFS master function in this use case.
Two frequency error limits are defined in this specification, conservative class A with 15ppb, and more advanced class B with 5 ppb.
Both frequency error limits and $|TE_L|$ time error limits are measured on the O-DU UNI after applying a first-order measurement low-pass filter bandwidth of 0.1Hz to the time samples.
Measurement condition is applicable when O-DU Master Clock is either in locked state or holdover, but excluding rare and temporary transients:

- Resynchronization to recovered source after holdover.
- Resynchronization to newly selected source after failure of the previous one, in case of redundancy.

It is considered that all master ports of the O-DU are fully synchronized together, and there is no port-to-port time error (see Annex H).

NOTE 2: dTE_{L+H} = accumulated dynamic time error (dTE_L and dTE_H) of a T-BC clock chain (excluding O-DU contribution) based on G.8271.1 Appendix IV calculation method. Refer to Annex H for detailed analysis for maximum number of T-BC Class B and C switches.

NOTE 3: The O-RU requirement is not specified in IEEE 802.1CM or ITU-T G.8271.1. O-RAN only specifies the total O-RU frequency error budget to allow design flexibility by different solution vendors. The O-RU solution vendor can make a tradeoff between FFO (Fractional Frequency Offset after applying O-RU filtering) and internal additive frequency error as long as the total frequency error budget with network total |dTE| limit is met. Refer to Annex H for analysis.

NOTE 4: Refer to Annex H for analysis of the number of switches to satisfy the allowed network limit. For the Time Error budget (both network and relative time errors) it is considered that the O-RU attenuates the dTE_H part of the noise present at its input, and therefore the budget uses $|TE_L|$ using a 0.1 Hz low-pass filter as described in ITU-T G.8271.1.

NOTE 5: The T-TSC are considered T-TSC embedded in end application as specified in G.8273.2 Appendix IV and the performance may not behave as a standalone T-TSC described in the normative section of the recommendation. However, to ensure interoperability among O-RAN O-RU vendors, the agreed performance will be based for O-RUs compliant with the classes defined in eCPRI and 802.1CM recommendation: under eCPRI Transport Network Requirements Specification [2] and IEEE 802.1CM [11], a regular O-RU with |TE|= 80 ns and an enhanced O-RU with |TE| = 35 ns. The O-RU time error accumulates linearly with the other contributors in the network.

NOTE 6: O-RU |TE| is the total budget proposed in IEEE 802.1CM for two examples of O-RU classes. O-RAN recommends O-RU internal split as shown in the table based on eCPRI and 802.1CM guidance. It is also acceptable for RAN solution vendor to use a different O-RU internal split to meet the same O-RU |TE| total budget.

NOTE 7: O-RAN recommends meeting the specified budget split between O-DU and O-RU |TE| across all categories based on eCPRI and 802.1CM guidance. A different budget split than what O-RAN recommends may meet the overall radio interface requirements, but shall be considered under the operator's responsibility. Typically, configuration LLS-C1 and LLS-C2 with category C target may allow more flexibility in O-DU $|TE_L|$, network $|TE_L|$ and O-RU |TE| budget split as long as the network limit (network $|TE_L|$ for time error and $|dTE_{L+H}|$ for frequency error) is still respected. For example, if an operator can guarantee the network $|TE_L|$ < 75ns which is below the 95ns limit set in this O-RAN specification, 20ns (95ns-75ns) can be moved into the O-DU $|TE_L|$ and/or O-RU |TE| budget. Inter-operability can be guaranteed by the network operator when all participating O-DU and O-RU vendors design to meet the operator-chosen budget split.

NOTE 8: In case the O-DU is synchronized using a local PRTC (typically a GNSS receiver or local T-GM), the time error limit at the input of the O-DU is specified as being Reference point A (or B) defined by ITU-T G.8271, ITU-T G.8271.1 and G.8271.2. The O-DU acting as a T-GM in normal operating mode may meet Time Error limits slightly larger than what is specified in ITU-T G.8272, and expected to be significantly lower than 1420 ns or 1325 ns.

NOTE 9: In case the O-DU is synchronized using network distribution (typically PTP and PLFS), the network time error limit at the input of the O-DU is specified as being Reference point C defined by ITU-T G.8271, ITU-T G.8271.1 for Full Timing Support or ITU-T G.8271.2 for Partial Timing Support. Similarly to NOTE 4, the relevant limit of the O-DU is the 0.1 Hz low-pass filtered $|TE_L|$, which implies that the O-DU low-pass filtering bandwidth is lower than 0.1 Hz (per clause 11.2.2.1).

The synchronization accuracy for the OTDOA feature is not defined by 3GPP. Hence absolute |TAE| at O-RU antenna and the corresponding network $|TE_L|$ are out of scope in O-RAN specification and are not covered in this analysis. The two O-RU |TE| classes indicated earlier may be considered also for this use case. In a deployed network, an operator can choose a target absolute |TAE| and then derive the corresponding network $|TE_L|$ and the allowed number of hops and required types of T-BC(s)/T-TC(s) based on ITU-T G.8271.1 guidance.

With Partial Timing Support, when using non-T-BC switches, network contribution limit requires further investigation.

11.3.2.2 Configuration LLS-C3

Based on IEEE 802.1CM and ITU-T G.8271.1 guidance, the following table summarizes the frequency and time error budgets across different elements of an O-RAN-compliant fronthaul network.

- PRTC/T-GM : shall not exceed allocated frequency error budget and time error budget (for chosen air interface target)
- O-RU : shall not exceed allocated frequency error budget and time error budget (for chosen air interface target). Note that the two O-RU classes described by IEEE 802.1 CM are deployment examples. The requirement shall be met only for O-RUs compliant with the limits of these classes.

O-RAN fronthaul network : shall not exceed network limit to satisfy both frequency error budget and time error budget (for chosen air interface target). The network limits for time error, applicable at the input of the O-RU, are those defined in clause 7.3 and 7.5 of G.8271.1 (Network limits at reference point C and for deployment cases with the PRTC deployed in the access network; only one of these would be applicable depending on the specific deployment), where the O-RU corresponds to the End Application with integrated PTP clock (Deployment Case 1, see Figure 7-1 in G.8271.1). The limits in G.8271.1 are expressed in terms of MTIE, max $|TE_L|$ and peak-to-peak dTE_H. Applicability of G.8271.1 peak-to-peak $dTE_H(t)$ limit is for further study. Further details and references are provided in Annex H. The requirements provided in this clause are aligned with IEEE 802.1CM.

- O-DU output measurement signal (1PPS): there is no such allocated frequency error budget as for LLS-C1/C2. Only time error shall be within ±1500ns limits as specified in clause 4.3. O-DU and O-RU can be traceable to a different PRTCs, as long as they share the same timescale.
  In case the O-DU is synchronized using a local PRTC (typically a GNSS receiver or local T-GM), the time error limit at the input of the O-DU is specified as being Reference point A (or B) defined by ITU-T G.8271.1 and G.8271.2.
  In case the O-DU is synchronized using network distribution (typically PTP and PLFS), the network time error limit at the input of the O-DU is specified as being Reference point C defined by ITU-T G.8271.1 for Full Timing Support or ITU-T G.8271.2 for Partial Timing Support.
- O-DU : shall not exceed allocated time error budget required by latency management as specified in clause 4.3.

The following table covers budget allocation for configuration LLS-C3 (Refer to **Figure 11-4** for reference point definition). **Requirements are in BOLD** :

Table 11-4 : Frequency and time error budget allocation (for topology configuration LLS-C3)

Frequency error budget allocation			
Timing Reference	**O-RAN fronthaul network contribution limit**	**O-RU**	**Air interface target**
PRTC/T-GM |freq error|≤ 2ppb • @ PRTC/T-GM UNI (see NOTE 1)	**total $|dTE_{L+H}|$ ≤ 63ns** (see NOTE 4) • Between PRTC output and O-RU UNI • Include PRTC/T-GM MTIE contribution • Allowed number of hops (see NOTE 2)	**O-RU |freq error| ≤48ppb** including both • FFO after O-RU filtering of dTE_{L+H} @ O-RU UNI • O-RU internal additive frequency error (see NOTE 3)	±50ppb per 3GPP spec
Time error budget allocation			
Timing Reference	**O-RAN fronthaul network contribution limit**	**O-RU** **(All allowed class options are shown)** **(see NOTE 6)** **O-DU (see NOTE 8)**	**Air interface target**
No relative |TE| contribution by PRTC/T-GM as long as PRTC/T-GM is common PTP and PLFS master to all co-operating O-RU (see NOTE 1)	**Relative $|TE_L|$ ≤60ns** • Between 2 O-RUs UNI • per IEEE 802.1CM • Allowed number of hops (see NOTE 5)	**O-RU |TE|** includes as per IEEE 802.1CM: **(enhanced O-RU can be used)** • **O-RU |TE| ≤ 35ns**	Per IEEE 802.1CM Category A
	Relative $|TE_L|$ ≤100ns (for regular O-RU) **Relative $|TE_L|$ ≤190ns (for enhanced O-RU)** • Between 2 O-RUs UNI • per IEEE 802.1CM • Allowed number of hops (see NOTE 5)	**O-RU |TE|** includes as per IEEE 802.1CM (see NOTE 6)**:** • **Either regular O-RU |TE| ≤ 80ns** • **Or enhanced O-RU |TE| ≤ 35ns**	Per IEEE 802.1CM Category B
Absolute |TE| ≤ 100ns or 40 ns (see NOTE 9) • @ PRTC/T-GM UNI • PRTC/T-GM spec per G.8272 (see NOTE 1) • Not including holdover budget	• **Network $|TE_L|$ ≤ 1100ns at the O-RU UNI** (see NOTE 7, NOTE 10) • **Short Clock Chain Network $|TE_L|$ ≤ 100ns at the O-RU UNI** (see NOTE 10, NOTE 11) • Allowed number of hops (see NOTE 5)	**O-RU |TE|** includes as per IEEE 802.1CM (see NOTE 6)**:** • **Either regular O-RU |TE| ≤ 80ns** • **Or enhanced O-RU |TE| ≤ 35ns**	Absolute |TAE| at O-RU antenna = 1.5µs derived from 3GPP (Same as IEEE 802.1CM Category C)
Absolute |TE| ≤ 100ns or 40 ns (see NOTE 9) • @ PRTC/T-GM UNI • PRTC/T-GM spec per G.8272 (see NOTE 1) Not including holdover budget	**Network |TE| is out of scope** • O-DU may get its synchronization from either local or remote PRTC	**O-DU absolute |TAE|** @ O-DU UNI < **±1.5µs** (see NOTE 8)	Absolute |TAE| at O-DU = 1.5µs (Same as IEEE 802.1CM Category C)

NOTE 1: PRTC/T-GM follows G.8272 specification for PRTC-A and PRTC-B, or G.8272.1 for ePRTC. MTIE specification is considering dynamic error during lock condition and its contribution is covered under dTE_{L+H} in note 2. However, there is possible semi-static frequency error which is not part of MTIE specification. Therefore, it is included here.

Both ITU-T G.8272 and G.8272.1 specify that for the combined PRTC and T-GM function, time error samples are measured through a moving-average low-pass filter of at least 100 consecutive time error samples. Assuming a message rate of 16 PTP messages per second as described in G.8275.1, the low-pass filtering bandwidth is 0.05 Hz, which is not same as a 0.1Hz low-pass-filtered $|TE_L|$. (The frequency response of a moving average filter is approximately similar to

a first-order linear filter of bandwidth 1/(πT), where T is the width of the moving average window).
In case the PRTC/T-GM is a multiple-port device, it is considered that all master ports are fully synchronized together, and there is no port-to-port time error.
Refer to Annex H for detailed analysis.

NOTE 2: dTE_{L+H} = accumulated dynamic time error (dTE_L and dTE_H) of a T-BC clock chain, including PRTC/T-GM contribution, to O-RU UNI (or O-DU UNI) based on G.8271.1 Appendix IV calculation method. Refer to Annex H for analysis for maximum number of T-BC Class B and C switches.

NOTE 3: The O-RU requirement is not specified in IEEE 802.1CM or ITU-T G.8271.1. O-RAN only specifies the total O-RU frequency error budget to allow design flexibility by different solution vendors. The O-RU solution vendor can make a tradeoff between FFO (Fractional Frequency Offset after applying O-RU filtering) and O-RU internal additive frequency error as long as the total frequency error budget with network total |dTE| limit is met. Refer to Annex H for analysis.

NOTE 4: The O-RAN fronthaul network contribution limits are stricter and therefore not consistent with peak-to-peak $dTE_H(t)$ limits defined in G.8271.1 for long synchronization chain (see clause 7.3 in G.8271.1). In order to tolerate the higher G.8271.1 limits (applicable to long network) the O-RU would be required to provide higher filtering capabilities. This is an option that may be considered but not further described in this specification. This requirement assumes shorter clock chains.

NOTE 5: Refer to Annex H for analysis of the number of switches to satisfy the allowed network limit.

NOTE 6: The T-TSC are considered T-TSC embedded in end application as specified in G.8273.2 Appendix IV and the performance may not behave as a standalone T-TSC described in the normative section of the recommendation. However, to ensure interoperability among O-RAN O-RU vendors, the agreed performance will be based for O-RUs compliant with the classes defined in eCPRI and 802.1CM recommendation: under eCPRI Transport Network Requirements Specification [2] and IEEE 802.1CM [11], a regular O-RU with |TE|= 80 ns and an enhanced O-RU with |TE| = 35 ns. The O-RU time error accumulates linearly with the other contributors in the network.

NOTE 7: Network $|TE_L|$ = 1100 ns is specified by G.8271.1 in clause 7.3 for the max $|TE_L|$. An O-RU may consume only 35 ns (enhanced O-RU) or 80 ns (regular O-RU). $\max|TE_L| \leq 1\,100$ ns may leave extra budget that could be allocated for instance to holdover that could be especially relevant in case of TDD applications. This additional budget would be added on top of the 35 ns (enhanced O-RU) or 80 ns (regular O-RU). Specification for holdover is for further study.

NOTE 8: TAE between O-DU and O-RU limit (as specified in clause 4.3) translates into absolute |TAE| for both O-DU and O-RU with half the budget. This is measured at any available O-DU output signal (either 1PPS or O-DU UNI).

NOTE 9: PRTC-B (max|TE| = 40 ns) is assumed in case of short clock chains by G.8271.1.

NOTE 10: the contribution from the network includes the noise generated by the PRTC/T-GM.

NOTE 11: the network limits for short clock chain networks are specified in clause 7.5 of G.8271.1.

The synchronization accuracy for the OTDOA feature is not defined by 3GPP. Hence absolute |TAE| at O-RU antenna and the corresponding network $|TE_L|$ are out of scope in O-RAN specification and therefore not covered in this analysis. The two O-RU |TE| classes indicated earlier may be considered also for this use case. In a deployed network, operator can choose a target absolute |TAE| and then derive the corresponding network $|TE_L|$ and the allowed number of hops and required types of T-BC(s)/T-TC(s) based on ITU-T G.8271.1 guidance.

With Partial Timing Support, when using non-T-BC switches, network contribution limit requires further investigation.

11.3.2.3 Configuration LLS-C4

The following table summarizes the frequency and time error budgets across different elements.

- O-DU output measurement signal (1PPS): there is no such allocated frequency error budget as for LLS-C1/C2. Only time error shall be within ±1500ns limits as specified in clause 4.3.
  In case the O-DU is synchronized using a local PRTC (typically a GNSS receiver or local T-GM), the time error limit at the input of the O-DU is specified as being Reference point A (or B) defined by ITU-T G.8271.1 and G.8271.2.
  In case the O-DU is synchronized using network distribution (typically PTP and PLFS), the network time error limit at the input of the O-DU is specified as being Reference point C defined by ITU-T G.8271.1 for Full Timing Support or ITU-T G.8271.2 for Partial Timing Support

- O-RU : shall not exceed allocated frequency error budget and time error budget (for chosen air interface target) Note that the two O-RU classes described by IEEE 802.1 CM are deployment examples. The requirement shall be met only for O-RUs compliant with the limits of these classes

The following table covers budget allocation for configuration LLS-C4:

Table 11-5 : Frequency and time error budget allocation (for topology configuration LLS-C4)

Frequency error budget allocation			
Timing Reference in network or O-DU	**O-RAN fronthaul network**	**O-RU**	**Air interface target**
No dependency	No dependency	O-RU |freq error| ≤ 50ppb including both • O-RU filtering on local PRTC noise • O-RU internal additive frequency error (NOTE 3, 4)	±50ppb per 3GPP spec
Time error budget allocation			
Timing Reference in network or O-DU	**O-RAN fronthaul network**	**O-RU (See NOTE 1,2, 3, 4) O-DU (see NOTE 5)**	**Air interface target**
No dependency	No dependency	Local time source |TE| ≤ 100ns (PRTC-A), ≤ 40ns (PRTC-B) O-RU |TE| ≤ 30ns	IEEE 802.1CM Category B
		local time source |TE| ≤ 100ns O-RU |TE| ≤ 1.4us including any holdover budget	Absolute |TAE| at O-RU antenna = 1.5μs derived from 3GPP (Same as IEEE 802.1CM Category C)
Absolute |TE| ≤ 100ns or 40 ns (see NOTE 1) • @ PRTC/T-GM UNI • PRTC/T-GM spec per G.8272 Not including holdover budget	**Network |TE| is out of scope** • O-DU may get its synchronization from either local or remote PRTC	**O-DU absolute |TAE|** @ O-DU UNI **< ±1.5μs (NOTE 5)**	Absolute |TAE| at O-DU = 1.5μs (Same as IEEE 802.1CM Category C)

NOTE 1: local PRTC (typically GNSS receiver) to O-RU. Therefore, budget is based on ITU-T G.8272 (PRTC class A, with max |TE| = 100ns, or PRTC class B, with max |TE| = 40ns) on each GNSS receiver. Because there is no requirement in ITU-T G.8272 on the relative time error between two neighbor local PRTC (GNSS receivers), it is not possible to remove any common time error in the budget at the O-RUs side.

NOTE 2: When feature for relative |TAE| is needed, LLS-C4 configuration will include 2x local PRTC-A |TE| = 200ns total, or 2x local PRTC-B |TE| = 80ns total. This disadvantage automatically prevents LLS-C4 configuration from meeting certain demanding relative |TAE| feature (as target by 802.1CM Category A). Hence, only 802.1CM Category B/C are shown.

NOTE 3: O-RU |TE| is not governed by eCPRI/802.1CM or ITU-T standard since LLS-C4 configuration is not based on network timing solution. O-RU |TE| budget is basically the remaining budget to satisfy target feature |TAE| after excluding the local PRTC |TE| contribution. O-RU |TE| includes the O-RU clock recovery (i.e. deriving a clean clock from local time source) error and any O-RU internal error.

NOTE 4: This O-RU option requires extra interface and extra hardware support including local PRTC (typically GNSS receiver and antenna) and likely a more expensive oscillator for noise filtering. Standard O-RU with network timing support (target for configuration LLS-C1, LLS-C2 and LLS-C3) cannot offer this option in general. Specific O-RU design is needed.

NOTE 5: TAE between O-DU and O-RU limit (as specified in clause 4.3) translates into absolute |TAE| for both O-DU and O-RU with half the budget. This is measured at any available O-DU output signal (either 1PPS or O-DU UNI).

The synchronization accuracy for the OTDOA feature is not defined by 3GPP. Hence absolute |TAE| at O-RU antenna and the corresponding O-RU|TE| are out of scope in O-RAN specification and is therefore not covered in this analysis. In a deployed network, operator can choose a target absolute |TAE| and then derive the corresponding O-RU |TE|.

11.4 Node Behavior Guidelines

11.4.1 Configurations LLS-C1 and LLS-C2

This clause covers O-RAN topology configurations LLS-C1 and LLS-C2 where the O-DU acts as PLFS and PTP master.

The operation of O-DU and O-RU during holdover and other related states is described in **Table 11-6**. O-RU holdover and O-DU holdover are independent events. Likewise, O-RU holdover behavior is optional (not mandatory to be supported by HW or SW).

In addition to synchronization state, the O-DU also considers estimated synchronization accuracy, because synchronization state alone does not necessarily reflect synchronization status – a node in the LOCKED or HOLDOVER mode may have synchronization accuracy outside of a required limit.

Table 11-6 : Node behavior during Holdover and Out-of-Synch

O-DU State	Synch accuracy	O-RU State	M-Plane State	Action
-	-	-	Disconnected	Clause 11.4.1.1
-	N/A	FREERUN	Connected	Clause 11.4.1.2
FREERUN	-	-	Connected	Clause 11.4.1.3
LOCKED/ HOLDOVER	In a limit	LOCKED/ HOLDOVER	Connected	Clause 11.4.1.4
LOCKED/ HOLDOVER	Out of a limit	LOCKED/ HOLDOVER	Connected	Clause 11.4.1.4

11.4.1.1 M-Plane disconnected

O-RAN allows hybrid M-plane model with M-plane communication between

- O-RU and O-DU
- O-RU and Service Management and Orchestration (SMO) function

As a result, the following M-plane disconnected events shall be considered:

- O-DU detects loss communication to O-RU
- SMO detects loss communication to O-RU
- O-RU detects loss communication to O-DU
- O-RU detects loss communication to SMO
- O-RU detects loss communication to both O-DU and SMO

NOTE: The following behavior is an assumption and expected to be described in M-Plane specification.

O-DU

If the O-DU detects a loss of M-plane communication to an O-RU, the O-DU shall stop sending any IQ data towards the O-RU. The O-DU shall also send an explicit command to the O-RU to disable RF transmission.

The O-DU shall keep RF transmission on the O-RU off, and shall not turn it on, if M-plane communication to the O-RU is broken.

Rationale: The requirement for the O-DU to send an explicit command is intended to prevent unsupervised radio operation, if M-plane communication broken in one direction only, and the fault stays undetected on the O-RU. On the other hand, if the O-RU detects the fault, it disables RF transmission autonomously (see below), and the O-DU shall stop IQ transmission accordingly.

O-RU

If the O-RU detects a loss of M-plane communication to the O-DU or both O-DU and SMO, the O-RU stops RF transmission. The O-RU shall keep RF transmission off, and shall not turn it on, if M-plane communication to the O-DU or SMO is broken.

If the O-RU detects a loss of M-plane communication to the SMO only, the O-RU action shall continue RF transmission. It is expected SMO can eventually detect a loss of M-plane communication to the O-RU based on some round-trip sanity check mechanism. Any SMO action is out of the scope of this specification.

SMO

If the SMO detects a loss of M-plane communication to an O-DU or an O-RU, the SMO action is out of scope of this specification.

11.4.1.2 O-RU in the FREERUN State

O-DU

If synchronization state on a connected O-RU transits to the FREERUN state, the O-DU shall stop sending C-Plane and U-Plane related data to the O-RU unless otherwise specified. When O-RU transits to LOCKED state, O-DU shall request carriers to be switched to ACTIVE to reenable transmission.

Rationale: The O-DU receives a notification that the O-RU switched to the FREERUN state and, as consequence, all configured carriers on the O-RU were disabled. The O-DU shall disable carriers to be aligned on the carrier configuration. After O-RU reestablish synchronization, the O-RU notifies the O-DU that the O-RU switched to the LOCKED state and then if the O-DU intends to reenable transmission, the O-DU shall set carriers to active.

NOTE: For carrier activation procedure and notifications related to carriers and synchronization state, please refer to the M-Plane specification [7].

O-RU

If synchronization state on an O-RU transits to the FREERUN state, the O-RU shall autonomously stop RF transmission and switch all carriers to INACTIVE state. The O-RU shall send a notification to the O-DU about synchronization and carriers state change. The O-RU shall re-enable RF transmission only when the O-DU requests it.

Rationale: The O-RU is obliged to stop RF transmission as soon as the accuracy of the signal can no longer be guaranteed. The O-RU shall send notification to the O-DU about any changes in its states. The O-RU shall not activate carriers by itself as it is not guaranteed that the O-DU is providing valid U-Plane and C-Plane related data to the O-RU.

11.4.1.3 O-DU in the FREERUN state

O-DU

If an O-DU transits to the FREERUN state, the O-DU shall disable RF transmission on all connected O-RUs, and keep it turned off until synchronization is reacquired.

NOTE: The O-DU shall support configuration option that allows O-DU to operate outside of the required synchronization limits, or without any synchronization at all.

O-RU

The O-RU shall only react on a change of Quality Level, received in PLFS SSMs, and Clock Class, received in PTP Announce messages:

- If the received Quality Level and Clock Class are acceptable the O-RU shall continue using the reference signal.
- If the received Quality Level or Clock Class becomes unacceptable the O-RU shall stop using the reference and transit to either the FREERUN state or a HOLDOVER state.
- Otherwise, the O-RU shall rely on O-DU to take care of the changed synchronization state. The O-RU need not react on the FREERUN state at the O-DU in any distinct way.

11.4.1.4 Operation in LOCKED/HOLDOVER state

Whether in “LOCKED” or “Holdover” state, it is expected that O-DU monitors the “LOCKED/HOLDOVER” status of the O-RUs under its management.

O-DU

- In configuration LLS-C1 and LLS-C2: by collecting the O-RUs' "LOCKED" or "HOLDOVER" state, as well as the received PLFS and PTP quality status , O-DU in "LOCKED" or "HOLDOVER" state is able to detect any self-estimated frequency and/or time accuracy degradation by the O-RUs.
  NOTE: all O-RUs in the LOCKED state and directly connected to the same master clock (typically the O-DU in LLS-C1, and the nearest T-BC in LLS-C2) preserve optimal relative time error between them, regardless the O-DU's status. As a result, intra-site features may remain enabled using different criteria compared to inter-site ones.
- NOTE: the port-to-port constant and dynamic time errors between two master ports of the same module (either the O-DU in configuration LLS-C1, and the T-BC ones in LLS-C2) may also contribute to the total error. This is currently not specified in ITU-T G.826x and ITU-T G.827x recommendations and shall be taken into account in the upcoming editions.

O-RU

The O-RU shall only react on a change of Quality Level received in PLFS SSMs, and Clock Class received in PTP Announce messages:

- If received Quality Level and Clock Class are acceptable, the O-RU shall keep on using the reference signal.
- If received Quality Level or Clock Class become unacceptable, the O-RU shall stop using the reference and transit to either the FREERUN state or a HOLDOVER state.
  ITU-T G.8271.1 network limits define the notion of "within holdover specification" (clock class values 7 and 135), and "exceeding the holdover specification" (clock class values 140, 150, 160, 165).
  ITU-T G.8271.1 defines this holdover specification as 400ns in the context of category C (as per IEEE 802.1CM). This value is however too high and useless for categories A/B, highlighting that the notion of "acceptable" is dependent on the category of each feature.
  As a consequence, it is mandatory that each O-RU reports to O-DU the received Quality Level and Clock Class, allowing O-DU to enable and disable accordingly the various RF features.
  Besides, the value for the holdover specification may be configurable, thus allowing each network operator to tune it to its own needs.

11.4.2 Configurations LLS-C3

This clause covers O-RAN topology configuration LLS-C3 based on IEEE 802.1CM bridging network. PRTC/GM is provided by the fronthaul network.

The operation of the Fronthaul network elements, O-DU and O-RU during holdover and other related states is described in **Table 11-7**. O-RU holdover and O-DU holdover are independent events. Likewise, O-RU holdover behavior is optional (not mandatory to be supported by HW or SW).

More than one PRTC/GM may be considered as a deployment option to improve redundancy. Should a PRTC/GM fail, then another should be available as a backup time source and the PTP network tree would automatically re-arrange. Only a short holdover (NOTE: duration needs to be defined from ITU-T G.8271.1) shall be supported inside the various network elements (as well as O-DU and O-RU) to provide a safe operation during this rearrangement scenario.
NOTE: the O-DU can also be configured to provide backup PLFS+PTP like in LLS-C2.

In addition to synchronization state, the O-DU also considers estimated synchronization accuracy because the synchronization state alone does not necessarily reflect synchronization status; a node in the LOCKED or HOLDOVER mode may have synchronization accuracy outside of a required limit.

Table 11-7 : Node behavior during Holdover and Out-of-Sync

Network State	Sync Accuracy	O-DU and O-RU State	M-Plane State	Action
-	-	-	Disconnected	Clause 11.4.2.1
-	N/A	FREERUN	Connected	clause 11. 4.2.2 and 11. 4.2.3
LOCKED/ HOLDOVER	In limit	LOCKED/ HOLDOVER	Connected	Clause 11. 4.2.4
LOCKED/ HOLDOVER	Out of limit	LOCKED/ HOLDOVER	Connected	Clause 11. 4.2.4

11.4.2.1 M-Plane disconnected

This clause is same as 11.4.1.1

11.4.2.2 O-RU in the FREERUN State

This clause is same as 11.4.1.2

11.4.2.3 O-DU in the FREERUN state

NOTES:

- if O-DU and O-RU are synchronized from the same fronthaul network and are connected to neighbor nodes in this network, it is most probable that the event leading to O-RU transiting to the FREERUN state will also lead to the same transition at the O-DU.

- if O-DU has backup frequency and time source, such as local or remote PRTC, it can become a backup Master like in configuration LLS-C2.

O-DU

If an O-DU transits to the FREERUN state, because the synchronizing network delivers unacceptable synchronization quality, the O-DU shall disable RF transmission on all connected O-RUs, and keep it turned off until synchronization is reacquired again.

NOTE: The O-DU shall support configuration option that allows O-DU to operate outside of the required synchronization limits, or without any synchronization at all.

O-RU

The O-RU is not synchronized from the O-DU. It may have no indication of the O-DU synchronization status, and therefore shall only rely on O-DU to take care of the changed synchronization state. The O-RU need not react to the FREERUN state at the O-DU in any distinct way.

11.4.2.4 Operation in SYNCED/HOLDOVER state

Whether in “LOCKED” or “HOLDOVER” state, it is expected that O-DU monitors the “LOCKED” or “HOLDOVER” state, as well as the received PLFS and PTP quality status. This is same as configurations LLS-C1 and LLS-C2 described in earlier clause.

11.4.3 Configurations LLS-C4

This clause covers O-RAN topology configurations LLS-C4 where the O-RU is synchronized by local PRTC (typically a GNSS receiver).

The operation of O-DU and O-RU during holdover and other related states is described in

Table 11-8**.** O-RU holdover and O-DU holdover are independent events. Likewise, O-RU holdover behavior is optional (not mandatory to be supported by HW or SW).

In addition to the synchronization state, the O-DU also considers estimated synchronization accuracy because the synchronization state alone does not necessarily reflect synchronization status; a node in the LOCKED or HOLDOVER mode may have synchronization accuracy outside of a required limit.

Table 11-8 : Node behavior during Holdover and Out-of-Sync

O-DU State	Sync Accuracy	O-RU State	M-Plane State	Action
-	-	-	Disconnected	Clause 11.4.3.1
-	N/A	FREERUN	Connected	Clause 11.4.3.2
FREERUN	-	-	Connected	Clause 11.4.3.3
LOCKED/ HOLDOVER	In limit	LOCKED/ HOLDOVER	Connected	Clause 11.4.3.4
LOCKED/ HOLDOVER	Out of limit	LOCKED/ HOLDOVER	Connected	Clause 11.4.3.4

11.4.3.1 M-Plane disconnected

This clause is same as 11.4.1.1

11.4.3.2 O-RU in the FREERUN State

This clause is same as 11.4.1.2

11.4.3.3 O-DU in the FREERUN state

O-DU

If an O-DU transits to the FREERUN state, the O-DU shall disable RF transmission on all connected O-RUs and keep it turned off until synchronization is reacquired again.

NOTE: The O-DU may support a configuration option that allows O-DU to operate outside of the required synchronization limits, or without any synchronization at all.

O-RU

The O-RU is not synchronized from the O-DU. It may have no indication of the O-DU synchronization status, and therefore shall only rely on O-DU to take care of the changed synchronization state. The O-RU need not react to the FREERUN state at the O-DU in any distinct way.

11.4.3.4 Operation in SYNCED/HOLDOVER state

Whether in “LOCKED” or “HOLDOVER” state, it is expected that O-DU monitors the “LOCKED” or “HOLDOVER” state. This is same as configurations LLS-C1 and LLS-C2 described in earlier clause.

11.5 S-Plane Handling in Multiple Link Scenarios

Behavior of S-Plane in scenarios with multiple links shall be based on the following principles:

O-DU - Grand Master (configurations LLS-C1 & LLS-C2)

There shall be an input sync reference signal on at least one link to an O-RU. Likewise, it is not prohibited to have input reference signal on multiple or all links to a given O-RU.

O-RU (all configurations)– Subordinates

O-RU shall be able to adapt at startup to the reception of a sync reference signal on any port from their master. How the ingress signal is detected (usually SSM for SyncE and Announce for PTP), and how the active port is selected (e.g., via round-robin, ITU-T G.8275.1 BMCA, etc.) are implementation-specific.

If the input reference is present on multiple links, the O-RU may, but is not required to, implement redundancy for the input reference signal if only capacity links are present on the module.

11.6 Announce Messages

O-RU shall check the following advertised parameters against a list of acceptable values based on its own design (assumed to be M-Plane configurable):

- Domain Number: Default: 24 (for Full Timing Support per ITU-T G.8275.1) or 44 (for Partial Timing Support per ITU-T G.8275.2)
- PTP Acceptable Clock Classes:
    - Default: 6, 7
    - Operator configurable: 6, 7, 135, 140, 150, 160, 248

11.7 Elementary Procedures

11.7.1 PTP Time Synchronization procedure

All procedures used to exchange time related information between a time synchronization master and subordinate shall be compliant to the ITU-T G.8275.1 or G.8275.2 telecom profile, which provides necessary details on utilization of the IEEE 1588 protocol in telecom applications.

11.7.2 System Frame Number Calculation from GPS Time

The general framework for System Frame Number (SFN) calculation from GPS (or GNSS) time is based on the following premises:

- PTP time on the fronthaul interface shall use PTP timescale
- The PTP epoch is 1 January 1970 00:00:00 TAI, which is 31 December 1969 23:59:51.999918 UTC.
- PTP time on the fronthaul interface shall be traceable to a PRTC if a network wide synchronization of O-RUs at the air interface is required (as in TDD 5G)
- From PTP time, the GPS seconds elapsed since GPS epoch (midnight January 6th, 1980) can be calculated, since the difference between PTP and GPS epoch is a constant. The GPS seconds are expressed as a real number.
- GPS seconds shall be used to calculate the frame number according to:

$$FrameNumber = floor(\frac{GPSseconds - \beta * 0.01 - \frac{\alpha}{1.2288 * 10^9}}{framePeriodinSeconds})mod(maxFrameNumber + 1)$$

    - number expressed in [seconds]
    - framePeriodinSeconds = 0.01 [seconds];
    - maxFrameNumber = 1023
    - NOTE: α and β are defined as follows:

Table 11-9 : α and β parameter exchange

Direction	Parameter	Range	Resolution
O-DU to O-RU	Offset α (NR) of radio frame timing (NOTE 1)	α : 0~1.2288e7 α * (1/1.2288 ns) : 0.00s~0.01s	1/1.2288 ns
O-DU to O-RU	Offset β (NR) of radio frame timing (NOTE 1)	β: -32768~32767 β * (10ms) : -327680ms~327670ms	10ms
NOTE 1: Parameter data types and values for α and β are provided by the O-RAN M-plane spec. Epoch for α and β (i.e. SFN=0) is set to 1980.1.6 00:00 (UTC)			

Figure 11-5 below shows examples when α and β are set to different values. The frame number at the antenna reference point is shown in relation to the $\text{GPS}_{\text{seconds}}$ time. For simplicity the GPS-time around 10 seconds is shown.

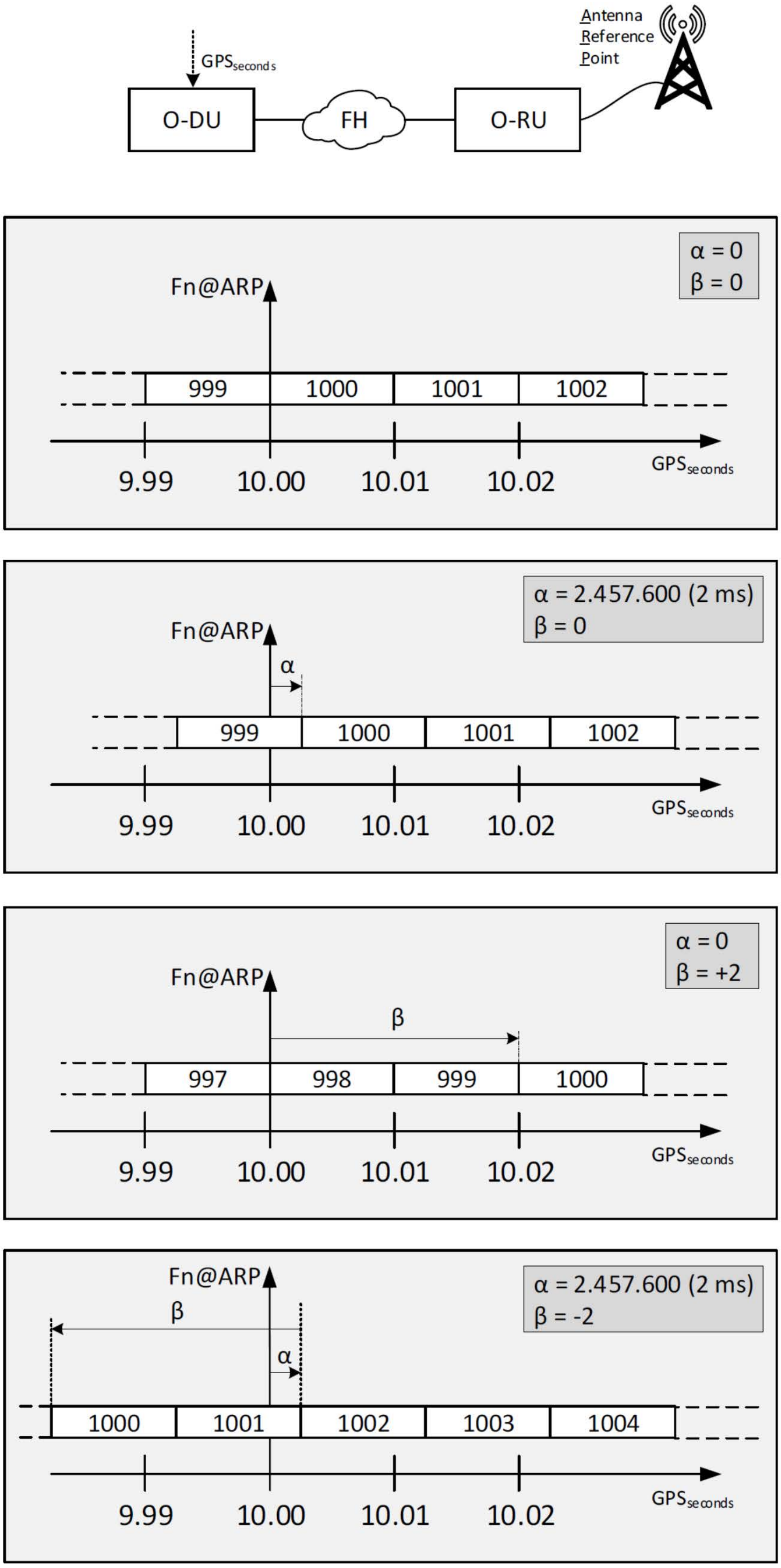


Figure 11-5 : α and β parameter example

The UL to DL radio frame timing offset in TDD systems is described in **Table 9-10** with default values applicable for 5G NR FR1 and FR2 ranges. $N_{TAoffset}$ is the configurable parameter used to control the timing offset. The absolute time offset $T_{TAoffset}$ can be calculated as $N_{TAoffset}$ *1/1.96608GHz.

Table 11-10 : Definition of $T_{TAoffset}$ & $N_{TAoffset}$

Frequency Range	Range	Resolution	Default
FR1	$N_{TAoffset}$: 0~65535 $T_{TAoffset}$: 0µs~33.3µs (NOTE 3)	0.5ns	$N_{TAoffset}$: 25600 (NOTE 2) $T_{TAoffset}$: 25600*Tc= 13.02µs (NOTE 1)
FR2	$N_{TAoffset}$: 0~65535 $T_{TAoffset}$: 0µs~33.3µs (NOTE 3)	0.5ns	$N_{TAoffset}$: 13792 (NOTE 2) $T_{TAoffset}$: 13792*Tc= 7.015µs (NOTE 1)
NOTE 1: Tc=~0.5ns=1/1.96608GHz NOTE 2: based on 3GPP TS38.133 Table 7.1.2-2 NOTE 3: $T_{TAoffset}$=0s for FDD systems			

12 Beamforming Functionality

12.1 General

The following clauses describe terminologies, rules, properties and uses cases related to beamforming and its functionalities. It is the baseline to follow by the O-DU, O-RU and modeling in M-plane.

12.2 Hierarchy of Radiation Structure in O-RU

The hierarchy of radiation structure in O-RU is depicted in Figure 12-1 and described below:

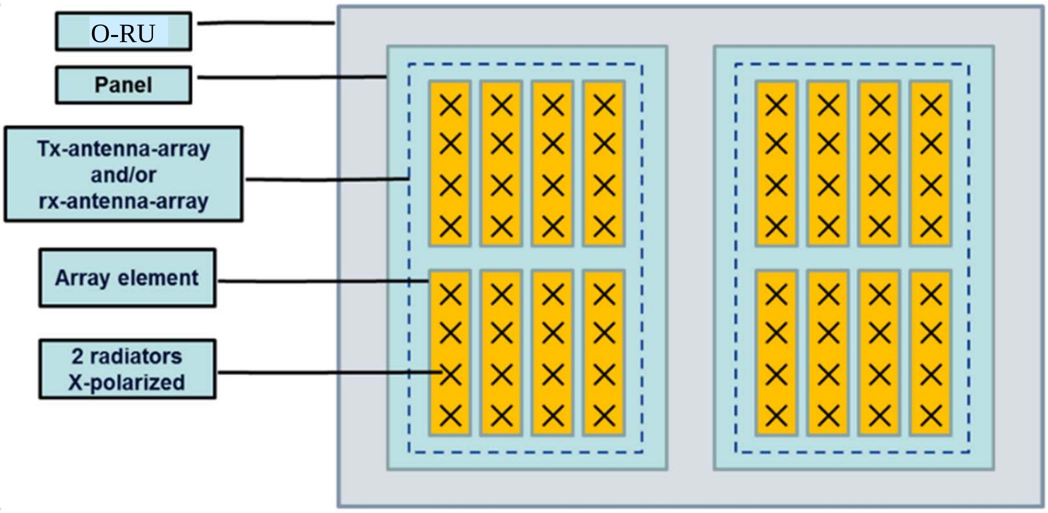


Figure 12-1. Hierarchy of Radiation Structure

- **O-RU:** each O-RU can have 1 or several Panels
- **Panel:** each panel can have 1 or several TX-antenna-arrays/RX-antenna-arrays
- **TX-antenna-array/RX-antenna-array:**
  - TX-antenna-array/RX-antenna-array is a logical construct used for data routing; there is a relationship to physical antennas as defined by the O-RU construction, and eAxC values may be assigned to TX-antenna-arrays/RX-antenna-arrays:
    - Multiple eAxC values may be assigned to a single TX-antenna-array or RX-antenna-array
    - An eAxC value may only be assigned to a single TX-antenna-array and RX-antenna-array
  - Each TX-antenna-array/RX-antenna-array has only 1 polarization for Category A O-RUs.
  - Each TX-antenna-array/Rx-antenna-array may include both polarizations for Category B O-RUs.
    - O-RUs shall at least provide for RX-antenna-arrays for each polarization
  - Each TX-antenna-array/RX-antenna-array terminates 1 or several RU_port_IDs.
  - Each TX-antenna-array/RX-antenna-array can have 1 or several array elements (one use case being antenna technologies wherein a single “array element” can support a high degree of beamforming, another use case being the need to address a single element in the array).
  - Each physical array element may be considered as a member of one or multiple TX-antenna-arrays / RX-antenna-arrays.
  - Amplitude and phase of each array element inside one TX-antenna-array/RX-antenna-array can be controlled via the weights pointed to by the beam_ID(s) which are received via the terminated RU_port_ID(s). Alternatively, the beam_IDs may indicate the “state” of an antenna array not explicitly comprising separate radiating elements which is also possible within the O-RAN specification.

  - It is assumed that when executing beamforming, array elements within an array are combined in phase to form directed energy, therefore the array elements belonging to a TX-antenna-array and RX-antenna-array are calibrated together as a group (although such a calibration operation may be hidden from the interface (e.g. executed once upon O-RU manufacture).

- **Array element:** each array element can have 1 or several Radiators (where amplitude and phase relation between the radiators cannot be changed dynamically during real-time)

- **Radiator:** radiating element – see IEEE Std 145-1993 (R2004)

12.3 Calibration

Calibration is the functionality of eliminating/minimizing relative amplitude and phase differences over frequency domain or time delay over the time domain between the array elements belonging to same TX-antenna-array/RX-antenna-array (including effect of front-end analog filters).

Calibration can also be applied between two or many TX-antenna-arrays/RX-antenna-arrays if those TX-antenna-arrays/RX-antenna-arrays belong to the same calibration Group which is part of the O-RU capabilities.

NOTE: Calibration capability between different TX/RX antenna arrays is expected to be conveyed via M-Plane but is not supported in this version of the specification. When not available, it should be assumed that calibration is not possible between the TX/RX antenna arrays.

12.4 beamId Use for Various Beamforming Methods

There are two main domains in which beamforming is executed, frequency-domain and time-domain; it is also possible to combine both (called “hybrid beamforming”). Frequency-domain beamforming is done between the RE mapping and FFT/iFFT processing stages (in UL and DL respectively) so is inherently a digital operation. Time-domain beamforming may be executed digitally or in the analog domain.

A characteristic of frequency-domain beamforming when used with OFDM is that different users may use the same time slot yet use different beams. In contrast, with time-domain beamforming all the users and signals in a time slot use the same beam. Hybrid beamforming allows different users in the same time slot to use different beams (the frequency-domain part) at the same time as all the users using a shared time-domain beam. An example is the case where the time-domain beam provides directivity in the elevation plane (so all users use the same elevation beam) while the frequency-domain beams provide directivity in the azimuth plane (so different users may use different azimuth beams).

When implementing any kind of beamforming, O-RAN supports the following beamforming methods (see also Annex J for a more mathematical description of the following beamforming methods):

a) Predefined-beam beamforming
b) Weight-based dynamic beamforming (based on real-time-updated weights)
c) Attribute-based dynamic beamforming (based on real-time-updated beam attributes)
d) Channel-information-based beamforming

When the Channel-information-based beamforming method is used for at least one layer (or spatial stream), then O-DU shall use same beamforming method for all layers (or spatial streams) of the specific time-frequency resource element(s). Similarly when the Attribute-based beamforming method is used for at least one layer (or spatial stream), then O-DU shall use same beamforming method for all layers (or spatial streams) of the specific time-frequency resource element(s).

A: Predefined-beam beamforming

In this case, an index called “beamId” indicates the specific beam pre-defined in the O-RU to use. The beamId could indicate a frequency-domain beam or a time-domain beam or a combination of both (“hybrid” beam) and the O-DU needs to know to ensure the beamId is properly applied e.g. the O-DU could not apply different time-domain beams to the different PRBs in the same OFDM symbol. The method the O-RU uses to generate the beam is otherwise not relevant, it could use the application of gain and phase controls on separate antenna elements, or use multiple shaped-energy antennas, or any other technology. The O-RU is expected to convey to the O-DU via the M-Plane on startup beam characteristics but the O-DU remains ignorant regarding how the beam is actually created by the O-RU.

B: Weight-based dynamic beamforming

Here the O-DU is meant to generate weights that create the beam so the O-DU needs to know the specific antenna characteristics of the O-RU including how many antenna elements are present in the vertical and horizontal directions and the antenna element spacing, among other properties. The weight vector associated with each beam has a beamId value and the interpretation of this beamId value is addressed in this clause.

C: Attribute-based dynamic beamforming

Like index-based beamforming, attribute-based beamforming allows the O-DU to tell the O-RU to use a specific beamId but in this case that beamId is associated with certain beam attributes as described in clause 7.4.7.2. How the O-RU achieves the implementation of the beams is not specified, however again the O-DU needs to know whether the beam identified by the beamId is generated as a frequency-domain beam or a time-domain beam to ensure the beamId is properly applied e.g. the O-DU could not apply different time-domain beams to the different PRBs in the same OFDM symbol.

D: Channel-information-based beamforming

In this case the O-DU provides channel information per UE periodically (generally less often than every slot) and then on a slot-by-slot basis the O-DU provides scheduling information which the O-RU uses along with the channel information to calculate the proper beamforming weights for the specific slot with its co-scheduled UEs. Here there is no beamId value associated with the beamforming, instead the ueID is provided associated with each data section. Therefore this clause regarding beamId usage is not relevant for this beamforming method.

12.4.1 Predefined-beam Beamforming

When implementing index-based beamforming, it is necessary for the O-RU to convey to the O-DU whether the beamforming type is frequency-domain, time-domain, or a mixture of the two ("hybrid beamforming"). In the case of frequency-domain-only or time-domain-only, the beamId is simply an index to the desired beamforming weight vector or other beamforming method. In the case of hybrid beamforming, there are present in the O-RU pre-loaded frequency-domain weight vectors and time-domain weight vectors (these are applied separately). The beamId points to a single combined frequency-domain and time-domain weight vector. However, in reality there will be the application of a frequency-domain beamforming weight vector and the separate application of multiple time-domain beamforming weight vectors, one per frequency-domain weight value.

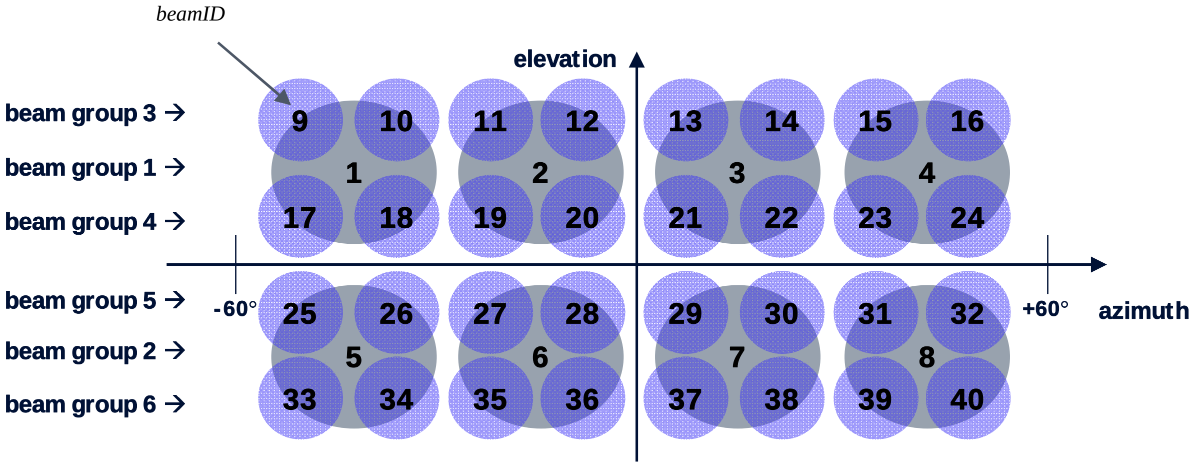
For this case the M-Plane shall convey from the O-RU to the O-DU upon start-up as part of the O-RU self-description:

- the list of available beamIds and their characteristics, see clause 12.4.1.1 below.

Note that there is no requirement that beamIds be in sequential order or that there are no gaps between beamId values. The only constraint is that the beamId value of zero is reserved so cannot refer to any frequency-domain, time-domain or hybrid beam.

12.4.1.1 Beam Characteristics

In order to use predefined-beam beamforming in a standardized way, O-RAN considers beamforming to be defined such that energy (in the DL) or sensitivity (in the UL) is focused into either a "coarse" or "fine" granularity with possible overlaps. In this way "broadcast" beams may be used to cover a wider area with less power or sensitivity, while higher-power or higher-sensitivity beams may be used in e.g. a per-UE fashion. Figure 12-2 shows an example of the assumed beam arrangement.



Note: ovals represent beams, numbers in ovals represent beamId

Figure 12-2 : Example of updatable-weight frequency-domain plus fixed time-domain beamforming

Figure 12-2 introduces the concept of "beam-group". A "beam-group" is the set of beamIds that may be used simultaneously for some reason. In Figure 12-2 an assumption can be made that the elevation direction is time-domain while the azimuth direction is frequency-domain; the frequency-domain beams in the same time-domain group may be used at the same time but different time-domain groups shall be separated in time.

Figure 12-2 also shows the need to describe beam overlap and beam adjacency. It is necessary, for example, to convey that beamId=1 is a coarse beam overlapping with fine beamIds={9,10,17,18} and that beamId=18 is a fine beam overlapping with coarse beamId=1. It is also necessary to convey that beamId=1 has as neighbors beamId={2,5,11,19,25,26} and that beamId=18 has as neighbors beamIds={2,5,10,17,19,26}. Overlapping beams shall not be scheduled together to avoid interference, and neighboring beams should not be scheduled together to avoid interference where possible.

The specific beam characteristics are therefore as follows, all per beamId:

1) beam-type for the subject beamId enumerated as COARSE-BEAM or FINE-BEAM;
    - there is no attempt to quantify "coarse" and fine" in terms of beamwidth, this is just a relative relationship
2) beam-group-id for the subject beamId as an integer;
3) coarse-fine-beam-relation as a list: if the subject beamId is coarse, this is a list of the associated fine beams, and if the subject beamId is fine, this is a list of the associated coarse beams;
4) neighbor-beam as a list: a list of all beams that may interfere with the subject beamId

An O-RU which supports predefined beamforming may also support shifting the coverage area upon M-plane command, referred to as the 'predefined-beam-tilt-offset' feature. The O-RU should convey its capability and associated parameters for the 'predefined-beam-tilt-offset' feature if supported. This feature allows an operator to adjust the coverage of the O-RU in elevation and/or azimuth angle. Figure 12-3 shows an example of the predefined beam-tilt-offset operation. As a default, a predefined beam has its own steering angle denoted by (elevation angle: "theta", azimuth angle: "phi"). If the O-RU controller configures the elevation-tilt-offset-angle as "+A" degrees and azimuth-tilt-offset-angle as "+B" degrees, then the O-RU shall regenerate the beams to change its steering angles to be elevation angle value: "theta + A", azimuth angle value: "phi + B". An M-plane command delivers the elevation and azimuth predefined beam-tilt-offset angles. To regenerate beams, O-RU may load one of the pre-stored multiple beam weight vectors corresponding to all supported predefined-beam tilt values or may regenerate beam weights upon receiving the M-plane command; this is up to the O-RU implementation.

Note, elevation-tilt-offset-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).

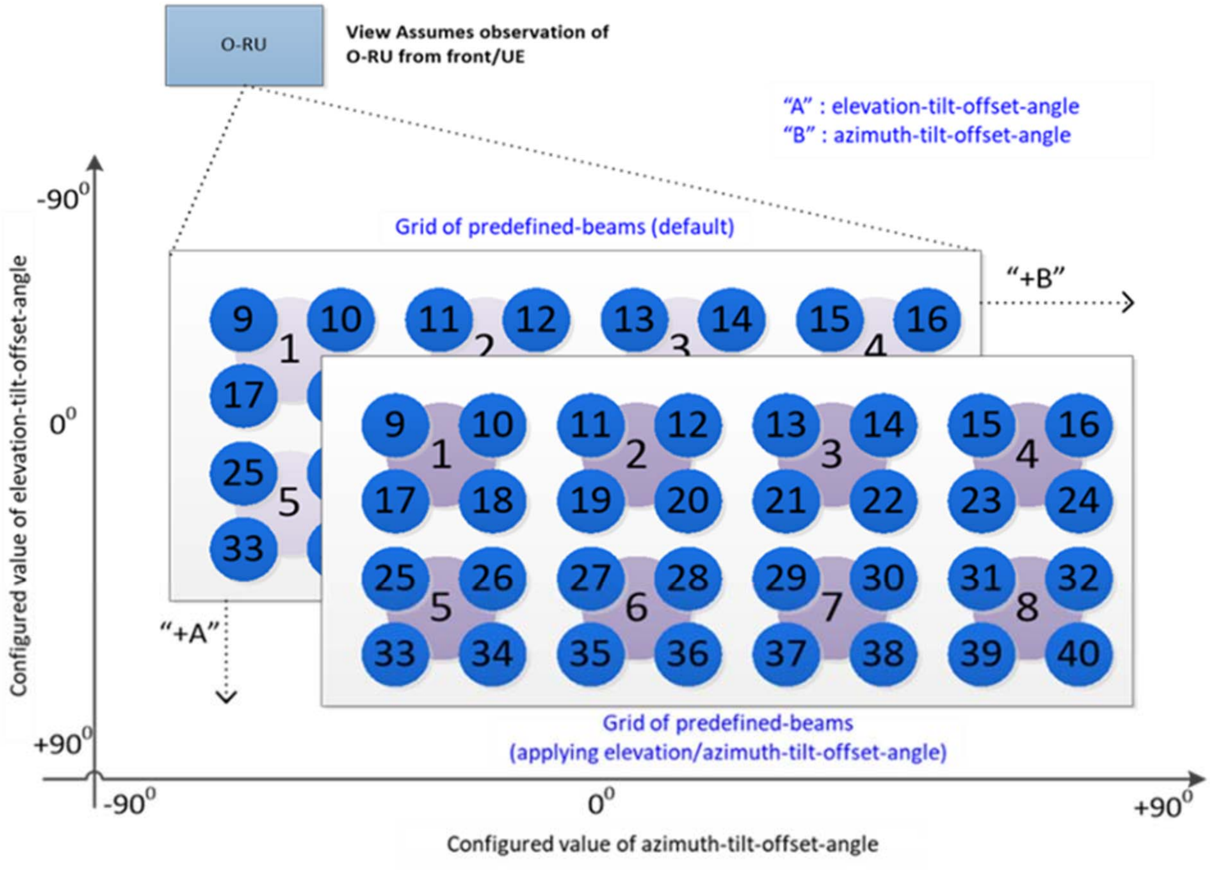

O-RU
View Assumes observation of O-RU from front/UE
"A" : elevation-tilt-offset-angle
"B" : azimuth-tilt-offset-angle
Grid of predefined-beams (default)
"+B"
"+A"
Grid of predefined-beams (applying elevation/azimuth-tilt-offset-angle)
-90⁰
0⁰
+90⁰
Configured value of elevation-tilt-offset-angle
Configured value of azimuth-tilt-offset-angle
9 10 11 12 13 14 15 16
1 2 3 4
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
5 6 7 8
33 34 35 36 37 38 39 40

Figure 12-3 : Example of shifting a grid of predefined beams with tilt-offset

12.4.2 Weight-based dynamic beamforming

Real-time-updated-weight-based beamforming operates the same as index-based beamforming, except that the need for the O-DU to convey actual beam weights to the O-RU introduces additional complexity.

12.4.2.1 Weight-based dynamic frequency-domain or time-domain beamforming (not hybrid)

In the case of either frequency-domain or time-domain beamforming wherein the beamforming weights can be updated in real-time and have a beamId value associated with the weights, the beamId is treated the same: it points to a set of weights that control the array elements' gain and phase and the number of weights equals the number of array elements. In many cases, the magnitude of each complex weight value will equal unity but this is not required; in particular "tapering" may require less-than-unity weight magnitudes for some array elements. The weight values prior to any compression will be fractional hence no I or Q value may exceed positive or negative unity.

The following list describes the information that the M-Plane shall carry from the O-RU to the O-DU upon start-up as part of the O-RU self-description, the information listed is per array (so per tx-array or per rx-array):

1) Beamforming type, enumerated as "frequency", "time" or "hybrid"
2) Maximum number of weight-based beamId values supported (could be zero) : "numBeams"
    - O-RUs may have memory limitations that mean the number of beams is limited; zero means no weight-based beamforming is supported by this tx-array or rx-array
3) Initial value of weight-based beamId supported: "initBeamId"
    - Different ranges of beamId may support weight-based beamforming versus e.g. predefined beamforming

4) Frequency granularity of time-domain beamforming, enumerated as “per component carrier” or “per band”.
    - Value is only present for time-domain beamforming
5) Time granularity of time-domain beamforming, enumerated as “per-OFDM-symbol” or “per-slot”.
    - Value is only present for time-domain beamforming

Because the beams are to be generated by the O-DU the O-RU will not know the beam characteristics so they are not reported.

The actual number of weights K in the frequency-domain or time-domain weight vectors will be clear from the O-RU antenna model, see the Clause 12.5 on that topic.

12.4.2.2 Weight-based dynamic hybrid beamforming

Here two sub-cases are considered, wherein for one sub-case both the frequency-domain and time-domain weights may be updated in real-time, and for the second sub-case the frequency-domain weights may be updated in real-time but the time-domain beams are fixed.

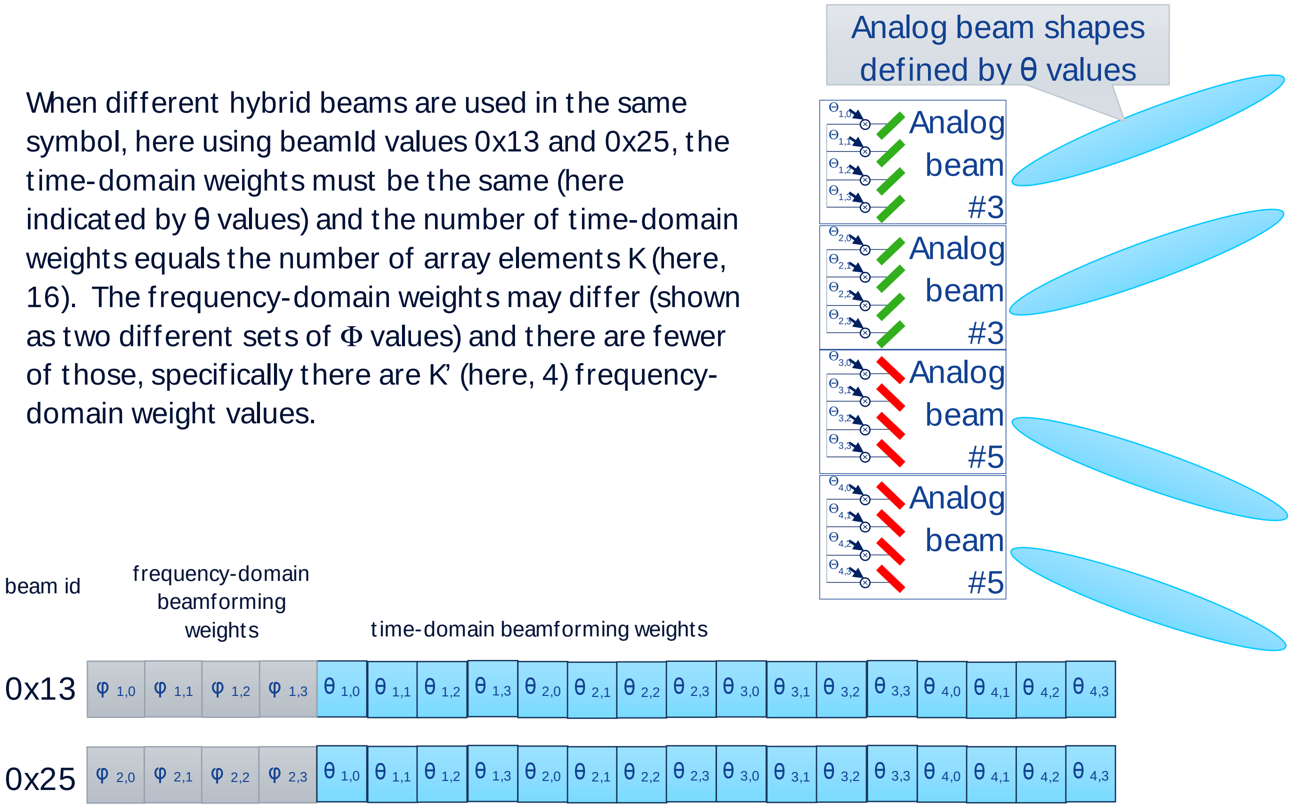
12.4.2.2.1 Hybrid beamforming with updatable frequency-domain and time-domain weights

For this sub-case the beamforming weight vector is a composite of the frequency-domain weights and the time-domain weights so can be considered as simply a longer weight vector. Where a block-based beam weight compression is employed (block floating point, block scaling or µ-law compression), the block size is a single beamforming weight vector (both frequency-domain and time-domain parts). The actual number of weights in the composite frequency-domain plus time-domain weight vectors (K’ + K) will be clear from the O-RU antenna model, see the Clause 12.5 on that topic.

The following list describes the information that the M-Plane shall carry from the O-RU to the O-DU upon start-up as part of the O-RU self-description, the information listed is per array (so per tx-array or per rx-array):

1) Beamforming type, enumerated as “frequency”, “time” or “hybrid” – here will be “hybrid”
2) Maximum number of weight-based beamId values supported (could be zero) : “numBeams”
    - O-RUs may have memory limitations that mean the number of beams is limited; zero means no weight-based beamforming is supported by this tx-array or rx-array
3) Initial value of weight-based beamId supported: “initBeamId”
    - Different ranges of beamId may support weight-based beamforming versus e.g. predefined beamforming
4) Frequency granularity of time-domain beamforming, enumerated as “per component carrier” or “per band”.
5) Time granularity of time-domain beamforming, enumerated as “per-OFDM-symbol” or “per-slot”.

Note that the number of time-domain beam weights associated with a given beamId is the same as the number of array elements which is K, but the number of frequency-domain weights is less, being K’. p’ represents the dimensionality of the time-domain beamforming operation, so that K = K’ * p’. The total length of the beamforming weight vector, including both the K’ frequency-domain weights and the K time-domain weights, is K’ + K = K’ + (K’ * p’) = K’ * (p’+1). Figure 12-4 shows an example where K = 16, K’ = 4 and p’ = 4, and the length of the beamforming weight vector (frequency-domain and time-domain combined) is 20 complex weights.



When different hybrid beams are used in the same symbol, here using beamId values 0x13 and 0x25, the time-domain weights must be the same (here indicated by θ values) and the number of time-domain weights equals the number of array elements K (here, 16). The frequency-domain weights may differ (shown as two different sets of Φ values) and there are fewer of those, specifically there are K' (here, 4) frequency-domain weight values.

beam id	frequency-domain beamforming weights				time-domain beamforming weights															
0x13	$\varphi_{1,0}$	$\varphi_{1,1}$	$\varphi_{1,2}$	$\varphi_{1,3}$	$\theta_{1,0}$	$\theta_{1,1}$	$\theta_{1,2}$	$\theta_{1,3}$	$\theta_{2,0}$	$\theta_{2,1}$	$\theta_{2,2}$	$\theta_{2,3}$	$\theta_{3,0}$	$\theta_{3,1}$	$\theta_{3,2}$	$\theta_{3,3}$	$\theta_{4,0}$	$\theta_{4,1}$	$\theta_{4,2}$	$\theta_{4,3}$
0x25	$\varphi_{2,0}$	$\varphi_{2,1}$	$\varphi_{2,2}$	$\varphi_{2,3}$	$\theta_{1,0}$	$\theta_{1,1}$	$\theta_{1,2}$	$\theta_{1,3}$	$\theta_{2,0}$	$\theta_{2,1}$	$\theta_{2,2}$	$\theta_{2,3}$	$\theta_{3,0}$	$\theta_{3,1}$	$\theta_{3,2}$	$\theta_{3,3}$	$\theta_{4,0}$	$\theta_{4,1}$	$\theta_{4,2}$	$\theta_{4,3}$

Figure 12-4 : Example of updatable-weight frequency-domain and time-domain beamforming

12.4.2.2.2 Hybrid beamforming with updatable frequency-domain weights and fixed time-domain beams

For this sub-case the beamforming weight vector is a composite of the frequency-domain weights and the time-domain beam numbers with the frequency-domain weights in the first half of the vector and the time-domain beam numbers in the second half of the vector. This vector shall not be considered as simply a longer weight vector because the frequency-domain weights may be compressed but the time-domain beam numbers shall not be compressed. Where a block-based beam weight compression is employed (block floating point, block scaling or μ-law compression), the block size is a single beamforming weight vector but only that half of the weight vector containing the frequency-domain weights. The remaining half of the vector containing the time-domain beam numbers may not be compressed and contains the integer time-domain beam numbers.

The actual number of weights in the composite frequency-domain weights plus time-domain beam-number vectors (K' and K) will be clear from the O-RU antenna model (see the Clause 12.5 on that topic) with the number of frequency-domain weights K' indicating which elements in the vector are subject to compression (the first K' complex values in the vector).

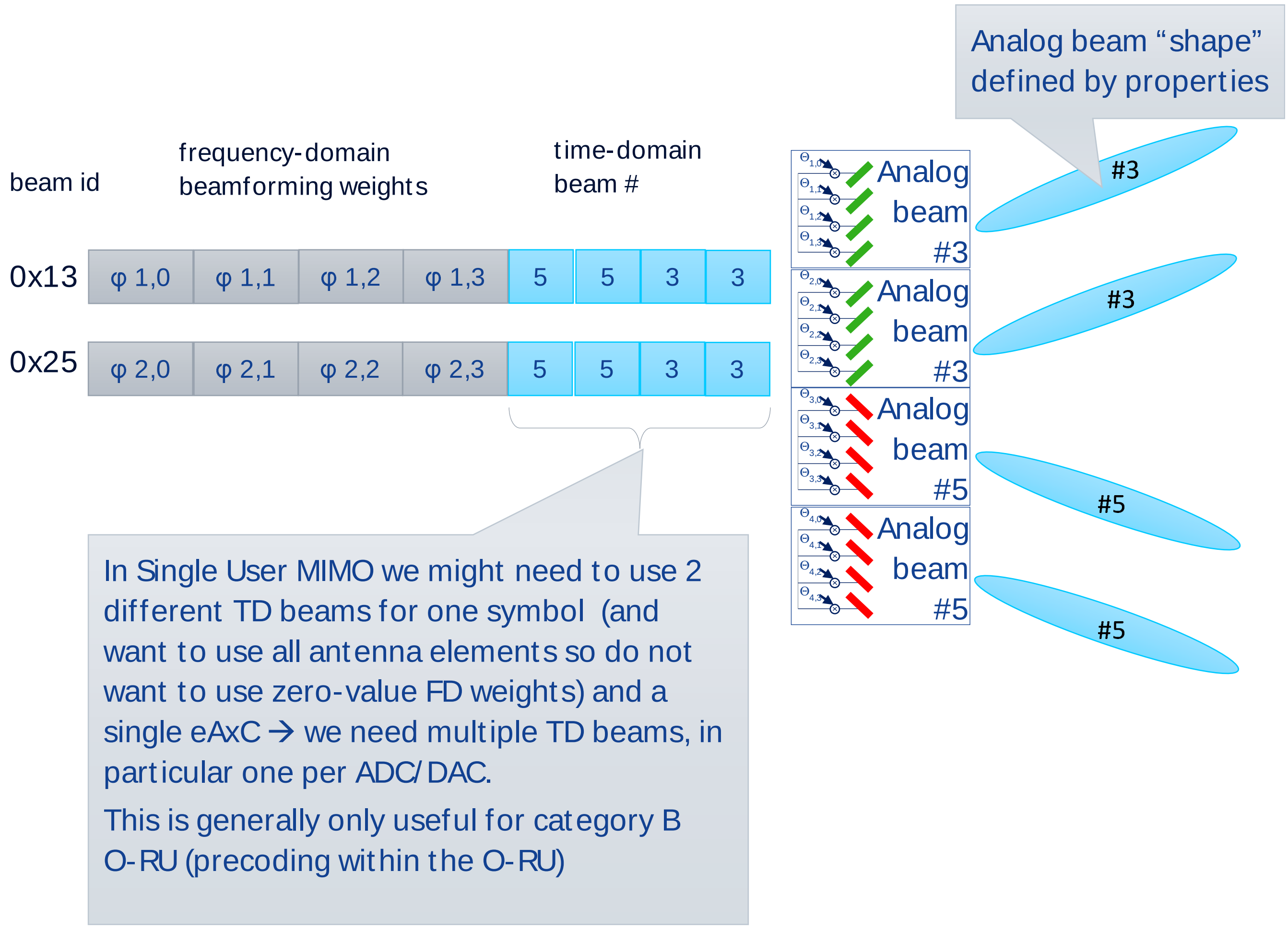
The following list describes the information that the M-Plane shall carry from the O-RU to the O-DU upon start-up as part of the O-RU self-description, the information listed is per array (so per tx-array or per rx-array):

1) Beamforming type, enumerated as “frequency”, “time” or “hybrid” – here will be “hybrid”
2) Maximum number of weight-based beamId values supported (could be zero) : “numBeams”
    a. O-RUs may have memory limitations that mean the number of beams is limited; zero means no weight-based beamforming is supported by this tx-array or rx-array
3) Initial value of weight-based beamId supported: “initBeamId”
    a. Different ranges of beamId may support weight-based beamforming versus e.g. predefined beamforming
4) Frequency granularity of time-domain beamforming, enumerated as “per component carrier” or “per band”.
    a. Value is only present for time-domain beamforming
5) Time granularity of time-domain beamforming, enumerated as “per-OFDM-symbol” or “per-slot”.

a. Value is only present for time-domain beamforming

6) For each of the time-domain beams (the number is known from the O-RU antenna model), beam attributes (see 11.3.1.1)

Note that the number of time-domain beam numbers associated with a given beamId (K’) will be the same as the number of frequency-domain beam weights for that beamId; this is because each frequency-domain beamforming weight is applied to a data stream that is subsequently time-domain beamformed using a specific beam number, so if there are e.g. K’=4 frequency-domain weights associated with a given beamId there will be four time-domain beam numbers also associated with that same beamId. See Figure 12-5 for an example wherein four frequency-domain weights $\Phi_{x,y}$ are applied with four time-domain beamforming numbers (#3 and #5). Here the number of frequency-domain weights K’ indicated by the antenna model would be “four” so the first four values in each vector would be complex fractional values and would be compressed, while the second four values would be real integers and not compressed.



beam id	frequency-domain beamforming weights				time-domain beam #			
0x13	φ 1,0	φ 1,1	φ 1,2	φ 1,3	5	5	3	3
0x25	φ 2,0	φ 2,1	φ 2,2	φ 2,3	5	5	3	3

Analog beam “shape” defined by properties

In Single User MIMO we might need to use 2 different TD beams for one symbol (and want to use all antenna elements so do not want to use zero-value FD weights) and a single eAxC → we need multiple TD beams, in particular one per ADC/DAC.

This is generally only useful for category B O-RU (precoding within the O-RU)

Figure 12-5 : Example of updatable-weight frequency-domain plus fixed time-domain beamforming

12.4.3 Attribute-based dynamic beamforming

Attribute-based dynamic beamforming operates similarly to weight-based dynamic beamforming except that it is inherently a time-domain beamforming operation (both are “dynamic” meaning the definition of a beam as indicated by a beamID value may be changed via a C-Plane message). Also, instead of beamforming weights associated with a specific beamId being sent from O-DU to O-RU in Section Extension =1, beam attributes associated with a specific beamId are sent instead from the O-DU to the O-RU in a different Section Extension =2.

The following list describes the information that the M-Plane shall carry from the O-RU to the O-DU upon start-up as part of the O-RU self-description:

1) Beamforming type, enumerated as “frequency” or “time” (not “hybrid”)
2) Maximum number of beamId values supported (could be zero) : “numBeams”
    a. O-RUs may have memory limitations that mean the number of beams is limited; zero means no weight-based dynamic beamforming is supported by this tx-array or rx-array
3) Initial value of beamId supported: “initBeamId”
    a. Different ranges of beamId may support generated beam beamforming versus e.g. predefined beamforming
4) Valid range of bfAzPt (see clause 7.4.7.2.2)
5) Valid range of bfZePt (see clause 7.4.7.2.3)
6) Valid range of bfAz3dd (see clause 7.4.7.2.4)
7) Valid range of bfZe3dd (see clause 7.4.7.2.5)
8) Valid range of bfAzSl (see clause 7.4.7.2.6)
9) Valid range of bfZeSl (see clause 7.4.7.2.7)

12.4.4 Channel-information-based beamforming

As stated earlier, beamId is irrelevant and unused in the case of channel-information-based beamforming.

12.5 O-RU Antenna Model supported by O-RAN

Knowledge of O-RU antenna model is critical for certain types of beamforming. The following model is applicable for O-RU with one or more antennas, where each antenna has array of elements that are

- uniform (all elements have same properties) and
- organized into rectangular array (with rows and columns) that is planar (flat).

O-RU exposes via M-plane logical model of O-RU consisting of one or more arrays compose of one or more array elements. Array element represents independently controllable entity including one or more radiating elements and related RF processing elements (Here, RF processing element is an entity that processes RF signal and is not related to processing element defined in M-plane). Note RX and TX are in general independently controllable for that in the model TX and RX arrays are described as separate entities. If O-RU supports beamforming, then beamforming is realized within each array separately i.e. beamforming weight vector is applicable within one array. One or more arrays can occupy same physical location e.g. RX array and TX array that use same set of radiators.

Beamforming methods that use dynamic beamforming with beamforming weights conveyed in C-plane message (in contrast to predefined beams) require the O-DU to know antenna properties. Different beamforming methods require knowledge of different subsets of antenna properties.

12.5.1 Coordinate Systems

Some of parameters describing model of antenna related to coordinate system that defines three axes and three angles.

There are two coordinate systems defined:

- array coordinate system
- O-RU coordinate system

The array coordinate system is presented below:

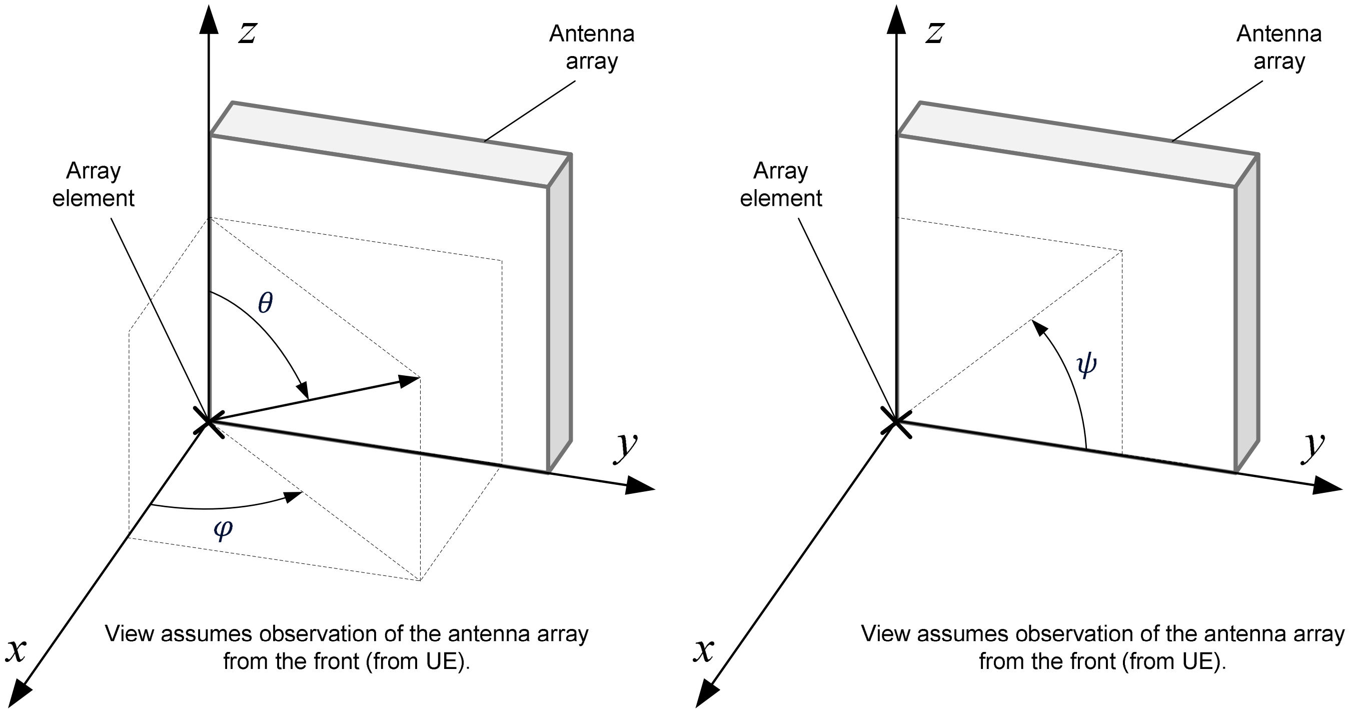

z
Antenna array
Array element
θ
y
φ
x
View assumes observation of the antenna array from the front (from UE).
z
Antenna array
Array element
ψ
y
x
View assumes observation of the antenna array from the front (from UE).

Figure 12-6 : Array Coordinate System

The diagram presents view from the front of array-panel (from UE). Arrows indicate increasing values of coordinates.

x points towards broad-side
y increases to right, with antenna-array's columns
z points towards zenith
φ (phi) is azimuth angle, counter-clockwise rotation around z-axis, 0° points to broad-side, 90° points to y-axis
θ (theta) is zenith angle, counter-clockwise rotation around y-axis, 0° points to zenith, 90° points to horizon
ψ (psi) is angle, counter-clockwise rotation around x-axis, 0° points to horizon, 90° points to zenith

The **array coordinate system** is centered on centre of the leftmost, bottom element of array. The array coordinate system is applicable within one array.

The **O-RU coordinate system** is the selected array coordinate system of an O-RU. The selection is fixed by O-RU design. The O-RU coordinate system is applicable within one O-RU.

12.5.2 O-RU Antenna Model Parameters

The O-RU antenna model can be described with following parameters:

- **K** – number of array elements in array (note that K = M*N*P*Q)
- **M** – number of rows of array elements in array. M>0; value 0 is reserved for future use.
- **N** – number of columns of array elements in array. N>0; value 0 is reserved for future use.
- **P** – number of polarizations in array. P>0, value 0 is reserved for future use.
- **Q** – number of overlapping array elements (array-layers) in array. Each array-layer has M rows, N columns and P polarizations. Q>0; value 0 is reserved for future use. See Annex K for more information on array-layers.
- **x**, **y**, **z** – position of centre of the leftmost, bottom element of array in O-RU coordinate system
- **φ**, **θ** direction of normal vector perpendicular to array's surface in O-RU coordinate system (array's normal vector corresponds to x axis in array's coordinate system)
- **dy** – mean distance between centres of nearby array elements in y direction in array coordinate system (distance between columns); value 0 is reserved for future use.
- **dz** – mean distance between centres of nearby array elements in z direction in array coordinate system (distance between rows); value 0 is reserved for future use.

- **list of polarizations in array** (this list has P elements, each representing p-th polarization); values ordered in ascending order of angle. Example: cross-polarized array having elements of one of two linear polarizations can be described by list: (-45°, +45°) indicating that array element with polarization index p=0 has linear polarization -45°, and array element with polarization index p=1 has linear polarization +45°.
- **independent power budget** per layer - in case of an array with multiple layers, corresponding elements (located in same row and column and same polarization) of different layers may have a shared power budget or have independent power budgets.

For an array supporting hybrid beamforming (see clause 12.4.2) there is a need for additional parameters:

- **K'** – number of frequency domain beamforming weights $\varphi_{k'}$ that can be applied within the array.
  0<K'≤K; value 0 is reserved for array not supporting hybrid beamforming.

- **h**(k) – mapping of array element k to frequency domain beamforming weight $\varphi_{k'}$ where k'=h(k).
  The mapping is represented as a list of lists: for every 0<k'<K' a list of K/K' numbers identifying array elements where frequency domain beamforming weight $\varphi_{k'}$ is applied. k'=h(k) if number k is in the list corresponding to k'.
  Clause 12.5.312.5.3 Identification and Ordering of Array Elements describes how numbers are assigned to array elements.

The model assumes the number of array elements corresponding to frequency domain beamforming weight $\varphi_{k'}$ is the same for every k' (0<k'<K') and the elements corresponding to beamforming weights form a rectangular shape without overlapping i.e. every array element is linked with exactly one frequency domain beamforming weight $\varphi_{k'}$.

In addition, the O-RU antenna model provides parameters describing key capabilities of array elements. The model assumes the array is uniform and all elements have the same properties. Each single value is applicable to all elements within the array.

Parameters describing array elements applicable to TX and RX arrays:

- horizonal plane half power (-3 dB) beam width of array element's radiation pattern
- vertical plane half power (-3 dB) beam width of array element's radiation pattern
- horizonal plane quarter power (-6 dB) beam width of array element's radiation pattern
- vertical plane quarter power (-6 dB) beam width of array element's radiation pattern

Beam widths above are angles (expressed in degrees) between half-power (-3 dB) points or quarter-power (-6 dB) points respectively of the main lobe with reference to peak radiated power of main lobe. Horizontal and vertical plane correspond to the xy-plane and xz-plane respectively of the array in the array coordinate system.

The parameter describing array elements specific for TX array:

- $\boldsymbol{m}_{a,k}$ - max rms power rating of array element of the array. Usage of max rms power rating is described in clause 8.1.3.3.

12.5.3 Identification and Ordering of Array Elements

In many applications there is a need to assign to array element a number k such that 0≤k<K. One example is mapping position (represented by k, such that 0≤k<K) of beamforming weight in beamforming vector to array element. Other example is identification of array elements in antenna model.

For purpose of identification and ordering a number k is assigned to each element of array by the function f(m,n,p.q):

$$f(m,n,p,q) = q \cdot P \cdot M \cdot N + p \cdot M \cdot N + m \cdot N + n$$

where:

m - row (bottom to top), 0≤m<M
n - column (left to right, view from the front of array), 0≤n<N
p - polarization index, 0≤p<P; polarization value of polarization index p is ψ_p
q – array-layer, 0≤q<Q

Note that for a rectangular array, the function f(m,n,p,q) can be inverted allowing to specify a "tuple" (m,n,p,q) of k-th element.

12.5.4 Relations Between Array Elements

Beamforming methods that use dynamic beamforming with beamforming weights conveyed in C-plane messages (in contrast to predefined beams) require the O-DU to know that specific elements of one array is co-located with elements of another array e.g. RX array and TX array that use same set of radiators. In addition, one or more TX arrays may share elements and parts of RF processing paths (e.g. a power amplifier) resulting in a shared power budget described by a maximum rms power rating.

If element k_a of array A and element k_b of array B are in same position (same physical row and column) then k_a and k_b are co-located.

If element k_a of array A and element k_b of array B are co-located and share a power budget, then k_a and k_b are shared.

Relation of co-location is symmetric: k_a and k_b are co-located if and only if k_b and k_a are co-located.

Relation of co-location is transitive: if k_a and k_b are co-located and k_b and k_c are co-located then k_a and k_c are co-located.

Relation of sharing implies co-location: if k_a and k_b are shared then k_a and k_b are co-located.

O-RU shall report via M-plane relations between array elements. O-RU shall avoid reporting redundant relations that can be derived from other relations by symmetric and transitive properties and implication of co-location relation by sharing relation. In addition, the O-RU shall provide a concise representation of the common case of two arrays that have all elements in relation (e.g. RX array of -45° polarization and corresponding RX array of +45° polarization).

12.5.5 Model Usage

The O-RU antenna model reported by the O-RU consists of RX arrays and TX arrays. RX arrays and TX arrays represent a capability for transmitting/receiving RF signal related to an eAxC and - if beamforming is supported by O-RU on given array - beamforming capability. In this clause examples are presented: red and green bars represent array elements of different polarizations, grey box represents physical device, white rectangles represent arrays reported by O-RU.

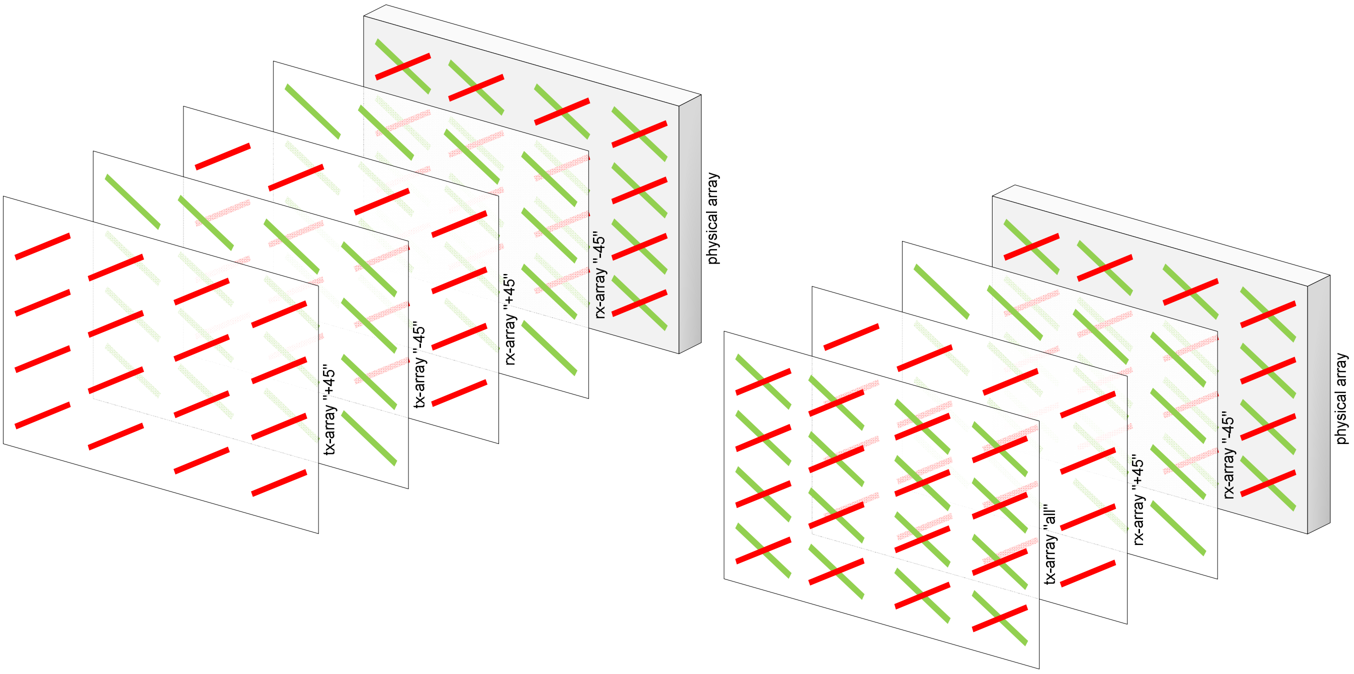


Figure 12-7 : Examples of Model Usage – TX as two single-polarization arrays or one cross-polarized array

As an example Figure 12-7 presents two O-RU designs: an O-RU with two TX arrays each of one polarization and an O-RU with one TX array of two polarizations (note number and dimension of TX array has an impact on the size of beamforming vectors). Of course, an O-RU that combines both above designs is possible as presented in Figure 12-8.

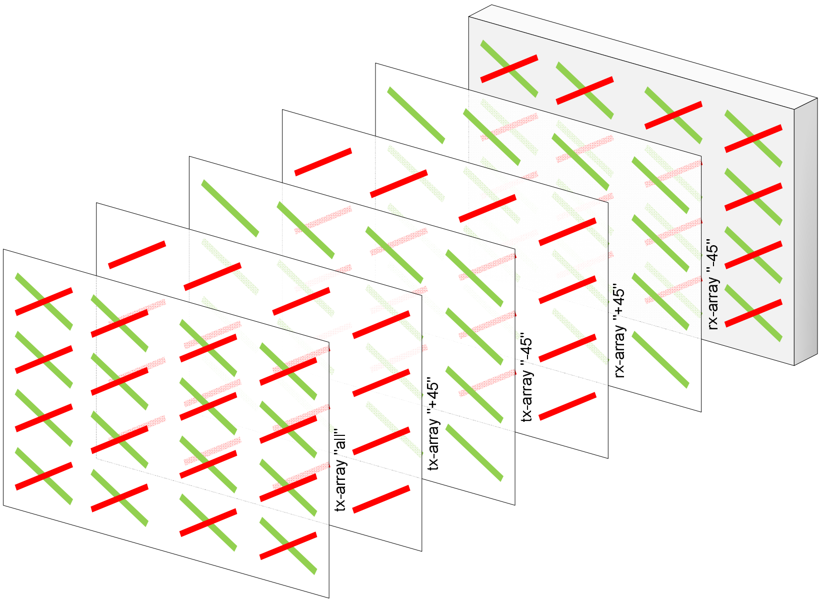


Figure 12-8 : Examples of Model Usage – TX as two single-polarization arrays and one cross-polarized array

Similarly, in RX – if an O-RU does not support the collecting raw SRS by use of beams designed to mute signal from all elements except one then that O-RU – in addition to full RX array – may expose RX arrays with single elements as presented in Figure 12-9.

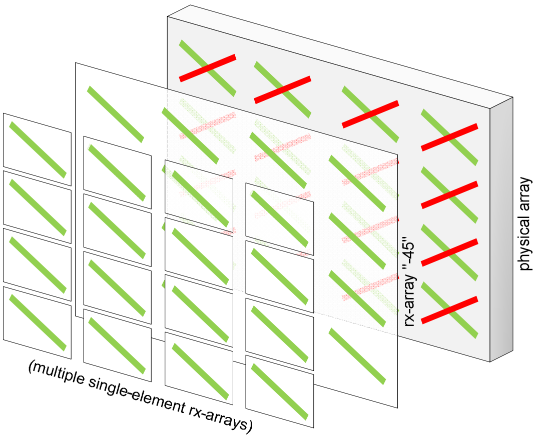


Figure 12-9 : Example of Model Usage – RX with multi-element and multiple single-element arrays

An O-RU that does not support beamforming can be represented with arrays having one element each. Note that a category B O-RU could be represented with an array with two elements to accommodate two polarizations while a category A O-RU would generally be represented with a TX array with only one polarization (P=1). Figure 12-10 presents the two design examples of a non-beamforming O-RU: an O-RU with two TX arrays each of one polarization and an O-RU with one TX array of two polarizations. Of course, an O-RU that combines both designs is also possible.

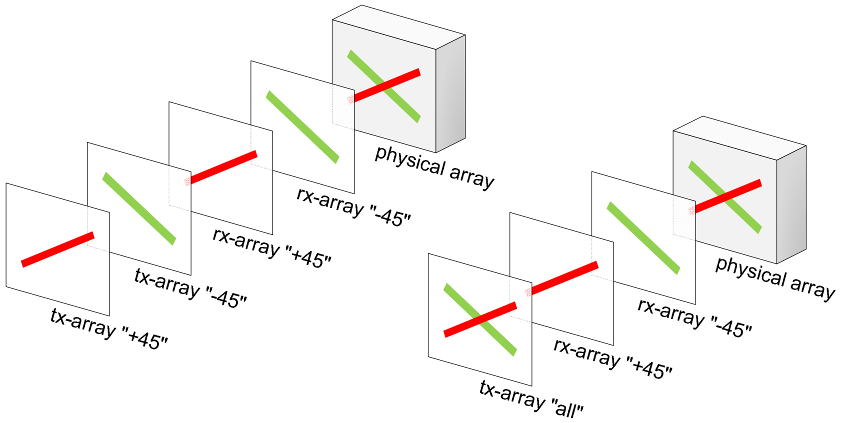


Figure 12-10 : Examples of Model Usage – Non-beamforming O-RU with TX as two single-polarization arrays or one cross-polarized array

13 Support of Shared Cell

13.1 General

This clause specifies support for “Shared Cell”. “Shared cell” is defined as the operation for the same cell (the cell can have one or multiple component carrier(s)) by several O-RUs.

There are 2 cases for realizing shared cell shown in Figure 13-1.

- **FHM mode:** Shared cell is realized by FHM and several O-RUs. In this case, FHM is placed between the O-DU and multiple O-RUs. FHM is modelled as an O-RU with LLS Fronthaul support (same as normal O-RU) and copy and combine function (additional to normal O-RU), but without radio transmission/reception capability. For the copy and combine function, FHM may support selective transmission and reception function. In this version, each O-RU under one FHM can be used for either operating the same cell (Single Cell Scenario at Figure11-1) or different cells (Multiple Cells Scenario at Figure11-1) by M-Plane configuration.
- **Cascade mode:** Shared cell is realized by several O-RUs cascaded in chain. In this case, one or more O-RU(s) are inserted between the O-DU and the O-RU. The O-RUs in the cascaded chain except for the last O-RU shall support Copy and Combine function. The O-RUs which support the Copy and Combine function are named “Cascade O-RU”. Note that the last O-RU may also support Copy and Combine function although it is not used (i.e. not only normal O-RU but also Cascade O-RU may work as the last O-RU). In this version, each O-RU in a cascaded chain is only used for operating the same cell.

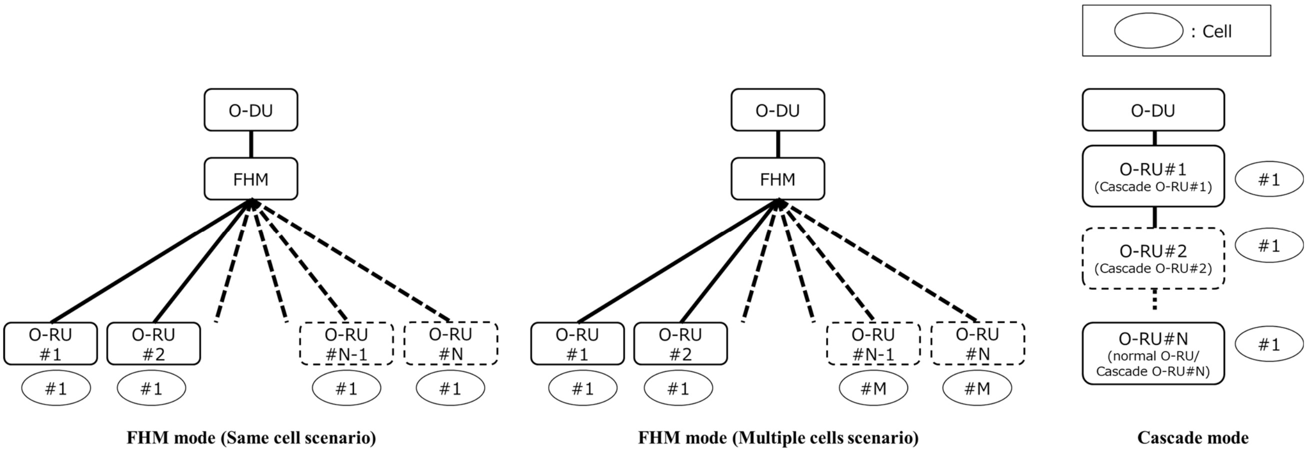


Figure 13-1 : Shared Cell Concept

For the cascade mode, the following generic reference to nodes in a cascaded chain is used. (For the FHM mode, the nodes are directly referred to as O-DU, FHM or O-RU, and the below terminologies are not used in this CUS-Plane specification. Please note that the terminologies: north-node and south-node are also used for FHM in the M-Plane specification to maximize commonality for M-Plane modelling.)

north-node: For a particular O-RU in the chain of cascaded O-RUs, the north-node is the adjacent O-RU which is located closer to the O-DU. In the case of the O-RU which is adjacent to the O-DU, the north-node is the O-DU.

south-node: For the O-DU or a particular O-RU in the chain of cascaded O-RUs, the south-node is the adjacent O-RU which is located farther from the O-DU.

In this clause, the following notations will be used;

N: the total number of O-RUs connected to a FHM for FHM mode, or the total number of O-RUs in a cascaded chain (including the last O-RU) for Cascade mode.

M: the total number of different cells realized by the total of *N* O-RUs connected to a FHM for FHM mode, or the total number of difference cells realized by the total *N* O-RUs in a cascaded chain for Cascade mode. *M* equals 1 for Same cell scenario for FHM mode. *M* equals 1 for Cascade mode in this version.

$\boldsymbol{N_m}$: the total number of O-RUs realizing a particular cell *m*. N_m equals *N* (and *m*=1) for Same cell scenario for FHM mode. N_m equals *N* (and *m*=1) for Cascade mode in this version.

n: used to denote a particular O-RU within a set of O-RUs realizing a particular cell.

13.2 Copy and Combine function

DL Copy function (shown in Figure 13-2**):**

In downlink case, FHM retrieves eCPRI messages coming from O-DU as payload of Ethernet frames, copies them (the entire eCPRI message including eCPRI header and eCPRI payload) without any modifications as payload into Ethernet frames and sends them towards the O-RUs realizing the shared cell. FHM determines these O-RUs from M-Plane configuration. Note that Figure 13-2 illustrates the Same cell scenario where copy is done for O-RU#1 to O-RU#N. For the Multiple cells scenario, for a particular cell *m*, copy will be for O-RU#1 to O-RU#N_m.

In downlink case, Cascade O-RU retrieves eCPRI messages coming from the north-node as payload of Ethernet frames, copies them (the entire eCPRI message including eCPRI header and eCPRI payload) without any modifications as payload into Ethernet frames and sends them towards the south-node.

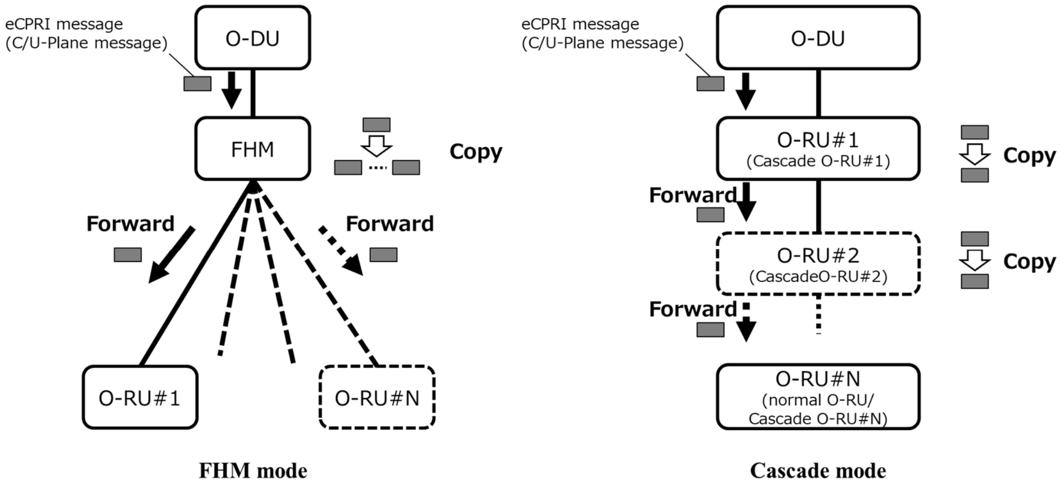


Figure 13-2 : Downlink flow for Shared Cell

UL Combine function (shown in Figure 13-3**):**

FHM combines IQ data corresponding to the same radio resource element from the multiple eCPRI messages for UL U-Plane in the Ethernet frames transmitted from each O-RU, where the general principles are listed below;

- The FHM identifies IQ data corresponding to the same radio resource element from the information elements in eCPRI transport header, application layer common header and application layer section field which are listed in Table 13-1**.** In case that transport layer fragmentation occurs, the FHM will need to re-assemble the fragmented message first.
- The FHM retrieves compression information (if present), iSample and qSample from those corresponding eCPRI messages and calculates the combined iSample and qSample by adding iSample and qSample individually, taking compression information into account as below;

  If U-Plane data compression is used then

  Combined iSample = ***Compress*** **(*Sum*** **(*Decompress*** **(*iSample_*#1), ... ,** ***Decompress*** **(*iSample_*#N_m))),**

Combined qSample = *Compress* (*Sum* (*Decompress* (*qSample_#*1), ... , *Decompress* (*qSample_#*N_m))),

Else

Combined iSample = *Sum* (*iSample_#*1, ... , *iSample_#*N_m),

Combined qSample = *Sum* (*qSample_#*1, ... , *qSample_#*N_m),

where

*iSample_#*n is the iSample received from the O-RU#n and
*qSample_#*n is the qSample received from the O-RU#n.

- If overflow occurs from the combine operation, iSample and/or qSample should be clamped to the closest value that can be represented in the compression format used for the combined IQ data.
- The compression format for the combined IQ data is the same as that in the IQ data received from the O-RUs.
- In this version of the specification, selective combining of UL signals is not supported.

After combining, the FHM generates eCPRI header (the ecpriSeqid field generated by the FHM is independent and not a copy of that received from O-RUs; other fields are just copies of those received from the O-RUs.), adds combined IQ data as eCPRI message body, constructs Ethernet frames carrying eCPRI messages as payload, and forwards them to the O-DU.

Note that Figure 13-3 illustrates the Same cell scenario where combine is done for O-RU#1 to O-RU#N. For the Multiple cells scenario, for a particular cell, combine will be for O-RU#1 to O-RU#N_m.

Cascade O-RU combines IQ data corresponding to the same radio resource element from air and from the eCPRI messages for UL U-Plane in the Ethernet frames transmitted from the south-node, where the general principles are listed below;

- The Cascade O-RU identifies IQ data corresponding to the same radio resource element from the information elements in eCPRI transport header, application layer common header and application layer section field which are listed in Table 11-1. In case that transport layer fragmentation occurs, the Cascade O-RU will need to re-assemble the fragmented message first.
- The Cascade O-RU retrieves compression information (if present), iSample and qSample from those corresponding eCPRI messages and calculates the combined iSample and qSample by adding iSample and qSample individually, taking compression information into account as below;

If U-Plane data compression is used then

Combined iSample = *Compress* (*Sum* (*Decompress* (*iSample_from_south-node*) and (*iSample_from air*))),

Combined qSample = *Compress* (*Sum* (*Decompress* (*qSample_from_south-node*) and *qSample_from air*)),

Else

Combined iSample = *Sum* (*iSample_from_south-node* and *iSample_from air*),

Combined qSample = *Sum* (*qSample_from_south-node* and *qSample_from air*),

where

*iSample_* ***from_south-node*** is the iSample received from the south-node,
*qSample_* ***from_south-node*** is the qSample received from the south-node,
*iSample_* ***from_air*** is the iSample received from the air, and
*qSample_* ***from_air*** is the qSample received from the air.

- If overflow occurs from the combine operation, iSample and/or qSample should be clamped to the closest value that can be represented in the compression format used for the combined IQ data.
- The compression format for the combined IQ data is the same as that in the IQ data received from the south-node.
- In this version of the specification, selective combining of UL signals is not supported.

After combining, the Cascade O-RU generates eCPRI header (the ecpriSeqid field generated by the Cascade O-RU is independent and not a copy of that received from the south-node; other fields are just copies of those received from the south-node.), adds combined IQ data as eCPRI message body, constructs Ethernet frames carrying eCPRI messages as payload, and forwards them to the north-node.

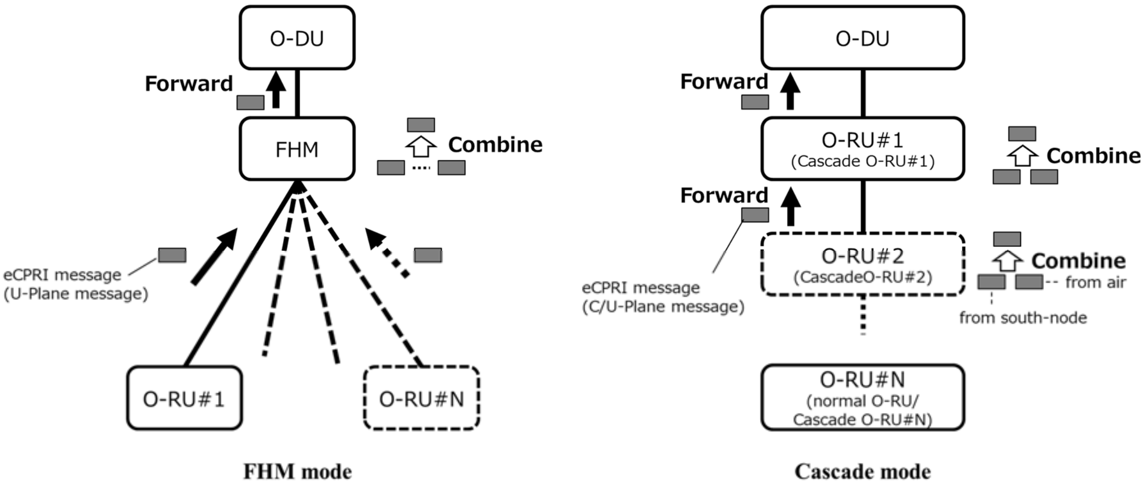


Figure 13-3 : Uplink flow for Shared Cell

Table 13-1 : information elements to be checked when FHM/Cascade O-RU combines UL U-Plane messages

Field	Information element
eCPRI transport header	ecpriPcid
Application layer common header	dataDirection, frameId, subframeId, slotId, symbolid
Application layer section field	rb, symInc, startPrbu, numPrbu

13.2.1 Selective transmission and reception using beamId

In order to avoid unnecessary interference and noise enhancement, selective transmission and reception function can be useful and it can be realized with minimum implementation impact on both O-DU/O-RU by using beamId.

The followings are noted for the selective transmission and reception function using beamId:

- Predefined-beam beamforming is assumed;
- Applicability for cascade mode (cascaded O-RUs or cascaded FHMs) is out of scope in this version due to the lack of study.

The concept is to utilize the multiple O-RUs involved in a shared cell collectively as a single beamforming O-RU, and to realize selective transmission and reception using beamforming operations.

- In normal beamforming operation, a unique SSB and/or CSI-RS are transmitted from each beam so that the O-DU scheduler can understand which beam should be used based on L1-RSRP reports from the UE; where each is further mapped to a unique beamId that the O-DU signals on the fronthaul.
- For shared cell with selective transmission and reception using beamId:
  - In the simplest form, all O-RUs involved can be non-beamforming O-RUs, and each of them can be regarded as separate beam of one collective beamforming O-RU. In this case, a unique SSB and/or CSI-RS are transmitted from each non-beamforming O-RU; where each is further mapped to a unique beamId that the O-DU signals on the fronthaul. Then the FHM, based on the beamId signalled from the O-DU, would route the messages to the appropriate non-beamforming O-RU. At this time, the FHM also translates the beamId so that the O-RU will transmit/receive using the appropriate beam. This assumes each O-RU handles unique user data meaning the FHM is not executing any copy/combining operation.

- Other forms can be considered. E.g.

  - All O-RUs involved can be still be non-beamforming O-RUs, but multiple O-RUs illuminating overlapping/neighbouring areas can be made to transmit a common SSB/CSI-RS (and common user data) and be regarded as a common beam from the O-DU perspective. The FHM, based on the beamId signalled from the O-DU which corresponds to the common beam, would route the messages to the appropriate set of multiple O-Rus and would execute a copy/combining operation for this data;

  - Multiple beamforming O-RUs can collectively be considered as a single beamforming O-RU. Each beam of the individual O-RUs may be regarded as a separate beam, or multiple beams illuminating overlapping/neighbouring areas can be made to transmit a common SSB/CSI-RS (and common user data) and be regarded as a common beam from the O-DU perspective. The FHM, based on the beamId signalled from the O-DU, would route the messages to the appropriate beamforming O-RU and would execute a copy/combining operation for this data, and also translate the beamId value so that the O-RU will transmit/receive using appropriate beams.

  NOTE: For the O-RU beams illuminating overlapping/neighbouring areas that are made to transmit a common SSB/CSI-RS and are regarded as a common beam from the O-DU perspective, coherency between the beams is not required.

From here on, the beamId used by the O-DU and signalled from the O-DU to FHM is referred to as the global beamId; and the beamId signalled from the FHM to O-RU and used by the O-RU is referred to as the local beamId.

Related M-plane aspects are described below.

- If O-DU receives the capability from FHM via M-Plane which indicates selective transmission and reception support, O-DU can configure the FHM to use selective transmission and reception function.

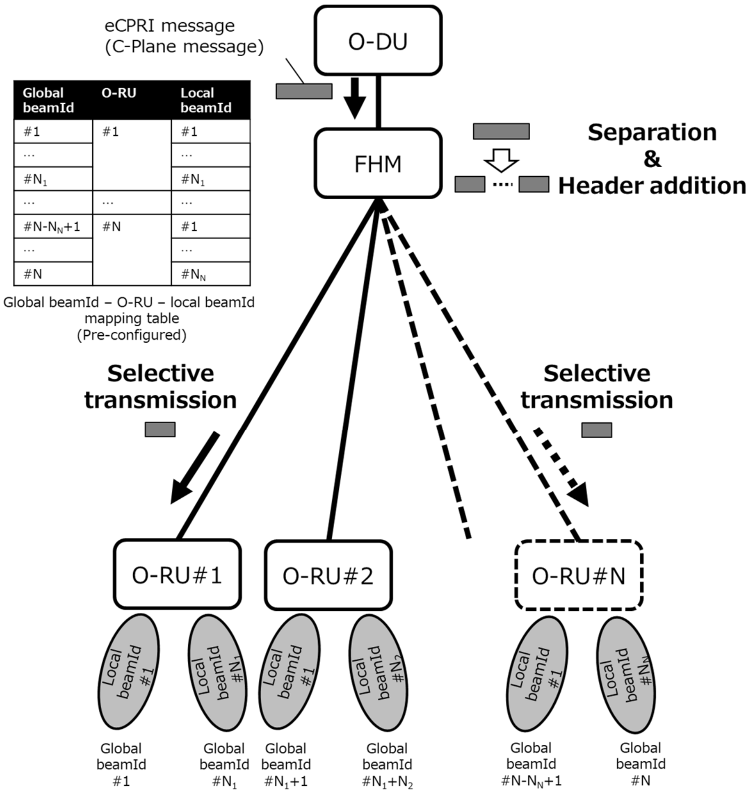
- The mapping information between each global beamId, O-RU(s) and their local beamId is configured to the FHM during the M-Plane start-up procedure.

- It is noted that the inter-beam relationship information such as coarse-fine-beam-relation and neighbour-beams are reported from each O-RU via M-Plane. Therefore, although O-DU considers the set of O-RUs as one beamforming O-RU, coarse-fine-beam-relation and neighbour-beams information across different O-RUs will not be available.

The details of copy combine function with selective transmission and reception function using beamId describted below.

Selective transmission function (shown in Figure 13-4**):**

In downlink case, FHM retrieves eCPRI messages coming from O-DU as payload of Ethernet frames, reads C/U-plane message-section field and separates C-Plane message section fields based on beamId and also separates U-Plane message section header fields/PRB fields based on sectionIds corresponding to each beamId. Then FHM creates new eCPRI messages by adding common header field on each new C/U-plane message which may includes multiple separated section fields for C-Plane and section header fields/PRB fields for U-Plane belonging to the same beam. The FHM sends them towards appropriate O-RUs according to the mapping table between beamId value and O-RU(s). For C-Plane messages, FHM needs to overwrite beamId field according to the mapping table between global beamId and local beamId. Note that Figure 13-4 illustrates the same cell scenario where selective transmission is done for O-RU#1 to O-RU#N. For the Multiple cells scenario, for a particular cell *m*, selective transmission will be for O-RU#1 to O-RU#N_m.



Global beamId	O-RU	Local beamId
#1	#1	#1
…		…
#N_1		#N_1
…	…	…
#N-N_N+1	#N	#1
…		…
#N		#N_N

C/U-Plane message selective transmission

Figure 13-4 : Downlink flow for selective transmission

Selective reception function (shown in Figure 13-5)**:**

FHM sends C-Plane messages as a same manner of selective transmission, i.e. the FHM sends C-Plane messages to RUs selectively based on the beamId value in the C-plane message received from O-DU and the mapping table between global beamId value and O-RU; and overwrites beamId value according to the mapping table between global beamId value and local beamId value for the O-RU.

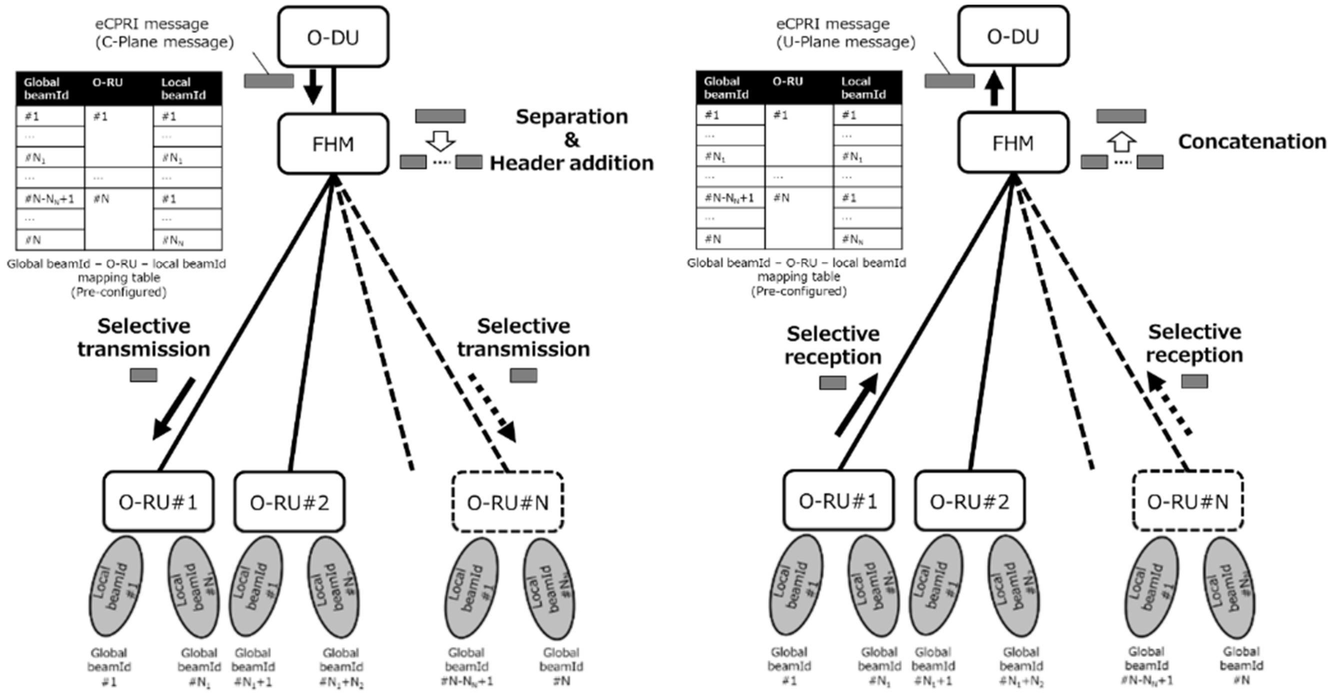
Then corresponding O-RU(s) transmits UL U-Plane messages which includes UL signal received by using the beam indicated by the beamId value in the C-Plane message received from FHM.

FHM combines IQ data according to 11.2. In case UL U-plane message received from different O-RUs do not include IQ data for the same radio resource element, combined IQ data is equivalent to IQ data received from particular O-RU.

FHM reconstructs eCPRI payload carrying UL U-Plane messages to be sent to the O-DU by concatenating combined IQ data and corresponding section header field with one UL U-Plane message common header.

After eCPRI payload reconstruction, the FHM generates eCPRI header (the ecpriSeqid field generated by the FHM is independent and not a copy of that received from O-RUs; other fields are just copies of those received from the O-RUs.), constructs Ethernet frames carrying eCPRI messages as payload, and forwards them to the O-DU.

Note that Figure 13-5 illustrates the same cell scenario where selective reception is done for O-RU#1 to O-RU#N. For the Multiple cells scenario, for a particular cell, selective reception will be for O-RU#1 to O-RU#N_m.



C-Plane message selective transmission
(same as downlink case)

U-Plane message selective reception

Figure 13-5 : Uplink flow for selective reception

Beam mapping parameters for selective transmission and reception using beamId are summarized in Table 11-2.

Table 13-2 : Beam mapping parameters for selective transmission and reception using beamId

Parameter	Description	Note
global beamId	This parameter indicates the beam ID to be applied to the U-Plane data on O-DU as same as existing beamId. One global beamId can be mapped to one or multiple local beamId. The mapping information between each global beamId, O-RU(s), and local beamId is configured to the FHM during M-Plane start-up procedure.	Since FHM and O-RUs are treated as a beamforming O-RU, global-beam-id = 0 is prohibited. Value range is same as that of existing beamId explained in 5.4.5.9.
local beamId	This parameter indicates the beam ID to be applied to the U-Plane data on O-RU as same as existing beamId. In one O-RU, different beams can not map to one global beamId. Local beamId shall be unique within O-RU.	

13.3 Delay management for Shared cell

Total fronthaul distance between O-DU and O-RU shall be reduced compared to the case when there are no FHM/Cascade O-RU. This is required in order to keep the same total fronthaul delay between O-DU and O-RU even in the presence of processing time at FHM/Cascade O-RU, and to ensure UL messages arrive at O-DU within O-DU reception window as well as DL messages arrive at O-RU within O-RU reception window. Further the transmission/reception timing of FHM/Cascade O-RU needs to be clarified. So, on top of clause 4.3, additional consideration is required for delay management which involves shared cell.

NOTE: Whether the delay management for shared cell in this clause can be used for measured transport method and/or dynamic timing advance is FFS.

13.3.1 DL delay management for Shared cell

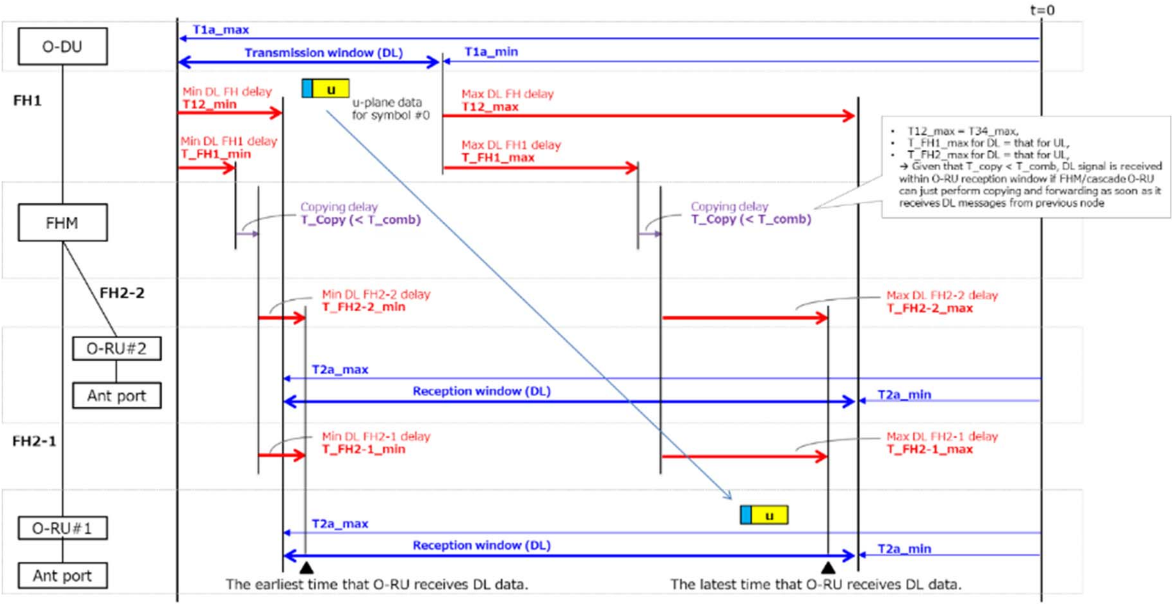


Figure 13-6 : Delay model parameters for FHM mode (2 O-RUs case, i.e. N_{ru} = 2, is illustrated as an example)

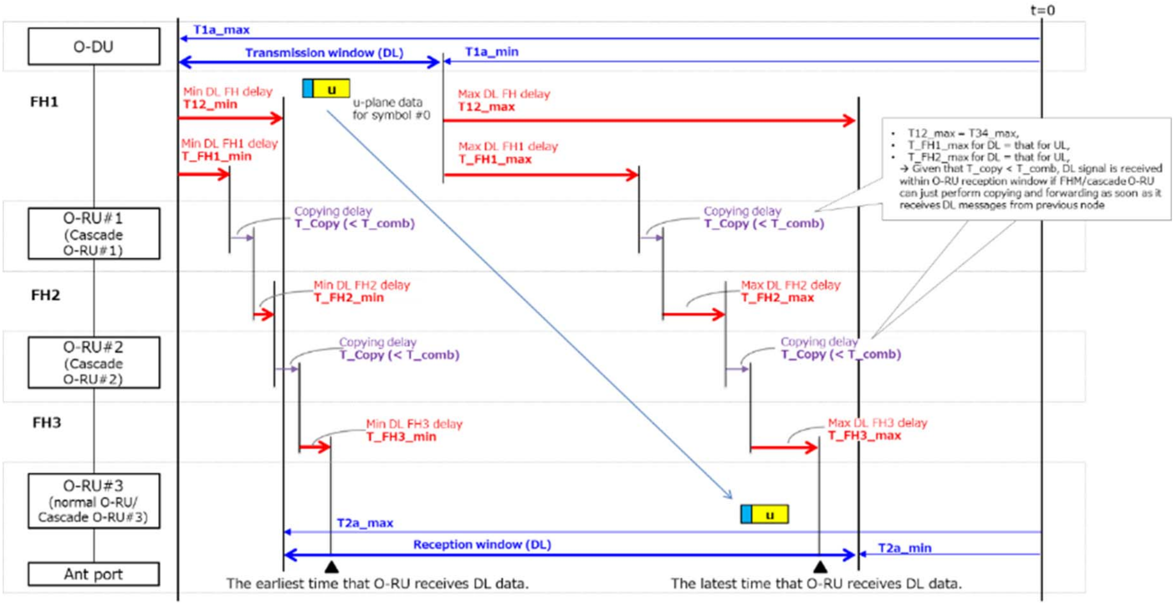


Figure 13-7 : Delay model parameters for Cascade mode (3 O-RUs case, i.e. *N*=3, is shown as an example)

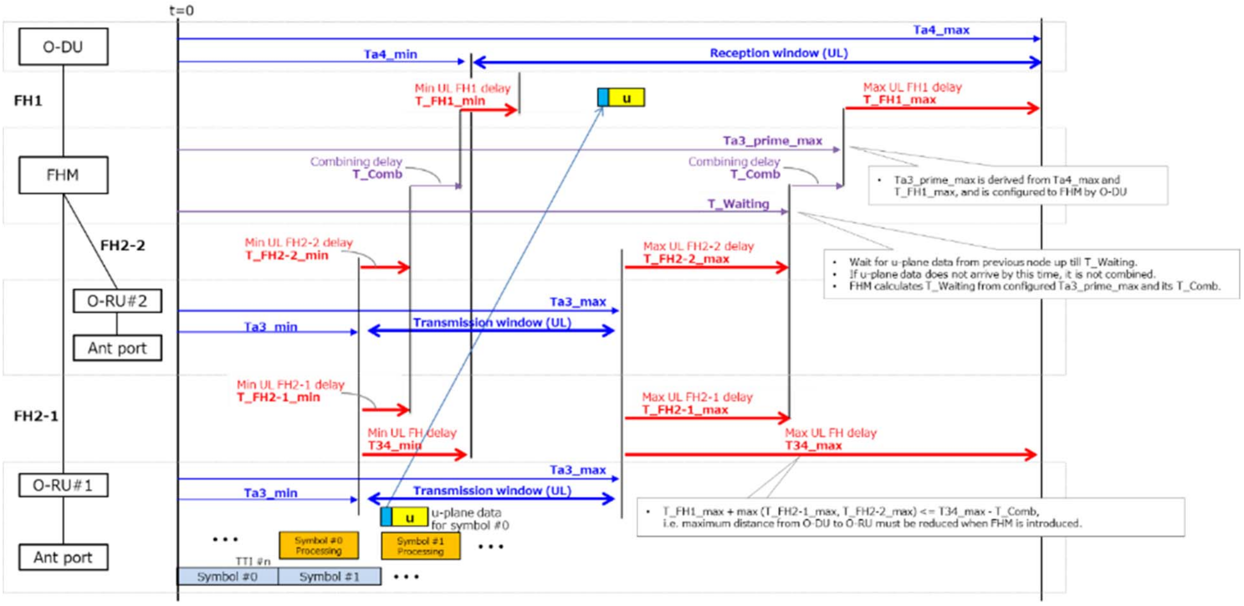
It is assumed that additional delay due to combining in UL case is larger than copying in DL case. Therefore as long as fronthaul and FHM/Cascade O-RU are configured to satisfy delay management for UL case, nothing additional is needed for DL delay management with shared cell. I.e., FHM/Cascade O-RU can just perform copying and forwarding as soon as it receives DL messages from O-DU (FHM mode)/north-node (Cascade mode), and this will ensure that DL messages are received within O-RU reception window. Although the additional requirement for DL is not needed, if O-DU knows processing delay for copy which is shorter than T_Comb, it may be beneficial for O-DU transmission window optimization. For this reason, T_Copy is specified as O-RU capability. Note that it would need additional consideration for FDD cell (DL only band) formed of one DL carrier.

Delay parameter for shared cell for DL is summarized in Table 13-3.

Table 13-3 : Delay parameters for shared cell for DL

Parameter	Description	Note
T_Copy	Corresponding to the maximum FHM or Cascade O-RU processing delay between receiving an IQ sample over the fronthaul interface from O-DU (FHM mode) or the north-node (Cascade mode), coping it and transmitting it over the fronthaul interface to O-RU (FHM mode) or the south-node (Cascade mode). In addition to the above, in FHM mode with selective transmission, T_Copy also includes reconstructing U-Plane message.	Capability parameter. FHM or Cascade O-RU reports the value by M-Plane.

13.3.2 UL delay management for Shared cell



- Ta3_prime_max is derived from Ta4_max and T_FH1_max, and is configured to FHM by O-DU

- Wait for u-plane data from previous node up till T_Waiting.
- If u-plane data does not arrive by this time, it is not combined.
- FHM calculates T_Waiting from configured Ta3_prime_max and its T_Comb.

- T_FH1_max + max (T_FH2-1_max, T_FH2-2_max) <= T34_max - T_Comb,
  i.e. maximum distance from O-DU to O-RU must be reduced when FHM is introduced.

Figure 13-8 : Delay model parameters for FHM mode (2 O-RUs case, i.e. N_m =2, is illustrated as an example)

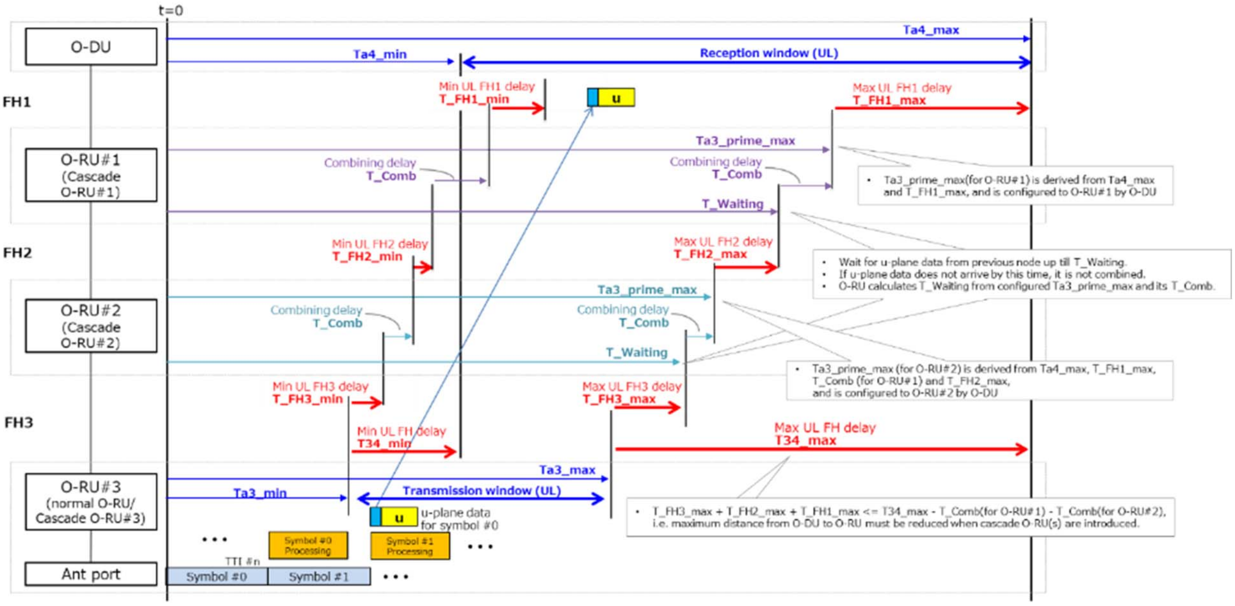
UL delay model parameters for FHM mode are shown in Figure 13-8. Since FHM processing delay for combining UL U-Plane messages effectively adds to the total fronthaul delay between O-DU and O-RU, total fronthaul distance between O-DU and O-RU shall be reduced compared to the case when there is no FHM in order to keep the same total fronthaul delay between O-DU and O-RU so as to ensure UL U-Plane messages arrive at O-DU within O-DU reception window. In other words, configuration of fronthaul including FHM shall meet;

T_FH1_max + *max* (T_FH2-1_max, ... , T_FH2-N_m_max) <= T34_max - T_Comb,
where
T_FH1_max is the maximum transport delay between FHM and O-DU,
T_FH2-*n*_max is the maximum transport delay between O-RU#*n* and FHM (*n* = 1, ... , N_m), and
T_Comb is FHM processing delay for combining which depends on FHM capability and is reported via M-Plane.

To ensure that UL U-Plane messages arrive at O-DU within O-DU reception window, the latest time FHM can send combined UL U-Plane messages towards O-DU is “Ta4_max - T_FH1_max”. Since the value of T_FH1_max depends on position of FHM, FHM needs to be told about the latest time. This time is defined as Ta3_prime_max and needs to be configured to FHM via M-Plane. Considering T_Comb, this means that UL U-Plane messages received by “Ta3_prime_max – T_Comb” are subject to UL U-Plane message combining. In other words, even if FHM does not receive all UL U-Plane messages from O-RUs by “Ta3_prime_max – T_Comb”, FHM shall combine whatever received UL U-Plane messages and send it to O-DU by Ta3_prime_max. This time “Ta3_prime_max (configured) – T_Comb

(FHM capability)" is defined as T_Waiting and calculated at FHM. In addition, in order to combine all UL U-Plane messages from O-RUs, T_Waiting needs to be larger than or equal to "Ta3_max + *max* (T_FH2-1_max, ... , T_FH2-N_m_max)". Therefore, configured Ta3_prime_max should meet following condition;

Ta3_max + *max* (T_FH2-1_max, ... , T_FH2-N_m_max) + T_Comb <= Ta3_prime_max <= Ta4_max - T_FH1_max



- Ta3_prime_max(for O-RU#1) is derived from Ta4_max and T_FH1_max, and is configured to O-RU#1 by O-DU

- Wait for u-plane data from previous node up till T_Waiting.
- If u-plane data does not arrive by this time, it is not combined.
- O-RU calculates T_Waiting from configured Ta3_prime_max and its T_Comb.

- Ta3_prime_max (for O-RU#2) is derived from Ta4_max, T_FH1_max, T_Comb (for O-RU#1) and T_FH2_max, and is configured to O-RU#2 by O-DU

- T_FH3_max + T_FH2_max + T_FH1_max <= T34_max - T_Comb(for O-RU#1) - T_Comb(for O-RU#2), i.e. maximum distance from O-DU to O-RU must be reduced when cascade O-RU(s) are introduced.

Figure 13-9 : Delay model parameters for cascade mode (3 O-RUs case, i.e. *N*=3, is shown as an example)

UL delay model parameters for Cascade mode are shown in Figure 13-9. In a similar manner with FHM mode, configuration of fronthaul including Cascade O-RU shall meet;

$$\sum_{i=1}^{N} \text{T\_FH}i\text{\_max} \;\leq\; \text{T34\_max} - \sum_{i=1}^{N-1} \text{T\_Comb (for Cascade O-RU\#}i\text{)}$$

where

T_FH*n*_max is

the maximum transport delay between O-RU#1 and O-DU (*n*=1),

the maximum transport delay between O-RU#n and O-RU#*n*-1 (*n*=2 … *N*),

T_Comb is Cascade O-RU processing delay for combining which depends on Cascade O-RU capability and is reported via M-Plane, and

T34_max is the maximum uplink transport delay between O-DU and the last O-RU (as specified in clause 4.3).

In addition, in a similar manner with FHM mode, Ta3_prime_max configured to each Cascade O-RU shall meet following condition;

$$\text{Ta3\_max} + \sum_{i=n+1}^{N} \text{T\_FH}i\text{\_max} + \sum_{i=n}^{N-1} \text{T\_Comb (for Cascade O-RU\#}i\text{)} \;\leq\; \text{Ta3\_prime\_max (for Cascade O-RU\#}n\text{), and}$$

$$\text{Ta3\_prime\_max (for Cascade O-RU\#}n\text{)} \;\leq\; \text{Ta4\_max} - \sum_{i=1}^{n} \text{T\_FH}i\text{\_max} - \begin{cases} \displaystyle\sum_{i=1}^{n-1} \text{T\_Comb (for Cascade O-RU\#}i\text{)}, & n \geq 2 \\ 0, & n = 1 \end{cases}$$

Delay parameters for shared cell for UL are summarized in Table 13-4.

Table 13-4 : Delay parameters for shared cell for UL

Parameter	Description	Note
T_Comb	Corresponding to the maximum FHM or Cascade O-RU processing delay between receiving an IQ sample over the fronthaul interface from all O-RUs (FHM mode) or the south-node (Cascade mode), combining them and transmitting it over the fronthaul interface to O-DU (FHM mode) or the north-node (Cascade mode).	Capability parameter. FHM or Cascade O-RU reports the value by M-Plane.
Ta3_prime_max	The latest time that FHM or Cascade O-RU is allowed to send UL U-plane message to O-DU (FHM mode) or north-node (Cascade mode) relative to reception timing at O-RU antenna.	Configuration parameter. Value configured to FHM or Cascade O-RU by M-Plane. See above for appropriate setting.
T_Waiting	Time when FHM or Cascade O-RU shall wait UL U-Plane message sent by O-RU (FHM mode) or south-node (Cascade mode).	Calculated parameter. FHM or Cascade O-RU calculates from configured Ta3_prime_max and its T_Comb. I.e. this parameter is not defined in M-Plane.

Regarding O-RU delay profile reporting via M-plane;

- For FHM, only T_comb is applicable. Other parameters (T2a and Ta3 related parameters and Tcp_adv_dl) are not applicable for FHM since it does not behave as an O-RU
- For Cascade O-RU, T_comb is applicable. Other parameters (T2a and Ta3 related parameters and Tcp_adv_dl) are also applicable and the values are for the case it behaves as a normal O-RU

13.4 S-plane for Shared cell

The same synchronization framework and requirements specified in clause 11 also apply to the FHM, O-DU and O-RU for Shared cell, where the FHM (in the FHM mode), and the cascaded O-RU(s) (in the Cascade mode), are typically regarded as Ethernet switches on the synchronization chain, meeting the requirements specified for T-BC or T-TC Ethernet clocks in Clause 11 from an S-plane point of view. Whenever an O-RU is not on the synchronization path to other O-RUs, the O-RU is not required to support T-BC or T-TC functions. In any case the existing limits specified in clause 11 between the S-plane input port and RF port apply to all O-RUs.

13.4.1 Node behavior at O-RU in FREERUN state

13.4.1.1 O-DU

If synchronization states on all O-RUs in a shared cell used for operating the same cell transit to the FREERUN or HOLDOVER state, the O-DU shall stop sending C-Plane and U-Plane related data to these O-RUs. In other words, the O-DU shall continue sending unless all O-RUs in the shared cell used for operating the same cell transit to the FREERUN or HOLDOVER state.

13.4.1.2 O-RU

If synchronization state on a cascade O-RU transits to the FREERUN state, the cascade O-RU shall autonomously stop RF transmission, switch all carriers to INACTIVE state and send a notification to the O-DU about synchronization and carriers state change. The Cascade O-RU shall enable to continue the function for copy and forward to south-node when the O-RU receives the C-Plane and U-plane related data from the north-node and shall be enable to continue forwarding without any combining function when the O-RU receives the U-plane related data from the south-node.

If synchronization state on an FHM transits to the FREERUN state, the FHM shall send a notification to the O-DU about the synchronization. The FHM shall enable to continue function for copy and forward to south-nodes when the O-RU receives the C-Plane and U-plane related data from the north-node and shall be enable to continue the function for combine and forward to the north-node when the FHM receives the U-plane related data from the south-nodes.

Rationale: The other O-RUs in a shared cell used for operating the same cell as cascade O-RU or FHM might be still LOCKED synchronization state. The cascade O-RU or FHM shall be able to continue copy and combine functions for the other O-RUs in chain or star topology which provides valid U-Plane and C-Plane related data from/to the O-RUs. However, it has to be considered that if one of the O-RUs in the cascaded chain or FHM in FHM mode moves to FREERUN or HOLDOVER, the clockClass values advertised by this FHM/O-RU towards south-node(s) might impact the sync plane of all the other RUs which are listening on PRTC clockClass values and might trigger state change on all O-RUs which eventually might move to FREERUN or HOLDOVER based on the sync state of that particular O-RU.

13.5 Cascade-FHM mode

13.5.1 General

Following clauses defines one hybrid mode Cascade-FHM. The mode is shown in Figure 13-10.

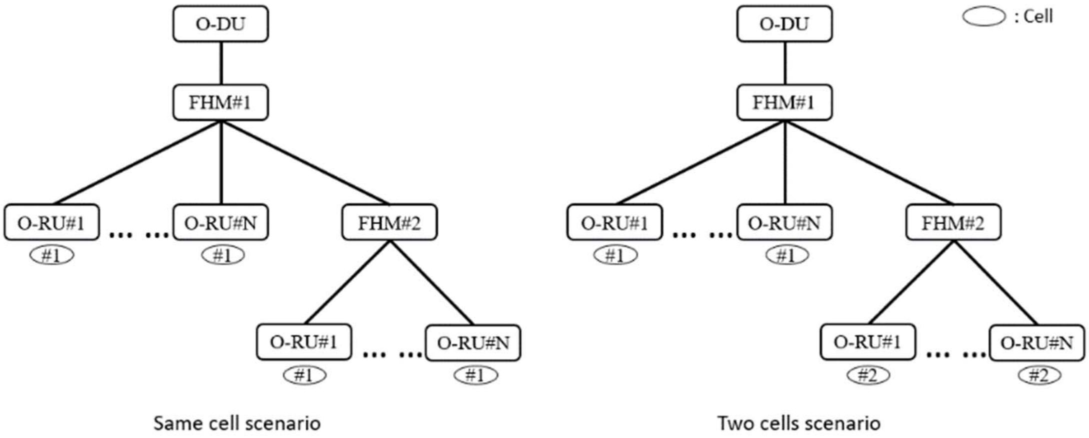


Figure 13-10: Cascade-FHM mode

Figure 13-10 shows two typical scenarios in Cascade-FHM mode, one is Same cell scenario, that is O-RU(s) serving cascaded FHMs belong to Same cell, another is Two cells scenario, that is O-RU(s) serving different FHM belong to different cell. Other different cell scenarios are also possible.

Cascade-FHM mode is realized by at least two chained FHMs and O-RUs connected to them. Star-like topology where many FHMs would be connected to one FHM is not in scope.

Limitation: in this version of the specification maximum level of cascaded FHMs is limited to 2, see below note. As for other usages they can be discussed in future as needed.

NOTE: In current version of specification, the reasons for limited depth of FHM chain are as follows: 1) Noise floor level may have negative impact when deeper chain is constructed. Such impact is not yet analyzed; 2) Delay management may be affected by deeper chain and also requires further analysis. 3) Cascade-FHM mode is suitable to be deployed in Indoor Distribution System, and generally the usage with two cascaded FHMs is typical and sufficient.

The first cascade FHM nearest to O-DU is named **FHM#1**, the second cascade FHM is named **FHM#2**. For FHM#1, its north-node is O-DU and south-nodes are its O-RUs and FHM#2. For FHM#2, its north-node is FHM#1 and south-nodes are its O-RUs.

13.5.2 Copy and Combine function

DL Copy function and UL Combine function for Cascade-FHM mode are actually same with FHM mode with a little attention that FHM#1 regards FHM#2 as its one O-RU from aspect of Copy and Combine. Following figures show downlink and uplink flow for Same cell scenario (Figure 13-11) and Two cells scenario (Figure 13-12).

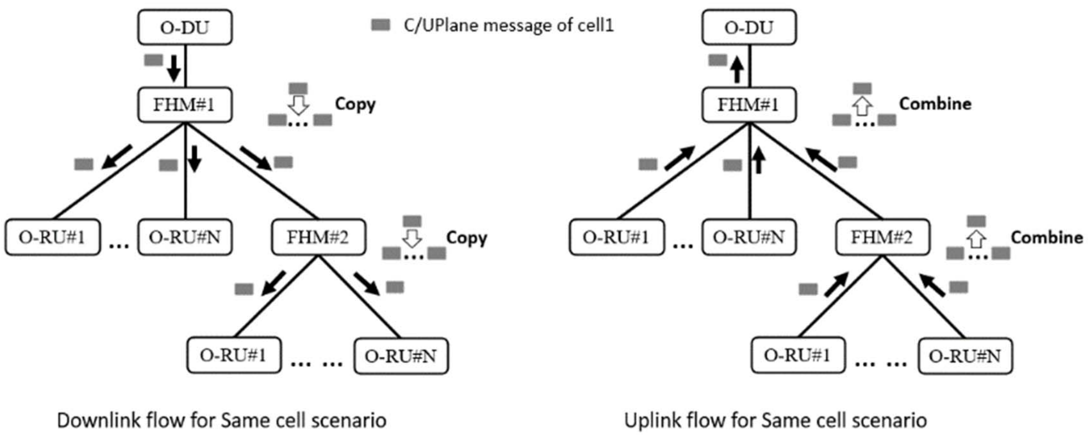

O-DU
C/UPlane message of cell1
FHM#1
Copy
O-RU#1
O-RU#N
FHM#2
O-RU#1
O-RU#N
Downlink flow for Same cell scenario
O-DU
FHM#1
Combine
O-RU#1
O-RU#N
FHM#2
O-RU#1
O-RU#N
Uplink flow for Same cell scenario

Figure 13-11: Cascade-FHM flow for Same cell

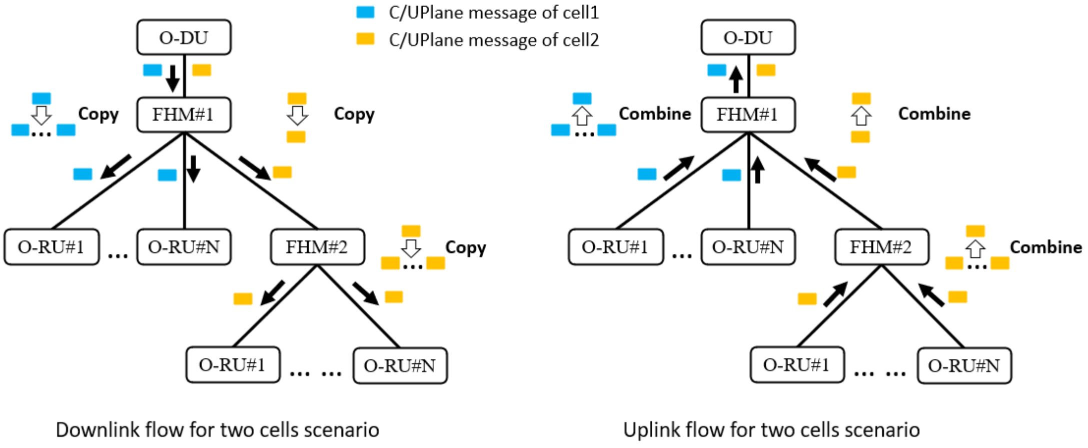

C/UPlane message of cell1
C/UPlane message of cell2
O-DU
FHM#1
Copy
O-RU#1
O-RU#N
FHM#2
O-RU#1
O-RU#N
Downlink flow for two cells scenario
O-DU
FHM#1
Combine
O-RU#1
O-RU#N
FHM#2
O-RU#1
O-RU#N
Uplink flow for two cells scenario

Figure 13-12: Cascade-FHM flow for two cells

13.5.3 Delay management

Delay management for Cascade-FHM mode looks like the combination of FHM mode and Cascade mode while FHM#2 cannot be treated as one normal O-RU from aspect of delay management since it has own processing delay and transport delay towards its O-RUs.

Assuming the configuration is two Cascaded FHMs: FHM#1 has one O-RU and FHM#2 has two O-RUs.

1) The DL delay model parameters in Same cell scenario are shown in Figure 13-13.

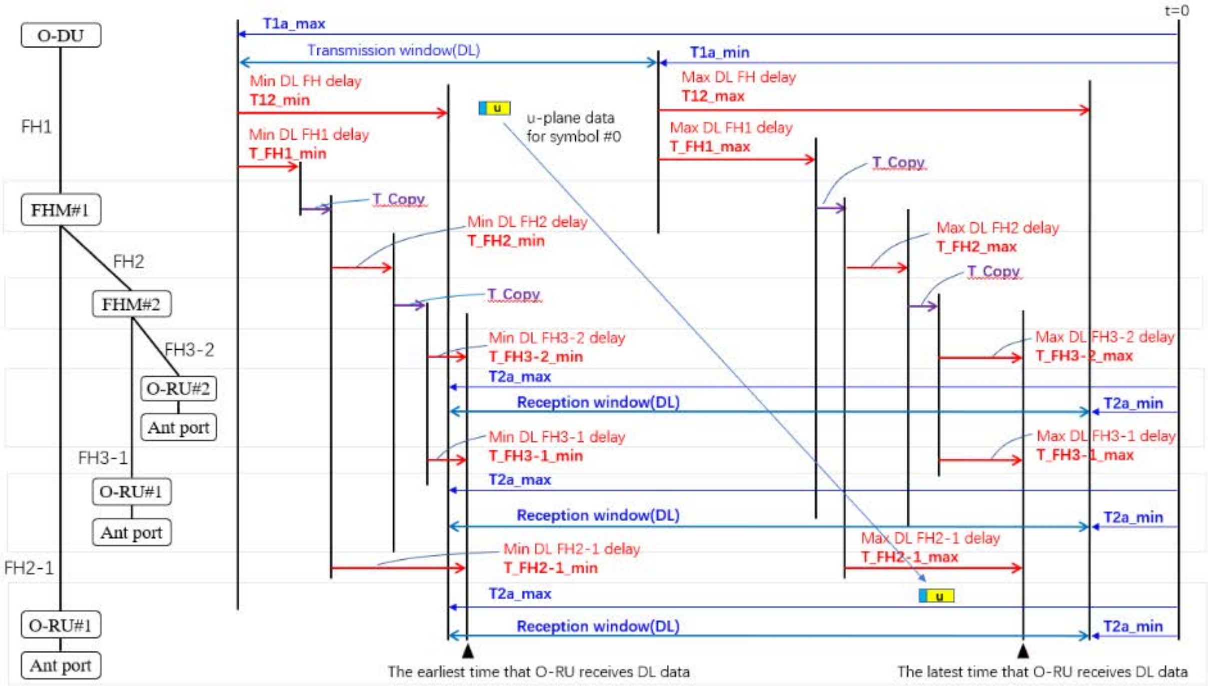


Figure 13-13: DL delay model parameters for Cascade-FHM mode

To ensure DL U-Plane messages arrive at O-RU within O-RU reception window the configuration of fronthaul including two cascaded FHMs shall meet:

T_FH1_min + *min* (*min* (T_FH2_1_min, ... , T_FH2-*N*_min), T_FH2_min + *min* (T_FH3_1_min, ... , T_FH3-*M*_min) + T_Copy) >= T12_min – T_Copy

where

T_FH1_min is the minimum transport delay between FHM#1 and O-DU,
T_FH2_min is the minimum transport delay between FHM#2 and FHM#1,
T_FH2-*n*_min is the minimum transport delay between O-RU#*n* and FHM#1 (*n* = 1, ... , *N*),
N is the total number of O-RUs realizing a particular cell *m* under FHM#1, and in this case *N*=1,
T_FH3-*m*_min is the minimum transport delay between O-RU#*m* and FHM#2 (*m* = 1, ... , *M*),
M is the total number of O-RUs realizing a particular cell *m* under FHM#2, and in this case *M*=2,
T_Copy is FHM processing delay for copying which depends on FHM capability and is reported via M-Plane.

T_FH1_max + *max* (*max* (T_FH2_1_max, ... , T_FH2-*N*_max), T_FH2_max + *max* (T_FH3_1_max, ... , T_FH3-*M*_max) + T_Copy) <= T12_max – T_Copy

where

T_FH1_max is the maximum transport delay between FHM#1 and O-DU,
T_FH2_max is the maximum transport delay between FHM#2 and FHM#1,
T_FH2-*n*_max is the maximum transport delay between O-RU#*n* and FHM#1 (*n* = 1, ... , *N*),
N is the total number of O-RUs realizing a particular cell *m* under FHM#1, and in this case *N*=1,
T_FH3-*m*_max is the maximum transport delay between O-RU#*m* and FHM#2 (*m* = 1, ... , *M*),
M is the total number of O-RUs realizing a particular cell *m* under FHM#2, and in this case *M*=2,
T_Copy is FHM processing delay for copying which depends on FHM capability and is reported via M-Plane.

2) The UL delay model parameters in Same cell scenario are shown in Figure 13-14.

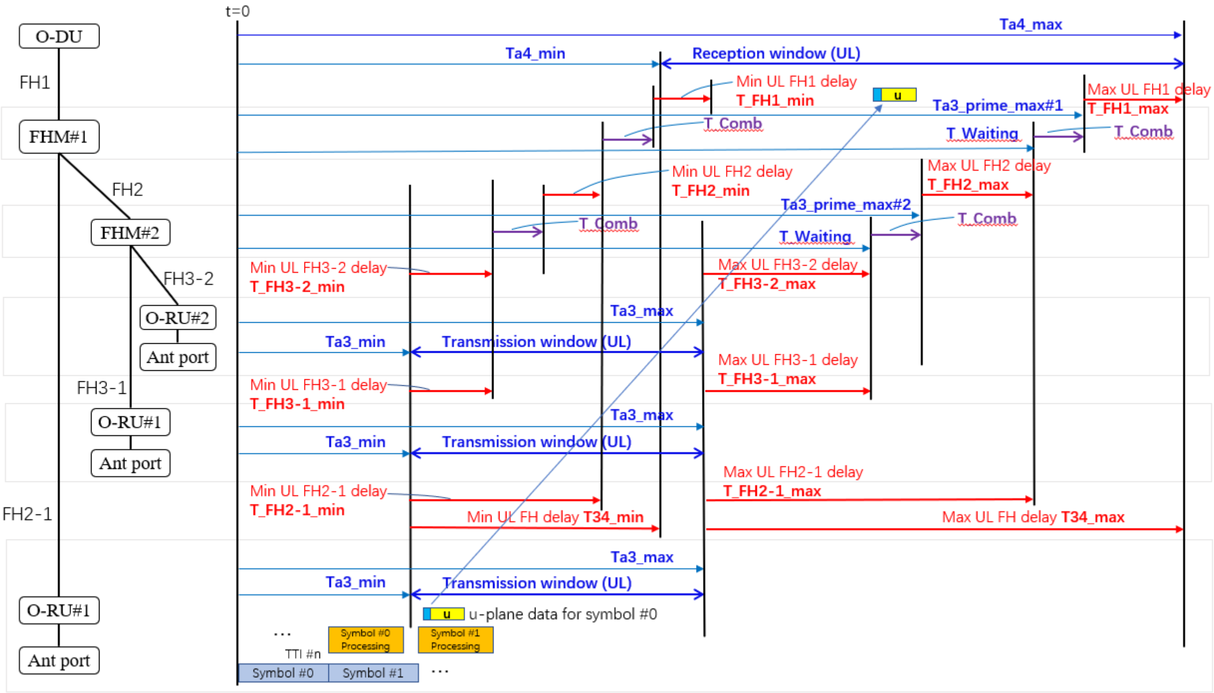


Figure 13-14: UL delay model parameters for Cascade-FHM mode

To ensure UL U-Plane messages arrive at O-DU within O-DU reception window the configuration of fronthaul including two cascaded FHMs shall meet:

T_FH1_min + *min* (*min* (T_FH2_1_min, ... , T_FH2-*N*_min), T_FH2_min + *min* (T_FH3_1_min, ... , T_FH3-*M*_min) + T_Comb) >= T34_min – T_Comb

where

T_FH1_min is the minimum transport delay between FHM#1 and O-DU,
T_FH2_min is the minimum transport delay between FHM#2 and FHM#1,
T_FH2-*n*_min is the minimum transport delay between O-RU#*n* and FHM#1 (*n* = 1, ... , *N*),
N is the total number of O-RUs realizing a particular cell *m* under FHM#1, and in this case *N*=1,
T_FH3-*m*_min is the minimum transport delay between O-RU#*m* and FHM#2 (*m* = 1, ... , *M*),
M is the total number of O-RUs realizing a particular cell *m* under FHM#2, and in this case *M*=2,
T_Copy is FHM processing delay for copying which depends on FHM capability and is reported via M-Plane.

T_FH1_max + *max* (*max* (T_FH2-1_max, ... , T_FH2-*N*_max), T_FH2_max + *max* (T_FH3-1_max, ... , T_FH3-*M*_max) + T_Comb) <= T34_max – T_Comb,

where

T_FH1_max is the maximum transport delay between FHM#1 and O-DU,
T_FH2_max is the maximum transport delay between FHM#2 and FHM#1,
T_FH2-*n*_max is the maximum transport delay between O-RU#*n* and FHM#1 (*n* = 1, ... , *N*),
N is the total number of O-RUs realizing a particular cell *m* under FHM#1, and in this case *N*=1,
T_FH3-*m*_max is the maximum transport delay between O-RU#*m* and FHM#2 (*m* = 1, ... , *M*),
M is the total number of O-RUs realizing a particular cell *m* under FHM#2, and in this case *M*=2,
T_Comb is FHM processing delay for combining which depends on FHM capability and is reported via M-Plane.

In addition, in order to combine all UL U-Plane messages from O-RUs the configured Ta3_prime_max for two cascaded FHMs should meet following condition:

Ta3_max + *max* (T_FH3-1_max, ... , T_FH3-*M*_max) + T_Comb <= Ta3_prime_max_FHM#2,

Ta3_max + *max* (*max* (T_FH2-1_max, ... , T_FH2-*N*_max), T_FH2_max + *max* (T_FH3-1_max, ... , T_FH3-*M*_max) + T_Comb) + T_Comb <= Ta3_prime_max#1 <= Ta4_max - T_FH1_max,

where

Ta3_prime_max_FHM#1 is Ta3_prime_max for FHM#1,
Ta3_prime_max_FHM#2 is Ta3_prime_max for FHM#2,

T_FH2-*n*_max is the maximum transport delay between O-RU#*n* and FHM#1 (*n* = 1, ... , *N*),
N is the total number of O-RUs realizing a particular cell *m* under FHM#1, and in this case *N*=1,
T_FH3-*m*_max is the maximum transport delay between O-RU#*m* and FHM#2 (*m* = 1, ... , *M*),
M is the total number of O-RUs realizing a particular cell *m* under FHM#2, and in this case *M*=2.

Annex A (informative): Compression Methods

A.1 Block Floating Point Compression

The compressed data representation is as follows. For each PRB, In-phase (I) and Quadrature (Q) samples are converted into floating point format. Subsequently, the samples are presented as a compressed bit sign and a mantissa (e.g., 9-bit) and a shared exponent (see **Figure A**.1). The compression procedure receives 12 subcarriers with 24 uncompressed I and Q samples. The I and Q samples are subsequently compressed to a bit signed mantissa and unsigned exponent. Further, the exponent is included for each compression block to be sent per PRB (see **Figure A**.2).

NOTE: Mantissa bitwidths are specified in the compHdr field of the relevant U-Plane or C-Plane message (range 1-16 bitwidth). However, support on O-RU HW for particular mantissa bitwidths is to be defined on individual basis. That is, there is no requirement on O-RU HW to support all possible mantissa bitwidths.

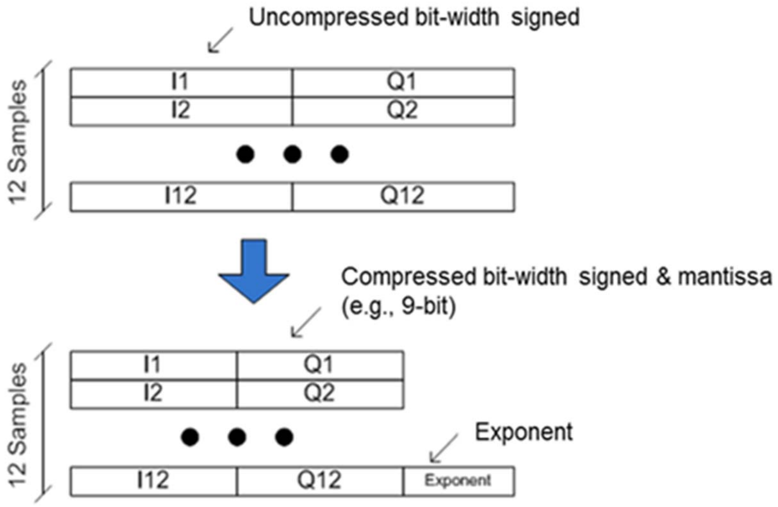


Figure A.1 : Block Floating Point Compression data representation

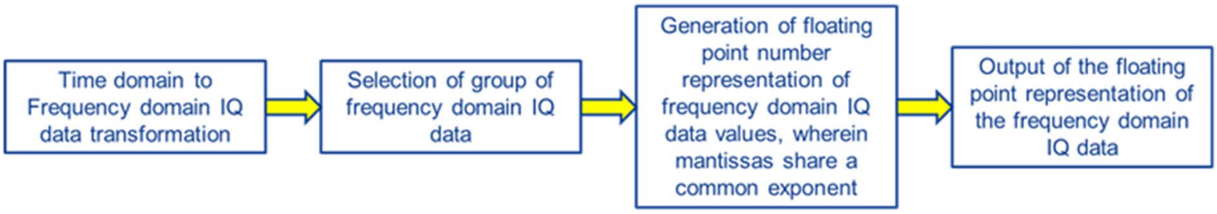

Time domain to Frequency domain IQ data transformation
Selection of group of frequency domain IQ data
Generation of floating point number representation of frequency domain IQ data values, wherein mantissas share a common exponent
Output of the floating point representation of the frequency domain IQ data

Figure A.2 : Block Floating Point Compression process overview

A.1.1 Block Floating Point Compression Algorithm

The following pseudo code depicts an example reference implementation of the compression algorithm.

Inputs:

- fPRB - Original physical resource block (PRB), 12 complex resource elements with “native” word length of the implementation e.g. 24 bits (UL should use as accurate value as possible from FFT & beamforming)
- iqWidth - Word length after compression (includes sign bit)

Outputs:

- cPRB - Compressed PRB, 12 complex resource elements with word length iqWidth
- exponent - Common exponent for compressed PRB

```
// Find max and min
$maxV$ = max(Re(fPRB), Im(fPRB)), $minV$ = min(Re(fPRB), Im(fPRB))

// Determine max absolute value
$maxValue$ = max($maxV$, $|minV|$-1)  (msb of negative value can be one higher)

// Calculate exponent
$raw\_exp$ = [floor($\log_2$($maxValue$) +1)] (msb of $maxValue$)

// Calculate shift value and limit to positive
$exponent$ = max($raw\_exp$ – iqWidth + 1, 0)

// Determine right shift value
$scaler = 2^{-exponent}$

For iRe = 1:length(fPRB)
    //Scale and round:
    Re(cPRB(iRE)) =  Quantize ($scaler \times$ Re(fPRB(iRE)))  /* mult. could be bit-shift, Quantize could be or-round */
    Im(cPRB(iRE)) = Quantize ($scaler \times$ Im(fPRB(iRE)))  /* mult. could be bit-shift, Quantize could be or-round */
End
```

A.1.2 Block Floating Point Decompression Algorithm

The following pseudo code depicts an example reference implementation of the decompression algorithm.

Inputs:

- cPRB - Compressed PRB, 12 complex resource elements with word length WL
- exponent - Common exponent for compressed PRB

Outputs:

- fPRB - Decompressed physical resource block (PRB), 12 complex resource elements with “native” word length for further processing. For example, 24 bits or 32 bits

```
//Determine scaler
$scaler = 2^{exponent}$

For iRe = 1:length(cPRB)
    //Scale
    Re(fPRB(iRE)) = $scaler \times$ Re(cPRB(iRE))  /* this could be replaced with a bit-shift operation */
    Im(fPRB(iRE)) = $scaler \times$ Im(cPRB(iRE))  /* this could be replaced with a bit-shift operation */
End
```

A.2 Block Scaling Compression

A block scaling algorithm is proposed which is similar in concept to the Block Floating Point representation except that instead of data being represented by mantissa values and exponent shared within the block, data is instead represented by post-scaled values and a multiplicative scale value shared within the block. It is proposed in the specification that the data block size for this function is a single PRB, same as for the proposed Block Floating Point representation. The following Figure A.3 shows the algorithm in principle (assuming an 8-bit scaler value).

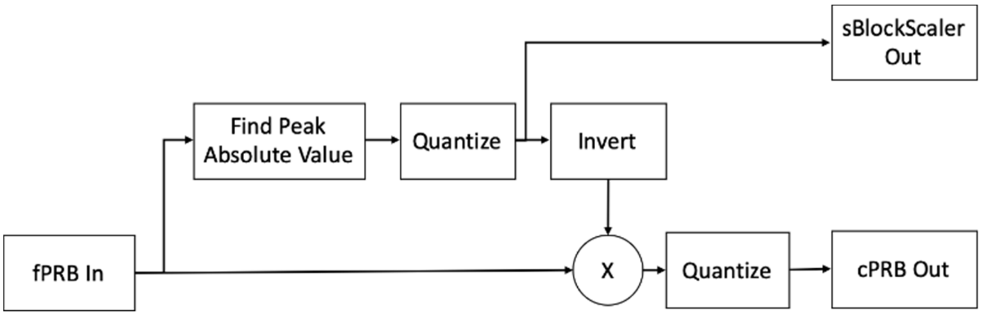


Figure A.3 : Block Scaling Compression Process Diagram

A.2.1 Block Scaling Compression Algorithm

The following pseudo code depicts an example implementation of the compression algorithm.

Inputs:

- fPRB - Original physical resource block (PRB), 12 complex resource elements with “native” word length of the implementation e.g. 16-bits I + 16-bits Q is assumed in this definition.
- iqWidth - Word length after compression (includes sign bit)

Outputs:

- cPRB - Compressed PRB, 12 complex resource elements with word length iqWidth
- sblockScaler- Common scaler for compressed PRB

```
// Find max and min of resource element real and imaginary parts
maxV = max(max(real(fPRB)), max(imag(fPRB)));
minV = min(min(real(fPRB)), min(imag(fPRB)));

// Determine maximum absolute value, which will be 15-bits
maxValue = max(maxV, abs(minV)-1);              // Negative values offset by 1 to fit 15-bits

// Map sBlockScaler to 8-bits
sblockScaler = ceil(maxValue/2^8);              // Q1.7, 0->128 represents 0->1.000

// Calculate inverse of sBlockScaler - this can be implemented as a look-up table
if(sblockScaler==0)
   sBlockScaler = 1;                         // Trap divide by zero
end
inverseBlockScaler = 2^7/sblockScaler;          // Scaled 128->1

// Max output value, used for saturating compressed samples to iqWidth
qs = 2^(iqWidth-1);

// Scale each RE. Scaling assumes 16-bit uncompressed data width.
for iRe =1:length(fPRB)
   cRe_re = round(inverseBlockScaler * real(fPRB(iRe)) / 2^(16-iqWidth));     // Scale real part
   cRe_re = min(max(cRe_re, -qs), qs-1);                                     // Saturate
   cRe_im = round(inverseBlockScaler * imag(fPRB(iRe)) / 2^(16-iqWidth));    // Scale imaginary part
   cRe_im = min(max(cRe_im, -qs), qs-1);                                     // Saturate
   cPRB(iRe) = complex(cRe_re, cRe_im);                                      // Complex output value
end
```

A.2.2 Block Scaling Decompression Algorithm

The following pseudo-code depicts an example reference implementation of the block scaling decompression algorithm.

Inputs:

- cPRB - Compressed PRB, 12 complex resource elements with word length iqWidth
- blockScaler - Common scaler for compressed PRB

Outputs:

- fPRB - Decompressed physical resource block (PRB), 12 complex resource elements with “native” word length for further processing. For example 16-bits I + 16-bits Q is assumed in this definition.

```
// Re-scale each resource element back to 16-bit uncompressed width
for iRe = 1:length(cPRB)
   fRe_re = round(sblockScaler * real(cPRB(iRe)) / 2^(iqWidth-9));
   fRe_im = round(sblockScaler * imag(cPRB(iRe)) / 2^(iqWidth-9));
   fPRB(iRe) = complex(fRe_re, fRe_im);                    // Complex output value
end
```

A.3 µ-Law Compression

A.3.1 µ-Law Compression Algorithm

Inputs:

- prbI & prbQ – Original physical resource block (PRB), 12 complex resource elements with a word length of 16-bits I and 16-bits Q. The input bit width is fixed to 16-bits.
- compBitWidth – the length of I bits and the length of Q bits after compression over the entire PRB. Note that this means that the µ-law compression is really considered as a “block” compression with the block size being one PRB (same as for block floating point and block scaling).

Outputs:

- compI & compQ – compressed PRB, 12 complex resource elements with word length compBitWidth, including sign, exponent and mantissa.
- compShift – the shift applied to the entire PRB.

The O-RAN µ-law compression method combines a simple bit shift operation (for dynamic range) with a nonlinear piece wise approximation of µ-law compression where for implementation efficiency, µ=8 and the sign & mantissa are 1 and 2-bits respectively.

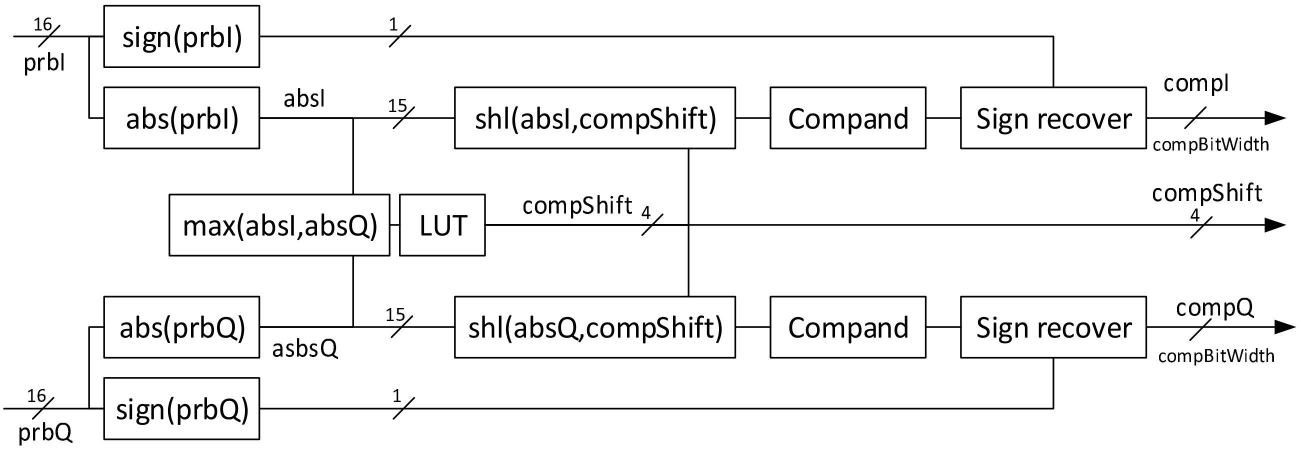


Figure A.4 : µ-Law Compression Algorithm

// extract the sign bit and absolute value for the PRB

signI=sign(prbI), signQ=sign(prbQ), absI=abs(prbI), absQ=abs(prbQ)

// Find the maximum in absI and absQ

```
maxVal=max(absI,absQ)

// Determine compShift, the shift to be applied to the entire PRB

if maxVal>=2^14 then compShift=0
if maxVal<2^14 then compShift=1
if maxVal<2^13 then compShift=2
if maxVal<2^12 then compShift=3
if maxVal<2^11 then compShift=4
if maxVal<2^10 then compShift=5
if maxVal<2^9   then compShift=6
if maxVal<2^8   then compShift=7

// Apply round and shift left (shl – make greater)
absI = shl(absI,compShift)
absQ= shl(absQ,compShift)

// compand each sample, absBitWidth=15

if absI(i) >   (2^absBitWidth-1) then absI(i) = (2^absBitWidth-1) // saturate
if absI(i) <= 2^(absBitWidth-2) then
    compI(i) = absI(i)/2^(absBitWidth- compBitWidth )
elseif absI(i) <= 2^(absBitWidth-1) then
    compI(i) = absI(i)/2^(absBitWidth- compBitWidth +1) + 2^( compBitWidth -3)
else
    compI(i) = absI(i)/2^(absBitWidth- compBitWidth +2) + 2^( compBitWidth -2)
end

if absQ(i) >   (2^absBitWidth-1) then absQ (i) = (2^absBitWidth-1)
if absQ (i) <= 2^(absBitWidth-2) then
    compQ(i) = absQ (i)/2^(absBitWidth- compBitWidth )
elseif absQ (i) <= 2^(absBitWidth-1) then
    compQ (i) = absQ (i)/2^(absBitWidth- compBitWidth +1) + 2^( compBitWidth -3)
else
    compQ (i) = absQ (i)/2^(absBitWidth- compBitWidth +2) + 2^( compBitWidth -2)
end

// re-apply sign

compI = round(signI .* compI);
compQ = round(signQ .* compQ);
```

A.3.2 µ-Law Decompression Algorithm

Inputs:

- compI & compQ– compressed physical resource block (PRB), 12 complex resource elements with a word length of compBitWidth bits I and compBitWidth bits Q.
- compShift – the shift applied to the entire PRB.

Outputs:

- decompI & decompQ – the decompressed PRB, 12 complex resource elements with word length fixed to 16-bits I and 16-bits Q, including sign, exponent and mantissa.

The O-RAN µ-law decompression method is a logical reverse function of the compression method.

```
// extract the sign bit and absolute value for the PRB

signI=sign(compI), signQ=sign(compQ), absI=abs(compI), absQ=abs(compQ)

// decompand each sample, absBitWidth=15

if absI(i) > (2^( compBitWidth -1)-1) then absI (i) = (2^( compBitWidth -1)-1)
if absI (i) <= 2^( compBitWidth -2) then
    decompI(i) = absI (i)*2^(absBitWidth- compBitWidth )
elseif absI (i) <= (2^( compBitWidth -2) + 2^( compBitWidth -3)) then
    decompI(i) = absI (i)*2^(absBitWidth- compBitWidth +1) - 2^13
else
    decompI(i) = absI (i)*2^(absBitWidth- compBitWidth +2) - 2^15
end

if absQ(i) > (2^( compBitWidth -1)-1) then
    absQ (i) = (2^( compBitWidth -1)-1)
if absQ (i) <= 2^( compBitWidth -2) then
    decompQ(i) = absQ (i)*2^(absBitWidth- compBitWidth )
elseif absQ (i) <= (2^( compBitWidth -2) + 2^( compBitWidth -3)) then
    decompQ(i) = absQ (i)*2^(absBitWidth- compBitWidth +1) - 2^13
else
    decompQ(i) = absQ (i)*2^(absBitWidth- compBitWidth +2) - 2^15
end;

//Apply sign and shift
decompI = signI * decompI
decompQ = signQ .* decompQ
decompI = decompI/2^compShift
decompQ = decompQ/2^compShift
```

A.3.3 µ-Law udCompParam and IQ data format

PRB fields are populated as follows:

- udCompParam (8 bits)
    - compBitWidth, 4 bits, (MSB)
    - compShift, 4-bits, (LSB)
- IQ samples, total bits = 12x 2x compBitWidth
    - 1st sample I, compBitWidth-bits
    - 1st sample Q, compBitWidth-bits
    - 2nd sample I, compBitWidth-bits
    - 2nd sample Q, compBitWidth-bits
    - …
    - 12th sample I, compBitWidth-bits
    - 12th sample Q, compBitWidth-bits

A.4 Beamspace Compression and Decompression

This compression algorithm is specific to beamforming weights and is not suitable for user or control IQ data. Hence this compression method will only be used as part of the bfwCompMeth in the C-Plane.

A.4.1 Beamspace Compression Algorithm

The following pseudo code depicts an example reference implementation of the compression algorithm.

Inputs:

- fBV - Original beamforming vector of K complex elements. K is the number of digital antenna ports supported by the O-RU (see clause 12.5.2 and Annex J) and is communicated to the DU during startup by the OAM subsystem. Each element is a complex number with a native bitwidth e.g. 16-bit I, and 16-bit Q. Element fBV(n) of this vector corresponds to beamforming vector element k = n-1 (see clause 12.5.3).

- iqWidth - Word length of each I and Q value after compression (includes sign bit)

Outputs:

- cBV - Compressed beamforming coefficients

- blockScaler- Common scaler for compressed beamforming coefficients
- activeBeamspaceCoefficientMask – Vector of bits activeBeamspaceCoefficientMask(n) for 1 ≤ n ≤ K. Bit value activeBeamspaceCoefficientMask(n)=1 indicates presence of beamspace coefficient associated with the beamforming vector element k) n-1 (see clause 12.5.3) in the compressed beamforming vector. The bit activeBeamspaceCoefficientMask(1) is conveyed in the most significant bit of the first octet of the field bfwCompParam.

```
// Generate DFT basis matrix
for k = 1 to K
    for l = 1 to K
        W (k,l) = exp(i*2*pi*k*l/(K))  // W is a K x K complex matrix
    end for
end for
// Transform into beamspace
cBV = W*fBV // multiplication of a KxK complex matrix with a Kx1 complex vector yields another complex vector.


/* The algorithm is initialized to assume that all Beamspace Coefficients are transmitted across the fronthaul link. */
for k = 1 to K
    activeBeamspaceCoefficientMask(k) = 1
end for


/* At this stage some of the beamspace coefficients may be removed from the vector of coefficients to transmit across the fronthaul. In this example implementation, if the absolute value of a beamspace coefficient is less than ‘threshold’, it is deemed inactive, i.e. the activeBeamspaceCoefficientMask is ‘0’ at that coefficient index and this index is not sent across the fronthaul. The decompression algorithm will assume a value of 0 for that coefficient. The value of threshold can be chosen by the implementer. Other methods to determine active or inactive beamspace coefficients are also allowed and do not violate the specification. */


t = 0
for k = 1 to K
    if abs(cBV(k)) < threshold
        activeIndex(k) = 0
        cBV(k) = null         // remove the element from the vector
    else
        activeIndex(k) = 1
        t = t + 1
    end if
end for
T = t
// Calculate scaler
maxValue = max(abs(Re(cBV)),abs(Im(cBV)))

blockScaler = maxValue /* scaler can be chosen to be larger than maxValue. */
For iRe = 1 to T
    //Scale and round:
```

```
    Re(cBV(iRE)) =  Quantize (Inverse(blockScaler) × Re(cBV(iRE)))  /* Quantize could be truncate or round,
    Inverse can be implemented via look up table or other methods. */
    Im(cBV(iRE)) = Quantize (Inverse(blockScaler) × Im(fBV(iRE)))  /* Quantize could be truncate or round,
    Inverse can be implemented via look up table or other methods. */
End
```

A.4.2 Beamspace Decompression Algorithm

The following pseudo-code depicts an example reference implementation of the block scaling decompression algorithm.

Inputs:

- cBV - Compressed beamforming coefficients
- blockScaler- Common scaler for compressed beamforming coefficients
- activeBeamspaceCoefficientMask – see definition in clause A.4.2

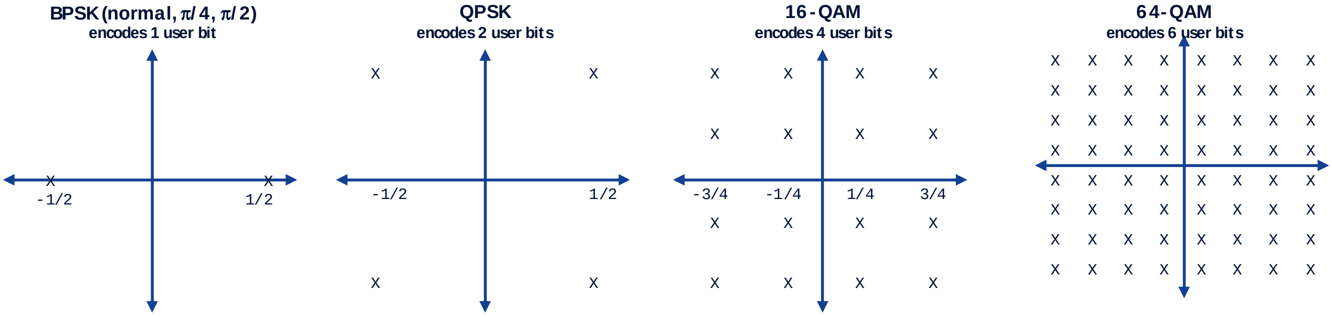
Outputs:

- fBV – Decompressed beamforming vector of K (see K in clause 12.5.2) complex elements. Element fBV(n) of this vector corresponds to beamforming vector element k = n-1 (see clause 12.5.3)

```
m = 0
for k = 1 to K
    if activeBeamspaceCoefficientMask (k) = 1
    //Scale
        fBSC(k) = blockScaler × cBV(m)
        m = m + 1
    else
        fBSC(k) = 0
    end if
end for
// Generate DFT basis matrix
for k = 1 to K
    for l = 1 to K
        W (k,l) = exp(-i*2*pi*k*l/(K))  // W is a K x K complex matrix
    end for
end for
fBV = W * fBSC
```

A.5 Modulation Compression

Modulation compression is an IQ data compression method that may be applied to DL data only and depends on the observation that modulated data symbols may be represented by a very limited number of I and Q bits. For example, a QPSK modulated symbol has only two potential states of I and two potential states of Q, so such a symbol may be represented with no loss of information with a single bit of I and a single bit of Q. Likewise, a 64QAM constellation point (8x8 constellation) may be represented by at most 3 bits of I and 3 bits of Q. This allows a dramatic reduction in DL throughput and approximates that achieved by a 7-3 DL split. See the two figures below for a description of this concept (note **Figure A**.5 and **Figure A**.6 assume a single modulation type, MCS, is used for the data section).



I is represented by 2 bit
Q is represented by 2 bits
(-1/2, 0, and 1/2 are needed)

modulation-compression:
from 32 bits (16I and 16Q)
to 4 bits (2I and 2Q)

I is represented by 1 bit
Q is represented by 1 bit

modulation-compression:
from 32 bits (16I and 16Q)
to 2 bits (1I and 1Q)

I is represented by 2 bits
Q is represented by 2 bits

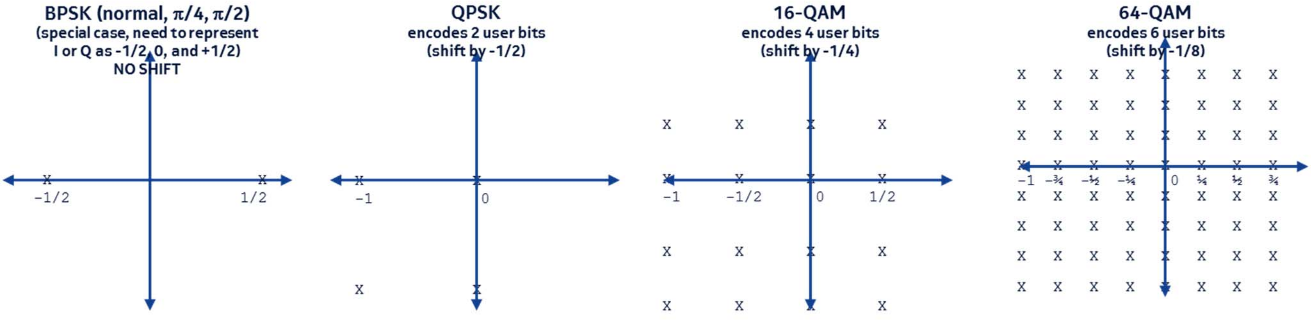
modulation-compression:
from 32 bits (16I and 16Q)
to 4 bits (2I and 2Q)

I is represented by 3 bits
Q is represented by 3 bits

modulation-compression:
from 32 bits (16I and 16Q)
to 6 bits (3I and 3Q)

Figure A.5 : Several Constellation types

To represent the constellation points as I and Q values that also overlap allowing multiple constellation sizes to be represented by a single word-width, the constellations are “shifted” to allow a twos-complement I and Q value to represent any constellation point. The figure below shows the same constellations after shifting.



I is -1/2 or +1/2
Q is 0

I is -1 or 0
Q is -1 or 0

I is -1, -1/2, 0 or 1/2
Q is -1, -1/2, 0 or 1/2

I is -1, -¾, -½, -¼, 0, ¼, ½, or ¾
Q is -1, -¾, -½, -¼, 0, ¼, ½, or ¾

Figure A.6 : Shifted Constellation Points

Once the constellations are shifted, the I and Q values may be encoded in a limited number of bits, being the larger number needed to represent the largest constellation possible in the compression block (the data section). This means that if some data in the section use 64QAM and others use QPSK (e.g. a reference RE) all REs would use the largest needed representation which in this example is 3 bits for I and 3 bits for Q (for 64QAM). This spoils the compression efficiency a little bit but because reference REs are a small fraction of the total number of REs, the efficiency degradation is small (there is further clarification below regarding mixing MCS in a data section). Note that in general, every user will have its own data section (and own beamforming index) so users with high-order modulation may need and use more bits of I and Q while users with lower-order modulation may need and use fewer bits of I and Q.

Note that some constellations should not be shifted. For example, BPSK needs I and Q data to take the values -1/2, zero, and +1/2 (different varieties of BPSK can rotate these as with π/4 BPSK and π/2 BPSK). For this reason, BPSK would use two bits for I and 2 bits for Q; while this seems counterintuitive (BPSK using more bits than QPSK) this is a small penalty given the rarity of BPSK as a modulation type. Here, BPSK would not be shifted. Likewise, PHICH constellations encode 3 states for each I and Q: -1, zero and 1. For this constellation the representation would be -½, 0 and ½ with no constellation shift needed. However, all “QAM” modulations do need a constellation shift (except possibly in mixed-MCS cases). The presence of absence of the need for a constellation shift is indicated by the “csf” field, where for every “one” in the reMask “csf” indicates whether to shift (csf=1) or not (csf=0) the associated RE.

Mixed-MCS cases represent another example when constellations are not shifted. This can occur when user data REs (at high MCS) and signaling data REs (at low MCS) are in the same PRB hence in the same data section. The reMask discriminates the REs at high MCS from the REs at the lower MCS (and provides different modCompScaler or mcScaleOffset values for the different-MCS data), but all the REs in the PRB shall use the same number of I and Q bit-widths. In this case only the high-MCS constellation is shifted, the lower-MCS constellation is not shifted because its data points already overlap with the shifted high-MCS data points. Figure A-7 shows an example of 16QAM data overlain with 64QAM data.

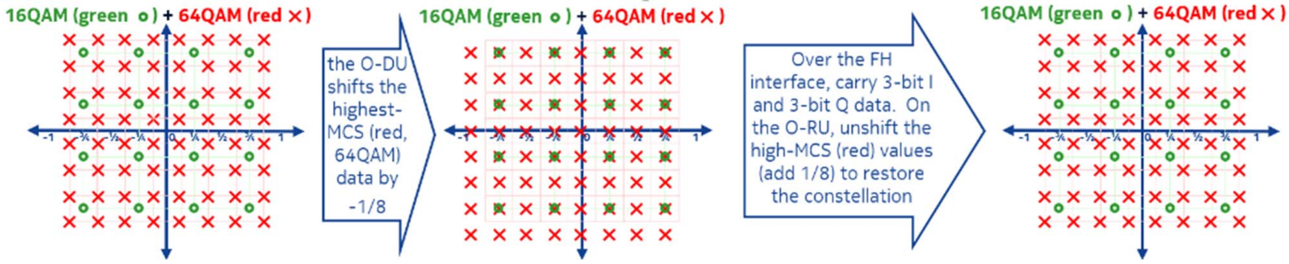

16QAM (green o) + 64QAM (red ×)
the O-DU shifts the highest-MCS (red, 64QAM) data by -1/8
16QAM (green o) + 64QAM (red ×)
Over the FH interface, carry 3-bit I and 3-bit Q data. On the O-RU, unshift the high-MCS (red) values (add 1/8) to restore the constellation
16QAM (green o) + 64QAM (red ×)
-1
-¾
-½
-¼
0
¼
½
¾
1

Figure A.7: Multiple-MCS example for Modulation Compression

In the Figure A.7 example, for overlain 16QAM (green) and 64QAM (red), the high-MCS (red) points are shifted by ½ the high-MCS resolution (here, -1/8) to allow all points to share the same "grid", as shown in the middle figure wherein the red and green points overlay. All I and Q values can be represented by 3 bits each on the fronthaul interface.

The O-DU uses the constellation shift flag (csf) to tell the O-RU which data (red points) to "unshift" by adding 1/8 to them, thereby restoring the original constellation values. After that, modCompScaler (or mcScaleOffset) is applied to set the data to the correct power levels (separate modCompScaler or mcScaleOffset values may be used for the differing MCS data).

When decompressing, the O-RU shall "unshift" the constellation (or perhaps not, depending on "csf") and also apply a scale factor for the constellation types represented in the section. There are expected to be either one or two modulation types in the section, no more. The modulation type is inferred from the reMask bits, where each "one" bit indicates the shift command ("csf") and scale factor ("modCompScaler" when using extension type 4, and "mcScaleOffset" when using extension type 5) for the REs in the subject PRB. The scale factor allows not only for correcting for different constellation scaling (e.g. for multiplexed channel data in a PRB inclusing QPSK and 16QAM, QPSK involves a 2/sqrt(2) factor while 16QAM involves a 4/sqrt(10) factor), but also allows different channel power scaling which is permitted as a 3GPP option.

Note that this compression method is essentially lossless, except that the scale factors, being 15 bits, impose a limit on the accuracy of representation. 15 bits is considered sufficient for all LTE and NR data representations.

When compressing, constellation points are shifted by the shift value defined in Table 7-26 in clause 7.4.7.4.1, and the I and Q values are represented as signed two's complement fractional notation and included in the U-plane message as udIqWidth bit vectors where udIqWidth is dependent on the modulation constellation type and is defined in Table 7-26.

Here is the specific decompression algorithm intended by this approach:

1. Read $iqSample$ as an udIqWidth bit vector in the U-plane message [this is all the IQ data in the data section]

2. Map $iqSample$ $[0,2^{\mathrm{udIqWidth}}-1]$ to $iqSampleFx$ $[-1,1)$ assuming that the udIqWidth bits are represented as Q1.(udIqWidth-1) [this is the normal twos-complement representation of the I and Q samples represented in fractional notation].

3X. For each RE in the PRB (using Section Extension =4):

3Xa: fetch the *"csf"* and *"modCompScaler"* values for which this RE has a "1" in the reMask

3Xb. If "csf" $==\ 1$ then $iqSampleFx\ =\ iqSampleFx\ +\ 2^{-\mathrm{udIqWidth}}$ [this is "unshifting" the constellation point].

3Xc. $iqSampleScaled\ =\ \mathrm{modCompScaler} \times iqSampleFx \times \sqrt{2}$ [this scales the constellation point]

3Y. For each RE in the PRB (using Section Extension =5):

3Ya: fetch the *"csf"* and *"mcScaleOffset"* values for which this RE has a "1" in the relevant mcScaleReMask

3Yb. If "csf" $==$ 1 then $iqSampleFx = iqSampleFx + 2^{-\mathrm{udIqWidth}}$ [this is "unshifting" the constellation point].

3Yc. $iqSampleScaled = \mathrm{mcScaleOffset} \times iqSampleFx \times \sqrt{2}$ [this scales the constellation point]

After decompression, | *iqSampleScaled* | shall be ≤ 1 and a value of | *iqSampleScaled* | = 1.0 matches 0 dBFS.

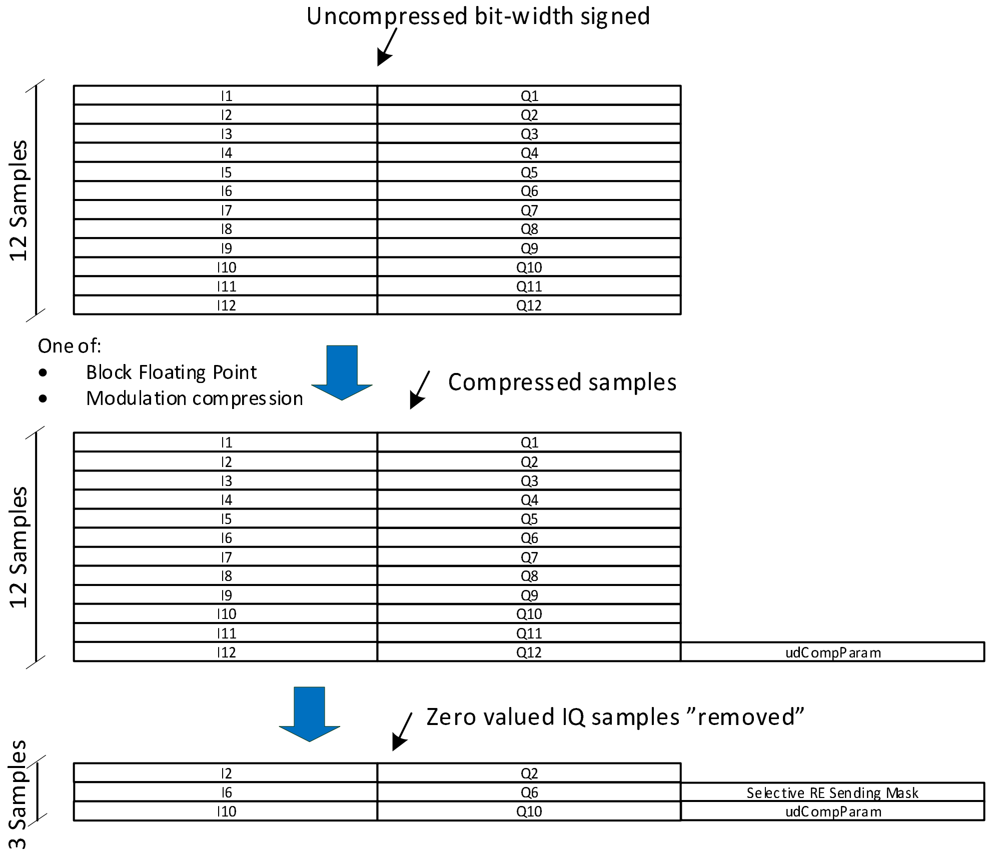
A.6 Selective RE sending Compression

The compressed data representation is as follows. For each IQ-samples where both the I- and the Q-value are 0 (zero) corresponding bit is set to the Selective RE sending bitmask (sReSMask) and that IQ-sample will not be transmitted over the interface in the U-Plane message.

If the total number of bits for the transmitted IQ-samples is not a multiple of 8 (1 byte) then bit-padding will be performed after the last part of the last IQ-samples so that byte alignment is achieved.

Figure below shows an example where 9 out of the 12 IQ-samples in a PRB are zero and thus removed from being sent in the corresponding U-Plane message.

The compression algorithm is performed on the original 12 IQ-samples including any possible zero valued IQ-samples.



Annex B (informative): Delay Management Use Cases

B.1 General

Intra-PHY lower layer fronthaul split has characteristic of a stringent bandwidth and tight latency requirement. This implies use of a special "Fronthaul Service Profile" to be supported by the transport network, and which may differ depending on the operating environment, topology and target use cases. The general concept and latency boundaries is based on eCPRI reference points for delay management definitions (See Figure B.1). However, this specification additionally differentiates between DL (Figure B.2) and UL (Figure B.3) latency boundaries. The parameters and how these are determined for a pre-defined latency configuration are explained below (an actual example of the parameters for a specific use case are presented in Annex B.2). Pre-defined latency is necessary when actual latency measurements are not provided; both the use of pre-defined latency value and use of a method for measuring actual network latency in the DL and UL are supported in this specification.

The following assumptions are considered for the delay boundaries definitions:

- Tcp_adv_dl: Smallest time advance to receive Downlink Data C-Plane message before the first IQ data can be processed
- The fronthaul transmission delay behavior for C-Plane for DL data, C-Plane for UL data, and U-Plane for DL data is equal. Thus, there is common usage of T12_min and T12_max parameters.
- The transmission window (T1a_max – T1a_min) for C-Plane for DL data, C-Plane for UL data, and U-Plane for DL data all have the same length.
- The reception window (T2a_max – T2a_min) for C-Plane for DL data, C-Plane for UL data, and U-Plane for DL data all have the same length.
- T2a_min_cp_ul: Latest availability at O-RU of C-Plane for UL data message before reception of the first IQ data sample of the respective user's U-Plane UL data packet is received over the air interface.

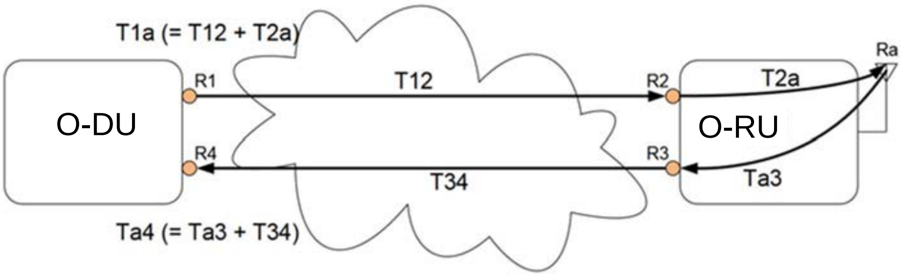


Figure B.1 : Definition of reference points for delay management (adapted from [2])

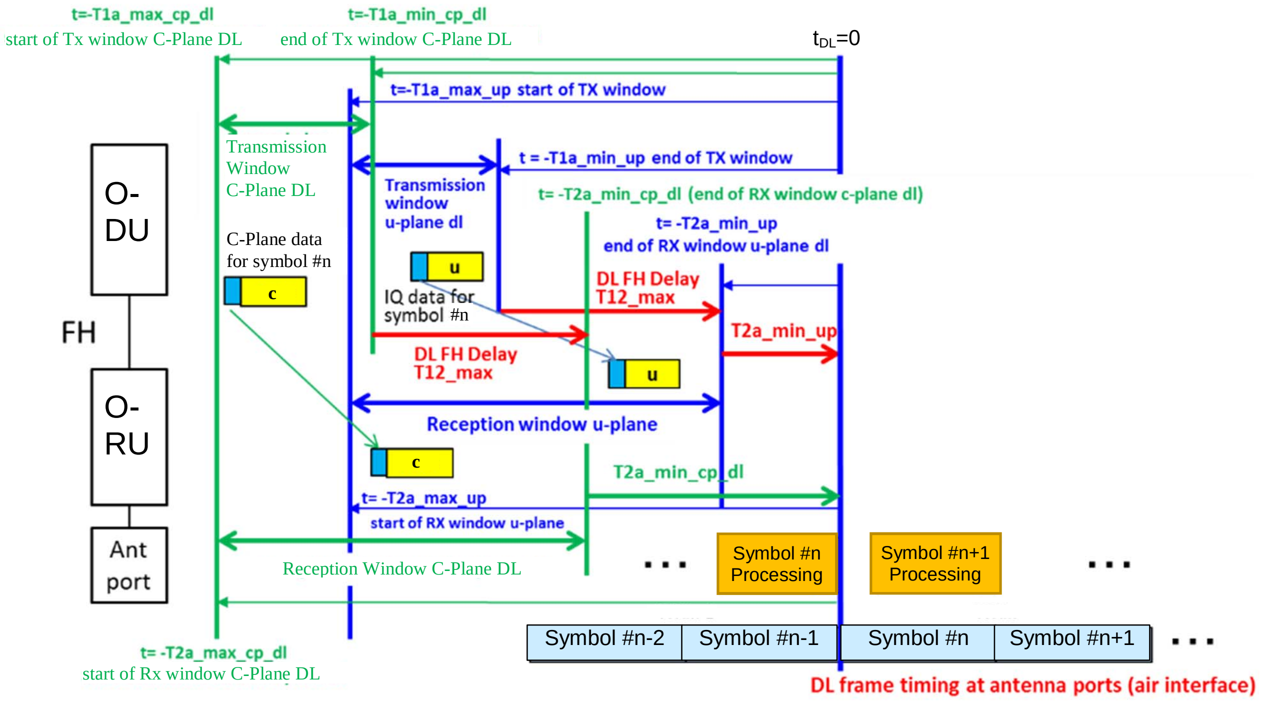

t=-T1a_max_cp_dl
start of Tx window C-Plane DL
t=-T1a_min_cp_dl
end of Tx window C-Plane DL
tDL=0
t=-T1a_max_up start of TX window
Transmission Window C-Plane DL
t = -T1a_min_up end of TX window
Transmission window u-plane dl
t= -T2a_min_cp_dl (end of RX window c-plane dl)
t= -T2a_min_up
end of RX window u-plane dl
O-DU
C-Plane data for symbol #n
u
c
IQ data for symbol #n
DL FH Delay T12_max
FH
DL FH Delay T12_max
T2a_min_up
u
O-RU
Reception window u-plane
c
T2a_min_cp_dl
t= -T2a_max_up
start of RX window u-plane
Ant port
Reception Window C-Plane DL
Symbol #n Processing
Symbol #n+1 Processing
Symbol #n-2
Symbol #n-1
Symbol #n
Symbol #n+1
t= -T2a_max_cp_dl
start of Rx window C-Plane DL
DL frame timing at antenna ports (air interface)

Figure B.2 : Timing relations per symbol IQ in DL direction (U-Plane and C-Plane)

Figure B.2 is based on the eCPRI delay measurement model on timing relations in DL direction. More detail is added to illustrate the following data transfer timing relations:

- U-plane DL data : Blue path
- C-plane for DL data : Green path

To understand this timing diagram, it is easier to work backward in timeline.

For U-plane DL data flow (use symbol #0 transmission as an example):

- t = 0 : time of transmission (at air interface) of the first sample for symbol #0 (see $t_{DL} = 0$)
- t = - T2a_min_up : O-RU has a fixed data processing delay (T2a_min_up). In order to meet air interface time for symbol#0 transmission at t = 0, symbol#0 data shall be presented on time for the processing unit (yellow block).
  - For symbol#0, start of processing time t = 0 – T2a_min_up = - T2a_min_up
- End of reception window : The latest time that O-RU can accept U-plane DL data for a specific symbol prior to start of data processing. If U-plane DL data arrives earlier than this time and is within reception window range, DL data may wait inside reception window buffer until the start of processing time, hence - the end of reception window is the same time point as the start of processing time as described immediately above.
  - For symbol#0, end of reception window time t = 0 – T2a_min_up = - T2a_min_up.
- Start of reception window : This earliest time that O-RU can accept U-plane DL data for a specific symbol prior to start of data processing. If U-plane DL data arrives later than this time and is within reception window range, DL data will wait inside reception window buffer.
  - For symbol#0, start of reception window time t = 0 - T2a_max_up = - T2a_max_up
- O-RU reception window range = T2a_max_up – T2a_min_up
- End of transmission window : The latest time that O-DU can send U-plane DL data for a specific symbol out to transport interface.
  - For symbol#0, end of transmission window time t = 0 – T1a_min_up = - T1a_min_up.
- Start of transmission window : The earliest time that O-DU can send U-plane DL for a specific symbol out to transport interface.
  - For symbol#0, start of transmission window time t = 0 – T1a_max_up = - T1a_max_up.
- O-DU transmission window range = T1a_max_up – T1a_min_up
- DL FH transport delay : T12_min and T12_max is the min and max transport delay. When FH transport is behaving properly, it should guarantee each packet transport (carrying C-plane or U-plane DL packet) delay is within this min and max boundary. Figure B.2 illustrates case where T12_min = 0.
- O-DU transmission window, FH transport delay and O-RU reception window shall satisfy the following criteria for proper behavior:
  - Start of reception window can accept a packet sent at start of transmission window AND experienced min FH transport delay => - T2a_max_up < -T1a_max_up + T12_min ---- (eq 1)
  - End of reception window can accept a packet sent at end of transmission window AND experienced max FH transport delay => - T2a_min_up > -T1a_min_up + T12_max ---- (eq 2)
  - Combining both criteria (-eq1 + eq2), the logical conclusion is as follows:
    - **O-RU reception window range > O-DU transmission window + FH DL transport max-min**
    - **(T2a_max_up – T2a_min_up) > (T1a_max_up – T1a_min_up) + (T12_max – T12_min)**

In other words, the reception window at O-RU MUST be greater than or equal to the total of the O-DU transmission variability and the Transport time variability.

For C-plane to support DL data flow (use symbol #n = 0 transmission as an example):

- Focus on the green path, same principle is applied to relate reception window, transmission window and FH transport delay.
- the logical conclusion is as follows:
  - **Reception window range > Transmission window + FH DL transport max-min**
  - **(T2a_max_cp_dl – T2a_min_cp_dl) > (T1a_max_cp_dl – T1a_min_cp_dl) + (T12_max – T12_min)**

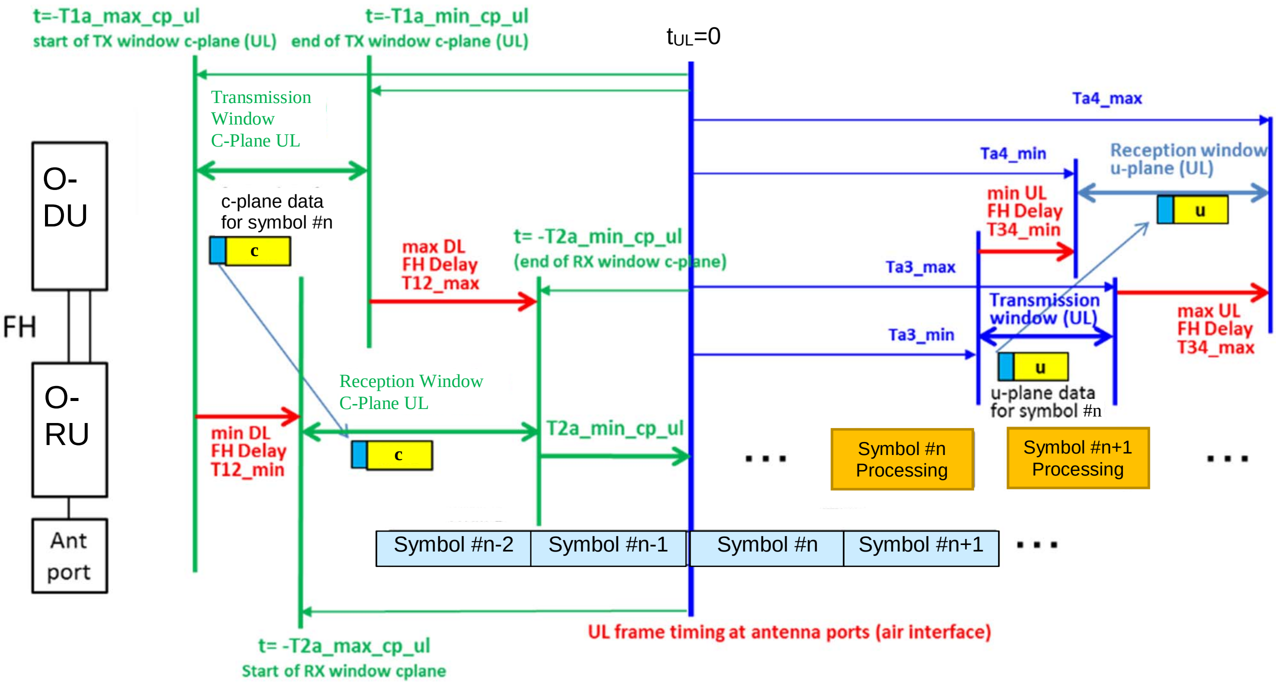


Figure B.3 : Timing relations per symbol IQ in UL direction (U-Plane and C-Plane)

Figure B.3 is based on the eCPRI delay measurement model on timing relations in UL direction. More detail is added to illustrate the following data transfer timing relations:

- U-plane UL data : Blue path
- C-plane for UL data : Green path

To understand this timing diagram, it is easier to work forward in timeline for the U-plane UL data and work backward in timeline for the C-plane to support UL data flow. Note this example is applicable for non-PRACH channel.

For U-plane UL data flow (use symbol #n = 0 reception as an example):

- t = 0 : time of reception (at air interface) of the first sample for symbol #0 (see t_{UL} = 0)
- t = 0 + Ta3_min_up : O-RU has a fixed data processing delay (Ta3_min_up). Air interface data is immediately presented to data processing unit (yellow block).
  - For symbol#0, end of processing = 0 + Ta3_min_up = Ta3_min_up
- Start of transmission window : The earliest time that O-RU can send U-plane UL for a specific symbol out to transport interface. The earliest time is immediately after data processing.

  - For symbol#0, start of transmission window = 0 + Ta3_min_up = Ta3_min_up.

- End of transmission window : The latest time that O-RU can send U-plane UL data for a specific symbol out to transport interface.
  - For symbol#0, end of transmission window = 0 + Ta3_max_up = Ta3_max_up.
- O-RU transmission window range = Ta3_max_up – Ta3_min_up
- Start of Reception window : This earliest time that O-DU can accept U-plane UL data for a specific symbol prior to start of O-DU data processing. If U-plane DL data arrives later than this time and is within reception window range, UL data will wait inside reception window buffer.
  - For symbol#0, start of reception window = 0 + Ta4_min = Ta4_min
- End of Reception window : The latest time that O-DU can accept U-plane UL data for a specific symbol prior to start of O-DU data processing.
  - For symbol#0, end of reception window = 0 + Ta4_max = Ta4_max.
- O-DU reception window range = Ta4_max – Ta4_min
- UL FH Transport delay : T34_min and T34_max is the min and max transport delay. When FH transport is behaving properly, it should guarantee each packet transport (carrying U-plane UL packet) delay is within this min and max boundary.
- Transmission window, FH transport delay and reception window shall satisfy the following criteria for proper behavior:
  - Start of reception window can accept a packet sent at start of transmission window AND experienced min FH transport delay => Ta4_min < Ta3_min_up + T34_min ---- (eq 1)
  - End of reception window can accept a packet sent at end of transmission window AND experienced max FH transport delay => Ta4_max > Ta3_max_up + T34_max ---- (eq 2)
  - Combining both criteria (-eq1 + eq2), the logical conclusion is as follows:
    - ▪ **O-DU reception window range > O-RU transmission window + FH UL transport max-min**
    - ▪ **(Ta4_max – Ta4_min) > (Ta3_max_up – Ta3_min_up) + (T34_max – T34_min)**

In other words, the reception window at O-DU MUST be greater than or equal to the total of the O-RU transmission variability and the Transport time variability.

For C-plane to support UL data flow (use symbol #0 transmission as an example):

- Focus on the green path, same principle to C-plane to support DL data is applied to relate reception window, transmission window and FH transport delay.
- the logical conclusion is as follows:
  - **Reception window range > Transmission window + FH transport min-max difference**
  - **(T2a_max_cp_ul – T2a_min_cp_ul) > (T1a_max_cp_ul – T1a_min_cp_ul) + (T12_max – T12_min)**

It is useful to consider several different delay management use cases because delay management, in particular the need to buffer large amounts of data within O-RUs, can have a significant cost and complexity impact on O-RU design. More specifically, accommodating long network delays (allowing long fiber lengths and/or many switch hops) can impose a significant buffering requirement on the O-RU while use cases involving short network delays e.g. an in-building application may allow much smaller in-O-RU buffering thereby allowing a lower-cost and lower-power design. Additionally, the SCS may affect the transmission windows described above, and thus use cases may be SCS specific as well. Note that here it is assumed the DL buffering will be done in the O-RU, otherwise the O-DU would have to always implement extensive buffering to accommodate any O-RU use case which would increase overall costs.

In the case of using pre-defined network latency values per use case, the following shall be pre-defined (for measured-network-latency cases these same values are determined via the measurement process):

T12_min, T12_max, T34_min and T34_max shall be determined based on the desired network configuration. Determination of T12_min and T34_min values shall include shortest transmission paths, both fiber and minimum switching delays. In addition, longest fibers, switching delays, and PDV introduced due to the variable delay nature of ethernet shall be accounted for in T12_max and T34_max.

T2a_min_up, T2a_min_cp_ul, and Ta3_min shall be determined across all O-RU equipment to be used in the system. The determined values shall be greater than or equal to the largest of these values across all supported equipment. Equipment with lower values need to have additional delay added to align with the determined values.

Similarly, the maximum transmission windows shall be determined across all O-DU and O-RU equipment. As stated above, the transmission windows for DL UP, DL CP and UL CP are all assumed to be the same at current. However, this is the maximum transmission window. Equipment may use less time for transmission of any symbol.

Additionally, the advance between DL UP/ CP shall be determined. (Tcp_adv_dl)

Once the above parameters are determined, the remaining values may be calculated as shown in the following tables:

Downlink Data Direction

Table B.1 : U-Plane DL delay boundaries

Downlink	Method
T1a_max_up	≤ T12_min + T2a_max_up
T1a_min_up	≥ T12_max + T2a_min_up
T2a_max_up	≥ T2a_min_up + (T12_max – T12_min) + O-DU Transmission Window
T2a_min_up	Specified per Use Case
T12_max	Specified per Use Case
T12_min	Specified per Use Case
O-DU Transmission Window	Specified per Use Case

Table B.2 : C-Plane DL delay boundaries

Downlink	Method
Tcp_adv_dl	Specified per Use Case
T1a_max_cp_dl	≥T1a_max_up + Tcp_adv_dl
T1a_min_cp_dl	T1a_min_up + Tcp_adv_dl
T2a_max_cp_dl	≥T2a_max_up + Tcp_adv_dl
T2a_min_cp_dl	T2a_min_up + Tcp_adv_dl
T12_max	Same as U-plane DL
T12_min	Same as U-plane DL

Uplink Data Direction (need not be the same as the Downlink values)

Table B.3 : U-Plane Uplink delay boundaries

Uplink	Method
Ta3 max	≤ Ta3_min + O-RU Transmission Window
Ta3 min	Specified per Use Case
Ta4 max	≥ Ta3_max + T34_max
Ta4 min	≤ Ta3_min + T34_min
T34 max	Specified per Use Case
T34 min	Specified per Use Case
O-RU Transmission Window	Specified per Use Case

Table B.4 : C-Plane Uplink delay boundaries

Downlink	Method
T1a_max_cp_ul	≤ T12_min + T2a_max_cp_ul
T1a_min_cp_ul	≥ T12_max + T2a_min_cp_ul
T2a_max_cp_ul	≥ T2a_min_cp_ul + (T12_max – T12_min) + O-DU uplink C-Plane Transmission Window duration
T2a_min_cp_ul	Specified per Use Case
T12_max	Specified per Use Case
T12_min	Specified per Use Case
O-DU Transmission Window	Specified per Use Case

B.2 Latency categories and sub-categories

The O-DU and O-RU are defined to be in “delay categories” and “delay sub-categories” for the purposes of allowing a matching of O-DU and O-RU units that will operate properly together from the point of view of accommodating a customer’s network delay. Network delay comprises the “time-of-flight” of signals through (typically) a fiber-optic cable (so can be known from the fiber length) added to the signal traversal latency through any switches in the network.

The delay category and delay sub-category values depend in part on the processing latency within the O-DU and O-RU. It may be expected especially for an O-RU that the processing latency may depend on the specific frequency band and sub-carrier spacing that is used. Further, a multi-band radio may experience different processing latencies for its different bands. Therefore, it may be expected that an O-RU (and perhaps more rarely an O-DU) will have different delay category and delay sub-category ratings for different bands.

O-DU categories are defined as A.##[##] (per direction, i.e. O-DU UL and O-DU DL are different categories)

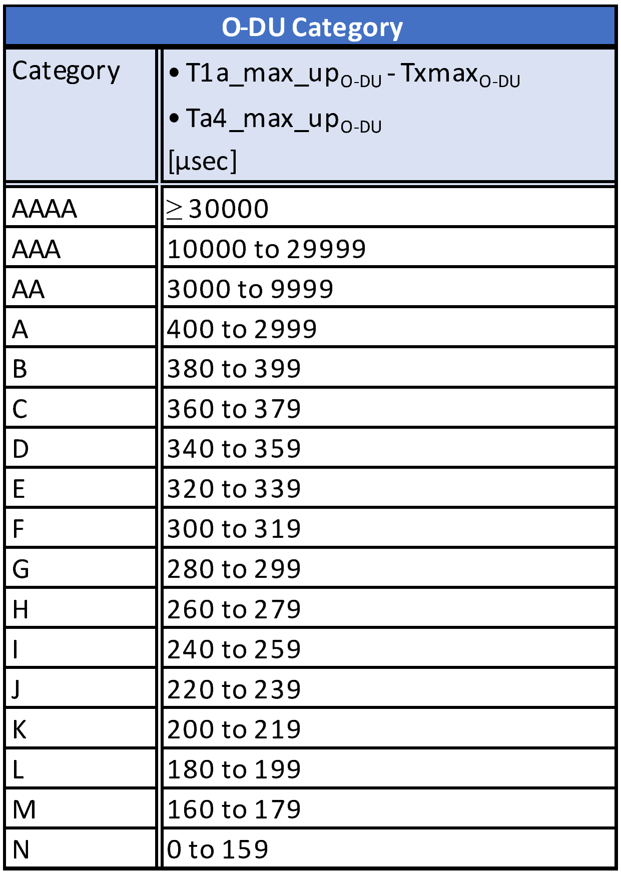
- Prefix(blank/f): The prefix distinguishes the type of O-DU with dynamic/fixed timing advance/retard
- A: Category from [AAAA-N] per table below
- .##[##]: sub-category from [.00 - .1000] per tables below

O-RU categories are defined as X.##[##] (per direction, i.e. O-RU UL and O-RU DL are different categories)

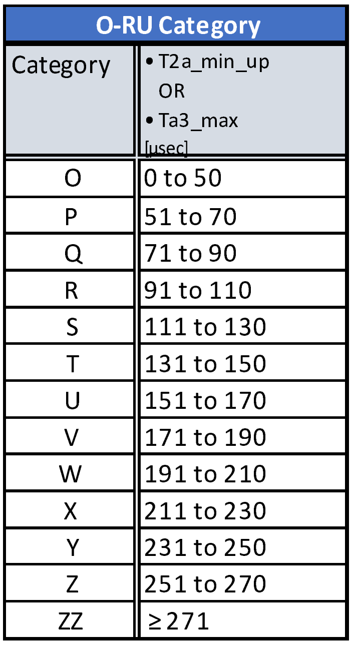
- X: Category from [O-ZZ] per table below
- .##[##]: sub-category from [.00-.1000] per tables below

The following tables are used by equipment vendors to assign categories to their equipment based on design characteristics.

Table B.5 : O-DU and O-RU Delay Categories



O-DU Category	
Category	• T1a_max_up$_{O\text{-}DU}$ - Txmax$_{O\text{-}DU}$ • Ta4_max_up$_{O\text{-}DU}$ [µsec]
AAAA	≥ 30000
AAA	10000 to 29999
AA	3000 to 9999
A	400 to 2999
B	380 to 399
C	360 to 379
D	340 to 359
E	320 to 339
F	300 to 319
G	280 to 299
H	260 to 279
I	240 to 259
J	220 to 239
K	200 to 219
L	180 to 199
M	160 to 179
N	0 to 159



O-RU Category	
Category	• T2a_min_up OR • Ta3_max [µsec]
O	0 to 50
P	51 to 70
Q	71 to 90
R	91 to 110
S	111 to 130
T	131 to 150
U	151 to 170
V	171 to 190
W	191 to 210
X	211 to 230
Y	231 to 250
Z	251 to 270
ZZ	≥ 271

NOTE: Categories are defined to group endpoints with similar delay characteristics for easy evaluation relative to use cases. Specifically, categories AA-AAAA and ZZ are defined for non-ideal fronthaul. The calculated value for O-DU or O-RU falling anywhere within the range for the category indicates that the endpoint is classified as that category. It is NOT required that the endpoint be able to support the full range of the category.

Table B.6 and B.7 are intended for use by network providers to determine the best and worst case T12_max/ T34_max values that can be supported by a given equipment combination (excluding non-ideal fronthaul). Alternatively, network providers may locate the desired T12_max/ T34_max and select from the equipment combinations meeting that criteria. Common criteria are identified by different colors on diagonals through the tables

Table B.6 : Latency_min (Minimum supported T12_max/ T34_max in µsec)

	O-DU													
RU	A	B	C	D	E	F	G	H	I	J	K	L	M	N
O	350	330	310	290	270	250	230	210	190	170	150	130	110	0
P	330	310	290	270	250	230	210	190	170	150	130	110	90	0
Q	310	290	270	250	230	210	190	170	150	130	110	90	70	0
R	290	270	250	230	210	190	170	150	130	110	90	70	50	0
S	270	250	230	210	190	170	150	130	110	90	70	50	30	0
T	250	230	210	190	170	150	130	110	90	70	50	30	10	0
U	230	210	190	170	150	130	110	90	70	50	30	10	0	0
V	210	190	170	150	130	110	90	70	50	30	10	0	0	0
W	190	170	150	130	110	90	70	50	30	10	0	0	0	0
X	170	150	130	110	90	70	50	30	10	0	0	0	0	0
Y	150	130	110	90	70	50	30	10	0	0	0	0	0	0
Z	130	110	90	70	50	30	10	0	0	0	0	0	0	0

Table B.7 : Latency_max (Maximum supported T12_max/ T34_max in µsec)

	O-DU													
RU	A	B	C	D	E	F	G	H	I	J	K	L	M	N
O	3000	399	379	359	339	319	299	279	259	239	219	199	179	159
P	2949	348	328	308	288	268	248	228	208	188	168	148	128	108
Q	2929	328	308	288	268	248	228	208	188	168	148	128	108	88
R	2909	308	288	268	248	228	208	188	168	148	128	108	88	68
S	2889	288	268	248	228	208	188	168	148	128	108	88	68	48
T	2869	268	248	228	208	188	168	148	128	108	88	68	48	28
U	2849	248	228	208	188	168	148	128	108	88	68	48	28	8
V	2829	228	208	188	168	148	128	108	88	68	48	28	8	0
W	2809	208	188	168	148	128	108	88	68	48	28	8	0	0
X	2789	188	168	148	128	108	88	68	48	28	8	0	0	0
Y	2769	168	148	128	108	88	68	48	28	8	0	0	0	0
Z	2749	148	128	108	88	68	48	28	8	0	0	0	0	0

Table B.8 : Use Case Mapping (Based on Latency_min)

Range (KM)	T12max T34max (µsec)	Minimum T12Max Guaranteed for Category Combinations (Combinations may support larger T12max)										
50	250	AO - AT	BO - BS	CO - CR	DO - DQ	EO - EP	FO					
45	225	AU	BT	CS	DR	EQ	FP	GO				
40	200	AV	BU	CT	DS	ER	FQ	GP	HO			
35	175	AW	BV	CU	DT	ES	FR	GQ	HP	IO		
30	150	AX - AY	BW - BX	CV - CW	DU - DV	ET - EU	FS - FT	GR - GS	HQ - HR	IP - IQ	JO - JP	KO		
25	125	AZ	BY	CX	DW	EV	FU	GT	HS	IR	JQ	KP	LO	
20	100		BZ	CY	DX	EW	FV	GU	HT	IS	JR	KQ	LP	MO
15	75			CZ	DY	EX	FW	GV	HU	IT	JS	KR	LQ	MP
10	50				DZ	EY - EZ	FX - FY	GW - GX	HV - HW	IU - IV	JT - JU	KS - KT	LR - LS	MQ - MR
5	25						FZ	GY	HX	IW	JV	KU	LT	MS
0	0							GZ	HY	IX	JW	KV	LU	MT

NOTE: The Use Case Mapping table can be used to determine the O-DU/ O-RU delay category combinations which are guaranteed to support a given latency use case. This is defined based on Latency_min for the category. It may be possible for equipment from a lower category to support a higher use case, but cannot be guaranteed. To determine the exact maximum latency for an O-DU/ O-RU pair, it is necessary to calculate based on the delay parameter values for the paired equipment.

Table B.9 : O-DU and O-RU Tx/Rx Window Sub-Categories

Receive Window Sub-Category	
Sub-Category	• T2a_max_up - T2a_min_up • RXmax$_{O\text{-}DU}$ [in usec]
.1000	≥10000
.300	3000 to 9999
.100	1000 to 2999
.40	400 to 999
.38	380 to 399
.36	360 to 379
.34	340 to 359
.32	320 to 339
.30	300 to 319
.28	280 to 299
.26	260 to 279
.24	240 to 259
.22	220 to 239
.20	200 to 219
.18	180 to 199
.16	160 to 179
.14	140 to 159
.12	120 to 139
.10	100 to 119
.08	80 to 99
.06	60 to 79
.04	40 to 59
.02	20 to 39
.01	10 to 19
.00	0 to 9

Transmit Window Sub-Category	
Sub-Category	• TXmax$_{O\text{-}DU}$ • Ta3_max - Ta3_min [in usec]
.20	≥200
.19	190 to 199
.18	180 to 189
.17	170 to 179
.16	160 to 169
.15	150 to 159
.14	140 to 149
.13	130 to 139
.12	120 to 129
.11	110 to 119
.10	100 to 109
.09	90 to 99
.08	80 to 89
.07	70 to 79
.06	60 to 69
.05	50 to 59
.04	40 to 49
.03	30 to 39
.02	20 to 29
.01	10 to 19
.00	0 to 9

NOTE: Sub-categories are defined to group endpoints with similar delay characteristics for easy evaluation relative to use cases. Specifically, sub-categories .100 to .1000 are defined for non-ideal fronthaul. The calculated value for O-DU or O-RU falling anywhere within the range for the category indicates that the endpoint is classified as that sub-category. It is NOT required that the endpoint be able to support the full range of the sub-category.

Table B.10 is used by service providers to identify equipment sub-category combinations which meet the desired network variability (excluding non-ideal fronthaul). Variability is shown in Km in the table. This range is based on 5 usec per Km.

Table B.10 : Dynamic Range (in Km) for sub-category pairs

	Receive Window																				
TxMax	.40	.38	.36	.34	.32	.30	.28	.26	.24	.22	.20	.18	.16	.14	.12	.10	.08	.06	.04	.02	.01
.20	40	36	32	28	24	20	16	12	8	4											
.19	42	38	34	30	26	22	18	14	10	6	2										
.18	44	40	36	32	28	24	20	16	12	8	4										
.17	46	42	38	34	30	26	22	18	14	10	6	2									
.16	48	44	40	36	32	28	24	20	16	12	8	4									
.15	50	46	42	38	34	30	26	22	18	14	10	6	2								
.14	52	48	44	40	36	32	28	24	20	16	12	8	4								
.13	54	50	46	42	38	34	30	26	22	18	14	10	6	2							
.12	56	52	48	44	40	36	32	28	24	20	16	12	8	4							
.11	58	54	50	46	42	38	34	30	26	22	18	14	10	6	2						
.10	60	56	52	48	44	40	36	32	28	24	20	16	12	8	4						
.09	62	58	54	50	46	42	38	34	30	26	22	18	14	10	6	2					
.08	64	60	56	52	48	44	40	36	32	28	24	20	16	12	8	4					
.07	66	62	58	54	50	46	42	38	34	30	26	22	18	14	10	6	2				
.06	68	64	60	56	52	48	44	40	36	32	28	24	20	16	12	8	4				
.05	70	66	62	58	54	50	46	42	38	34	30	26	22	18	14	10	6	2			
.04	72	68	64	60	56	52	48	44	40	36	32	28	24	20	16	12	8	4			
.03	74	70	66	62	58	54	50	46	42	38	34	30	26	22	18	14	10	6	2		
.02	76	72	68	64	60	56	52	48	44	40	36	32	28	24	20	16	12	8	4		
.01	78	74	70	66	62	58	54	50	46	42	38	34	30	26	22	18	14	10	6	2	
.00	80	76	72	68	64	60	56	52	48	44	40	36	32	28	24	20	16	12	8	4	2

B.3 Example Case: Evaluating O-DU / O-RU Combinations

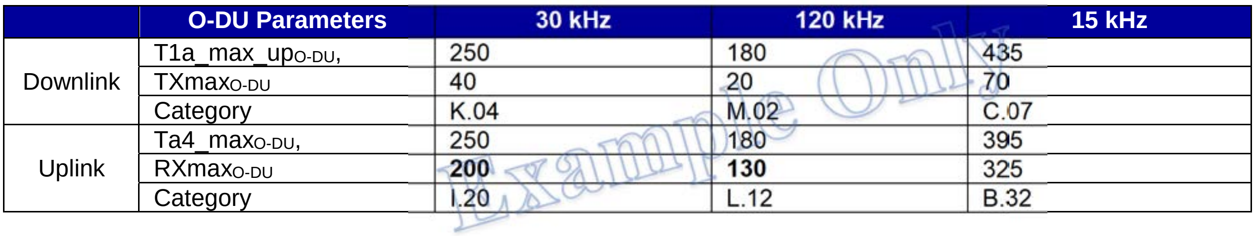
This example illustrates an O-RU and an O-DU, each supporting multiple SCS. It illustrates how the actual O-DU/ O-RU delay profiles map to categories, and then how the corresponding categories are used to evaluate the resulting transport network limitations. The values represented are exemplary only. The example is provided to illustrate:

- Interoperability across multiple SCS
  - O-RU and O-DU will have multiple delay profiles
  - Delay profiles are different for uplink and downlink
- applicability service provider use case

Table B.11 : Delay profiles for the example O-RU

	O-RU Parameters	30 kHz	120 kHz		15 kHz
Downlink	T2a_min_up,	**50**	**50**		**205**
	T2a_max_up,	250	180		435
	T2a_min_cp_dl,	175	175		330
	T2a_max_cp_dl,	375	205		460
	Tcp_adv_dl	125	125		125
	Category	O.20 (250-50)=200	O.12		W.22
Uplink	Ta3_min,	50	50		70
	Ta3_max	100	70		235
	T2a_min_cp_ul,	125	125		125
	T2a_max_cp_ul	325	255		360
	Category	R.05	P.02		Y.16

Table B.12 : Delay profiles for the example O-DU.



	O-DU Parameters	30 kHz	120 kHz	15 kHz
Downlink	T1a_max_up$_{O-DU}$,	250	180	435
	TXmax$_{O-DU}$	40	20	70
	Category	K.04	M.02	C.07
Uplink	Ta4_max$_{O-DU}$,	250	180	395
	RXmax$_{O-DU}$	**200**	**130**	325
	Category	I.20	L.12	B.32

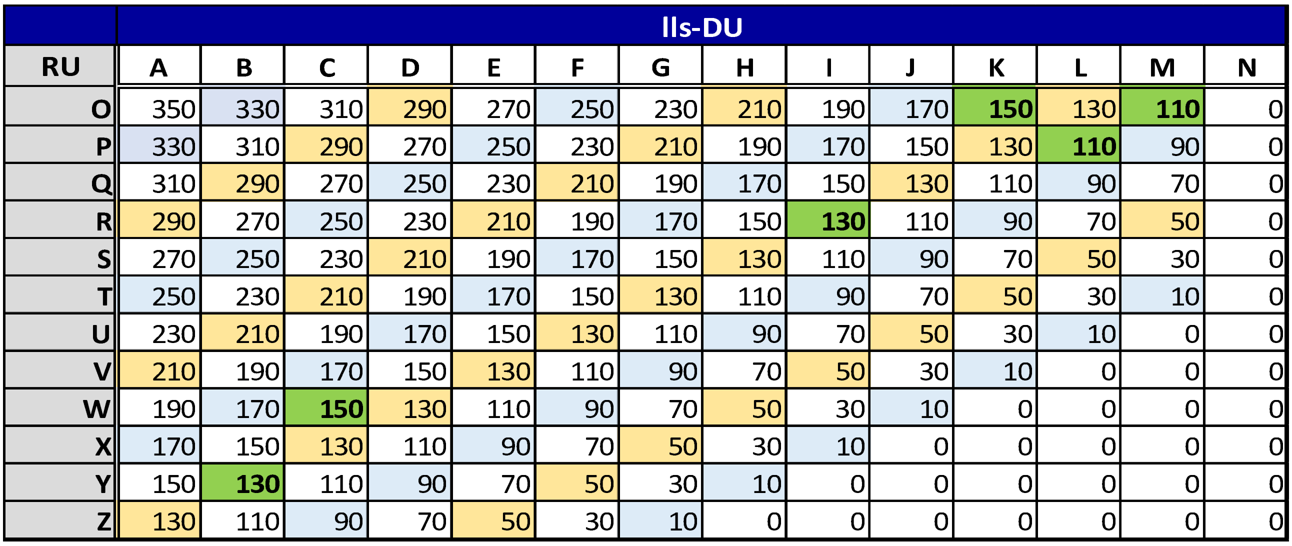
This results in 6 different category combinations:

Table B.13 : Resulting 6 different category combinations

	30 kHz	120 kHz	15 kHz
Downlink	KO	MO	CW
Uplink	IR	LP	BY

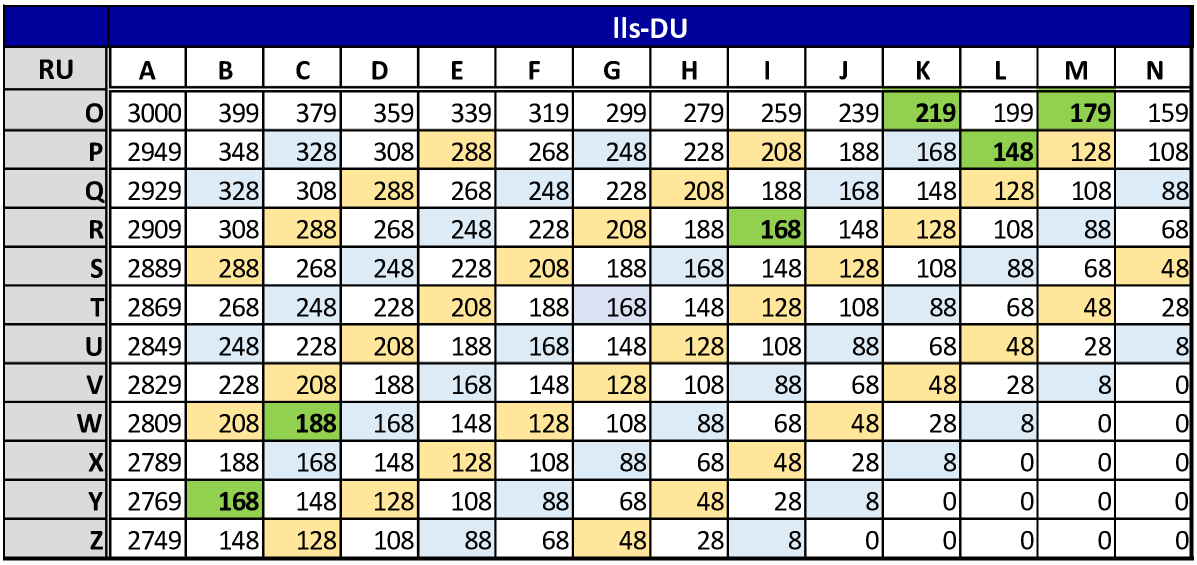
The respective minimum and maximum T12max values are highlighted in the tables below. Using the 30 kHz as an example, it can be seen that the range on the transport is limited by the uplink (IR) to between 130 usec and 168 usec. This means that this combination can be guaranteed to be able to support at least 130 usec of delay, and may be capable of supporting up to 168 usec of delay. Using the actual delay values for the combination at 30 kHz, the uplink is limited to Ta4_max – Ta3_max = 250 – 100 = 150 usec. (The downlink value is higher, so uplink becomes the limiting factor for this combination.)

Table B.14 : Delay Category O-DU and O-RU with highlighted valid options for this example, minimum T12max



	lls-DU													
RU	A	B	C	D	E	F	G	H	I	J	K	L	M	N
O	350	330	310	290	270	250	230	210	190	170	**150**	130	**110**	0
P	330	310	290	270	250	230	210	190	170	150	130	**110**	90	0
Q	310	290	270	250	230	210	190	170	150	130	110	90	70	0
R	290	270	250	230	210	190	170	150	**130**	110	90	70	50	0
S	270	250	230	210	190	170	150	130	110	90	70	50	30	0
T	250	230	210	190	170	150	130	110	90	70	50	30	10	0
U	230	210	190	170	150	130	110	90	70	50	30	10	0	0
V	210	190	170	150	130	110	90	70	50	30	10	0	0	0
W	190	170	**150**	130	110	90	70	50	30	10	0	0	0	0
X	170	150	130	110	90	70	50	30	10	0	0	0	0	0
Y	150	**130**	110	90	70	50	30	10	0	0	0	0	0	0
Z	130	110	90	70	50	30	10	0	0	0	0	0	0	0

Table B.15 : Delay Category O-DU and O-RU with highlighted valid options for this example, maximum T12max

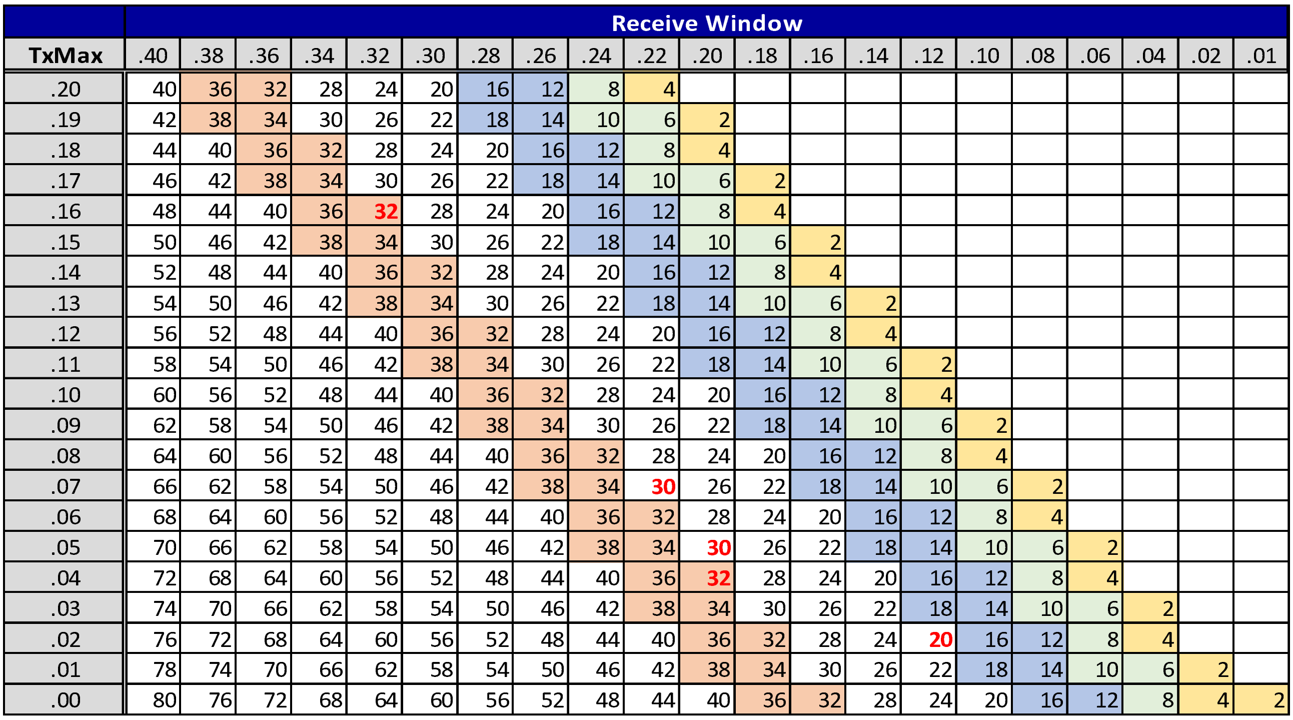


	lls-DU													
RU	A	B	C	D	E	F	G	H	I	J	K	L	M	N
O	3000	399	379	359	339	319	299	279	259	239	**219**	199	**179**	159
P	2949	348	328	308	288	268	248	228	208	188	168	**148**	128	108
Q	2929	328	308	288	268	248	228	208	188	168	148	128	108	88
R	2909	308	288	268	248	228	208	188	**168**	148	128	108	88	68
S	2889	288	268	248	228	208	188	168	148	128	108	88	68	48
T	2869	268	248	228	208	188	168	148	128	108	88	68	48	28
U	2849	248	228	208	188	168	148	128	108	88	68	48	28	8
V	2829	228	208	188	168	148	128	108	88	68	48	28	8	0
W	2809	208	**188**	168	148	128	108	88	68	48	28	8	0	0
X	2789	188	168	148	128	108	88	68	48	28	8	0	0	0
Y	2769	**168**	148	128	108	88	68	48	28	8	0	0	0	0
Z	2749	148	128	108	88	68	48	28	8	0	0	0	0	0

Evaluating the overall combination across all SCS, it can be seen that the maximum possible range for this combination if all 3 SCS are to be used is 148 usec (limited by the 120 kHz uplink) and the minimum guaranteed range is 110 usec. (In actuality, the limit is 110 usec for this combination.)

Similarly, there are 6 combinations of sub-categories to evaluate. (In this case, the sub-category combinations happen to be the same for 120 kHz uplink and downlink.) In this case the different between T12max and T12min is limited by the 120 kHz delay profile, with a maximum range of ~20 KM (~100 usec).

Table B.16 : Delay Sub-Category O-DU and O-RU with highlighted (in red) valid options for this example



	Receive Window																				
TxMax	.40	.38	.36	.34	.32	.30	.28	.26	.24	.22	.20	.18	.16	.14	.12	.10	.08	.06	.04	.02	.01
.20	40	36	32	28	24	20	16	12	8	4											
.19	42	38	34	30	26	22	18	14	10	6	2										
.18	44	40	36	32	28	24	20	16	12	8	4										
.17	46	42	38	34	30	26	22	18	14	10	6	2									
.16	48	44	40	36	**32**	28	24	20	16	12	8	4									
.15	50	46	42	38	34	30	26	22	18	14	10	6	2								
.14	52	48	44	40	36	32	28	24	20	16	12	8	4								
.13	54	50	46	42	38	34	30	26	22	18	14	10	6	2							
.12	56	52	48	44	40	36	32	28	24	20	16	12	8	4							
.11	58	54	50	46	42	38	34	30	26	22	18	14	10	6	2						
.10	60	56	52	48	44	40	36	32	28	24	20	16	12	8	4						
.09	62	58	54	50	46	42	38	34	30	26	22	18	14	10	6	2					
.08	64	60	56	52	48	44	40	36	32	28	24	20	16	12	8	4					
.07	66	62	58	54	50	46	42	38	34	**30**	26	22	18	14	10	6	2				
.06	68	64	60	56	52	48	44	40	36	32	28	24	20	16	12	8	4				
.05	70	66	62	58	54	50	46	42	38	34	**30**	26	22	18	14	10	6	2			
.04	72	68	64	60	56	52	48	44	40	36	**32**	28	24	20	16	12	8	4			
.03	74	70	66	62	58	54	50	46	42	38	34	30	26	22	18	14	10	6	2		
.02	76	72	68	64	60	56	52	48	44	40	36	32	28	24	**20**	16	12	8	4		
.01	78	74	70	66	62	58	54	50	46	42	38	34	30	26	22	18	14	10	6	2	
.00	80	76	72	68	64	60	56	52	48	44	40	36	32	28	24	20	16	12	8	4	2

Using only the O-DU/ O-RU category/ sub-category combinations, if the corresponding O-DU/ O-RU is to be used across all 3 SCS, the resulting delay constraints limit the implementation to:

- 110 usec ≤ T12max ≤ 148 usec
- T12min = T12max – 100 usec

The service provider can then use this analysis to determine if the resulting combination suits their target use case.

B.4 Example Case: Non-ideal transport O-DU/O-RU Categories

This example illustrates the usage of the non-ideal transport delay categories and sub-categories included in Table B.5 and Table B.9 respectively. These categories and sub-categories were created to allow for the use of transport networks which have delay introduced at the MAC layer that is larger than the propagation delay e.g. DOCSIS networks, E/GPON networks, microwave links, or G.Fast networks. In such systems, delay is decoupled from the transport network length and therefore better specified in microseconds or milliseconds.

Non-ideal transport is defined by 3GPP in document TR 36.932 [i.2] per **Table B**.17**.**

Table B.17 : Non-Ideal transport categorization from 3GGP TR 36.932 [i.2] Table 6.1-1

Backhaul Technology	Latency (One way)	Throughput	Priority (1 is the highest)
Fiber Access 1	10-30ms	10M-10Gbps	1
Fiber Access 2	5-10ms	100-1000Mbps	2
Fiber Access 3	2-5ms	50M-10Gbps	1
DSL Access	15-60ms	10-100 Mbps	1
Cable	25-35ms	10-100 Mbps	2
Wireless Backhaul	5-35ms	10Mbps – 100Mbps typical, maybe up to Gbps range	1

In **Table B**.18, example deployment use cases are used to illustrate O-DU/O-RU delay profile mapping. The values, while meant to representative, are exemplary only.

Table B.18 : Non-Ideal fronthaul O-DU/O-RU Delay Profiles by link type

		DOCSIS		Microwave		PON		G.Fast	
		T12 (DL)	T34 (UL)	T12 (DL)	T34 (UL)	T12 (DL)	T34 (UL)	T12 (DL)	T34 (UL)
	Latency (RTT)	< 1 ms	< 29 ms	< 10 ms	<10 ms	< 500 µs	< 2.5 ms	< 2 ms	< 2 ms
	Jitter/PDV	< 320 µs	< 9 ms	< 2.5 ms	<2.5 ms	< 1 ms	<1 ms	< 1 ms	< 1 ms
O-DU	T1a_max_up_O-DU	1,500 µs	-	12,500 µs	-	750 µs	-	2,250 µs	-
	TXmax_O-DU	50 µs	-	100 µs	-	50 µs	-	50 µs	-
	Ta4_max_O-DU	-	29,500 µs	-	12,500 µs	-	2,750 µs	-	2,250 µs
	Rx_max_O-DU	-	9,500 µs	-	3,250 µs	-	1,250 µs	-	1,250 µs
	Category	**A.05**	**AAA.300**	**AAA.10**	**AAA.300**	**A.05**	**A.100**	**A.05**	**A.100**
O-RU	T2a_min_up	100 µs	-	50 µs	-	50 µs	-	70 µs	-
	T2a_max_up	500 µs	-	2,650 µs	-	1,100 µs	-	1,200 µs	-
	Ta3_min	-	50 µs	-	50 µs	-	100 µs	-	70 µs
	Ta3_max	-	150 µs	-	200 µs	-	200 µs	-	200 µs
	Category	**R.40**	**T.10**	**O.100**	**W.15**	**O.40**	**W.10**	**P.100**	**W.13**

The use of non-ideal transport links for fronthaul traffic comes with system performance tradeoffs. See Appendix L for further discussion of these considerations.

Annex C (informative): M-Plane Impacts

The CUS-Plane makes certain demands on the M-Plane as listed below (list not exhaustive):

1. **Generic static O-RU configuration:** there are many parameters e.g. frequency band, number of antennas, power level, etc. that will need to be gathered from the O-RU and configured by the O-DU. These are expected to be much the same as is currently experienced with existing radio modules.
2. **O-RU management:** this includes status monitoring, KPI measurements (PM counters), alarm collection and software download. This is expected to be handled much the same as with existing radio modules.
3. **Compression:** command to use static method (compression and IQ bit width) or use dynamic method using udCompHdr.
4. **Rtcid**: M-Plane provides the bit-widths for the four defined fields (shall sum to 16 bits total).
5. **Synch state**: it is expected the O-RU will report its sync state to the EMS via the M-Plane.
6. **Synch**: it is expected the EMS via the M-Plane will convey to the O-RU the clock quality being received (or does this come directly from the GM?)
7. **BeamId format**: to accommodate hybrid BF, and maybe even for other cases, the O-RU shall convey via M-Plane characteristics of the O-RU so that the O-DU can “know” how to generate BF weights.
8. **PRB raster and offset-setting**: The M-Plane needs to convey the minimum PRB raster (based on the minimum SCS, for LTE generally 15 kHz) to allow PRB counting across multiple SCS values (to support mixed-numerology channels). In addition, the M-Plane shall convey the offset to the zeroth PRB.
9. **Beam-weights / beam attributes**: It is intended that the M-Plane can download beam weights or beam attributes to an O-RU when weight updating does not need to be real-time. The number of weights or types of attributes applicable to the O-RU is meant to be conveyed from the O-RU to the O-DU at start-up via M-plane messaging.
10. **Power-Efficiency**: the M-Plane will very likely include commands to enable O-RU power-saving techniques which may be vendor-dependent.
11. **Delay-Management and Transport Priority**: for each eAxC (Pcid), whether the UL data is delay-managed or not, and what the transport priority should be.

Annex D (informative): IQ Sample and Exponent Packetization for Different Bitwidths

D.1 General

Bit-ordering and packetization for I and Q samples and compression parameters follows patterns that repeat after every transmission of 12 resource elements. At this point, the pattern repeats starting from the udCompParam information element, and is followed by the I and Q samples for the next 12 resource elements.

The cells in the following Tables indicate the bit ordering for the following IEs

- udCompParam (assumed to be one byte in the tables)
- I samples denoted by $I_{btwidth-1}\ldots I_0$
- Q samples denoted by $Q_{bitwidth-1}\ldots Q_0$

Table D.1. IQ data samples bit-ordering (6-bit IQ bitwidth example)

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
…									
udCompParam								1	Octet N
I_5	I_4	I_3	I_2	I_1	I_0	Q_5	Q_4	1	N+1
Q_3	Q_2	Q_1	Q_0	…				1	N+2
…									
			…	I_5	I_4	I_3	I_2	1	N+17
I_1	I_0	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	N+18
udCompParam								1	Octet M
I_5	I_4	I_3	I_2	I_1	I_0	Q_5	Q_4	1	M+1
Q_3	Q_2	Q_1	Q_0	…				1	M+2
…									
			…	I_5	I_4	I_3	I_2	1	M+17
I_1	I_0	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	M+18
…									

Table D.2. IQ data samples bit-ordering (7-bit IQ bitwidth example)

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
…									
udCompParam								1	Octet N
I_6	I_5	I_4	I_3	I_2	I_1	I_0	Q_6	1	N+1
Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	…		1	N+2
…									
	…	I_6	I_5	I_4	I_3	I_2	I_1	1	N+20
I_0	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	N+21
udCompParam								1	Octet M
I_6	I_5	I_4	I_3	I_2	I_1	I_0	Q_6	1	M+1
Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	…		1	M+2
…									
	…	I_6	I_5	I_4	I_3	I_2	I_1	1	M+20
I_0	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	M+21
…									

Table D.3. IQ data samples bit-ordering (8-bit IQ bitwidth example)

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
…									
udCompParam								1	Octet N
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	1	N+1
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	N+2
…									
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	1	N+23
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	N+24
udCompParam								1	Octet M
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	1	M+1
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	M+2
…									
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	1	M+23
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	M+24
…									

Table D.4. IQ data samples bit-ordering (9-bit IQ bitwidth example)

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
…									
udCompParam								1	Octet N
I_8	I_7	I_6	I_5	I_4	I_3	I_2	I_1	1	N+1
I_0	Q_8	Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	1	N+2
Q_1	Q_0	…						1	N+3
…									
					…	I_8	I_7	1	N+25
I_6	I_5	I_4	I_3	I_2	I_1	I_0	Q_8	1	N+26
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	N+27
udCompParam								1	Octet M
I_8	I_7	I_6	I_5	I_4	I_3	I_2	I_1	1	M+1
I_0	Q_8	Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	1	M+2
Q_1	Q_0	…						1	
…									
					…	I_8	I_7	1	M+25
I_6	I_5	I_4	I_3	I_2	I_1	I_0	Q_8	1	M+26
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	M+27
…									

Table D.5. IQ data samples bit-ordering (10-bit IQ bitwidth example)

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
…									
udCompParam								1	Octet N
I_9	I_8	I_7	I_6	I_5	I_4	I_3	I_2	1	N+1
I_1	I_0	Q_9	Q_8	Q_7	Q_6	Q_5	Q_4	1	N+2
Q_3	Q_2	Q_1	Q_0	…				1	N+3
…									
			…	I_9	I_8	I_7	I_6	1	N+28
I_5	I_4	I_3	I_2	I_1	I_0	Q_9	Q_8	1	N+29
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	N+30
udCompParam								1	Octet M
I_9	I_8	I_7	I_6	I_5	I_4	I_3	I_2	1	M+1
I_1	I_0	Q_9	Q_8	Q_7	Q_6	Q_5	Q_4	1	M+2
Q_3	Q_2	Q_1	Q_0	…				1	M+3
…									
			…	I_9	I_8	I_7	I_6	1	M+28
I_5	I_4	I_3	I_2	I_1	I_0	Q_9	Q_8	1	M+29
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	M+30
…									

Table D.6. IQ data samples bit-ordering (11-bit IQ bitwidth example)

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
…									
udCompParam								1	Octet N
I_{10}	I_9	I_8	I_7	I_6	I_5	I_4	I_3	1	N+1
I_2	I_1	I_0	Q_{10}	Q_9	Q_8	Q_7	Q_6	1	N+2
Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	…		1	N+3
…									
	…	I_{10}	I_9	I_8	I_7	I_6	I_5	1	N+31
I_4	I_3	I_2	I_1	I_0	Q_{10}	Q_9	Q_8	1	N+32
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	N+33
udCompParam								1	Octet M
I_{10}	I_9	I_8	I_7	I_6	I_5	I_4	I_3	1	M+1
I_2	I_1	I_0	Q_{10}	Q_9	Q_8	Q_7	Q_6	1	M+2
Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	…		1	M+3
…									
	…	I_{10}	I_9	I_8	I_7	I_6	I_5	1	M+31
I_4	I_3	I_2	I_1	I_0	Q_{10}	Q_9	Q_8	1	M+32
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	M+33
…									

Table D.7. IQ data samples bit-ordering (12-bit IQ bitwidth example)

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
…									
udCompParam								1	Octet N
I_{11}	I_{10}	I_9	I_8	I_7	I_6	I_5	I_4	1	N+1
I_3	I_2	I_1	I_0	Q_{11}	Q_{10}	Q_9	Q_8	1	N+2
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	N+3
…									
I_{11}	I_{10}	I_9	I_8	I_7	I_6	I_5	I_4	1	N+34
I_3	I_2	I_1	I_0	Q_{11}	Q_{10}	Q_9	Q_8	1	N+35
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	N+36
udCompParam								1	Octet M
I_{11}	I_{10}	I_9	I_8	I_7	I_6	I_5	I_4	1	M+1
I_3	I_2	I_1	I_0	Q_{11}	Q_{10}	Q_9	Q_8	1	M+2
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	M+3
…									
I_{11}	I_{10}	I_9	I_8	I_7	I_6	I_5	I_4	1	M+34
I_3	I_2	I_1	I_0	Q_{11}	Q_{10}	Q_9	Q_8	1	M+35
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	M+36
…									

Table D.8. IQ data samples bit-ordering (13-bit IQ bitwidth example)

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
…									
udCompParam								1	Octet N
I_{12}	I_{11}	I_{10}	I_9	I_8	I_7	I_6	I_5	1	N+1
I_4	I_3	I_2	I_1	I_0	Q_{12}	Q_{11}	Q_{10}	1	N+2
Q_9	Q_8	Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	1	N+3
Q_1	Q_0	…						1	N+4
…									
					…	I_{12}	I_{11}	1	N+36
I_{10}	I_9	I_8	I_7	I_6	I_5	I_4	I_3	1	N+37
I_2	I_1	I_0	Q_{12}	Q_{11}	Q_{10}	Q_9	Q_8	1	N+38
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	N+39
udCompParam								1	Octet M
I_{12}	I_{11}	I_{10}	I_9	I_8	I_7	I_6	I_5	1	M+1
I_4	I_3	I_2	I_1	I_0	Q_{12}	Q_{11}	Q_{10}	1	M+2
Q_9	Q_8	Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	1	M+3
Q_1	Q_0	…						1	M+4
…									
					…	I_{12}	I_{11}	1	M+36
I_{10}	I_9	I_8	I_7	I_6	I_5	I_4	I_3	1	M+37
I_2	I_1	I_0	Q_{12}	Q_{11}	Q_{10}	Q_9	Q_8	1	M+38
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	M+39
…									

Table D.9. IQ data samples bit-ordering (14-bit IQ bitwidth example)

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
…									
udCompParam								1	Octet N
I_{13}	I_{12}	I_{11}	I_{10}	I_{9}	I_{8}	I_{7}	I_{6}	1	N+1
I_{5}	I_{4}	I_{3}	I_{2}	I_{1}	I_{0}	Q_{13}	Q_{12}	1	N+2
Q_{11}	Q_{10}	Q_{9}	Q_{8}	Q_{7}	Q_{6}	Q_{5}	Q_{4}	1	N+3
Q_{3}	Q_{2}	Q_{1}	Q_{0}	…				1	N+4
…									
…				I_{13}	I_{12}	I_{11}	I_{10}	1	N+39
I_{9}	I_{8}	I_{7}	I_{6}	I_{5}	I_{4}	I_{3}	I_{2}	1	N+40
I_{1}	I_{0}	Q_{13}	Q_{12}	Q_{11}	Q_{10}	Q_{9}	Q_{8}	1	N+41
Q_{7}	Q_{6}	Q_{5}	Q_{4}	Q_{3}	Q_{2}	Q_{1}	Q_{0}	1	N+42
udCompParam								1	Octet M
I_{13}	I_{12}	I_{11}	I_{10}	I_{9}	I_{8}	I_{7}	I_{6}	1	M+1
I_{5}	I_{4}	I_{3}	I_{2}	I_{1}	I_{0}	Q_{13}	Q_{12}	1	M+2
Q_{11}	Q_{10}	Q_{9}	Q_{8}	Q_{7}	Q_{6}	Q_{5}	Q_{4}	1	M+3
Q_{3}	Q_{2}	Q_{1}	Q_{0}	…				1	M+4
…									
…				I_{13}	I_{12}	I_{11}	I_{10}	1	M+39
I_{9}	I_{8}	I_{7}	I_{6}	I_{5}	I_{4}	I_{3}	I_{2}	1	M+40
I_{1}	I_{0}	Q_{13}	Q_{12}	Q_{11}	Q_{10}	Q_{9}	Q_{8}	1	M+41
Q_{7}	Q_{6}	Q_{5}	Q_{4}	Q_{3}	Q_{2}	Q_{1}	Q_{0}	1	M+42
…									

Table D.10. IQ data samples bit-ordering (15-bit IQ bitwidth example)

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
…									
udCompParam								1	Octet N
I_{14}	I_{13}	I_{12}	I_{11}	I_{10}	I_{9}	I_{8}	I_{7}	1	N+1
I_{6}	I_{5}	I_{4}	I_{3}	I_{2}	I_{1}	I_{0}	Q_{14}	1	N+2
Q_{13}	Q_{12}	Q_{11}	Q_{10}	Q_{9}	Q_{8}	Q_{7}	Q_{6}	1	N+3
Q_{5}	Q_{4}	Q_{3}	Q_{2}	Q_{1}	Q_{0}	…		1	N+4
…									
…		I_{14}	I_{13}	I_{12}	I_{11}	I_{10}	I_{9}	1	N+42
I_{8}	I_{7}	I_{6}	I_{5}	I_{4}	I_{3}	I_{2}	I_{1}	1	N+43
I_{0}	Q_{14}	Q_{13}	Q_{12}	Q_{11}	Q_{10}	Q_{9}	Q_{8}	1	N+44
Q_{7}	Q_{6}	Q_{5}	Q_{4}	Q_{3}	Q_{2}	Q_{1}	Q_{0}	1	N+45
udCompParam								1	Octet M
I_{14}	I_{13}	I_{12}	I_{11}	I_{10}	I_{9}	I_{8}	I_{7}	1	M+1
I_{6}	I_{5}	I_{4}	I_{3}	I_{2}	I_{1}	I_{0}	Q_{14}	1	M+2
Q_{13}	Q_{12}	Q_{11}	Q_{10}	Q_{9}	Q_{8}	Q_{7}	Q_{6}	1	M+3
Q_{5}	Q_{4}	Q_{3}	Q_{2}	Q_{1}	Q_{0}	…		1	M+4
…									
…		I_{14}	I_{13}	I_{12}	I_{11}	I_{10}	I_{9}	1	M+42
I_{8}	I_{7}	I_{6}	I_{5}	I_{4}	I_{3}	I_{2}	I_{1}	1	M+43
I_{0}	Q_{14}	Q_{13}	Q_{12}	Q_{11}	Q_{10}	Q_{9}	Q_{8}	1	M+44
Q_{7}	Q_{6}	Q_{5}	Q_{4}	Q_{3}	Q_{2}	Q_{1}	Q_{0}	1	M+45
…									

Table D.11. IQ data samples bit-ordering (16-bit IQ bitwidth example)

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
…									
udCompParam								1	Octet N
I_{15}	I_{14}	I_{13}	I_{12}	I_{11}	I_{10}	I_9	I_8	1	N+1
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	1	N+2
Q_{15}	Q_{14}	Q_{13}	Q_{12}	Q_{11}	Q_{10}	Q_9	Q_8	1	N+3
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	N+4
…									
I_{15}	I_{14}	I_{13}	I_{12}	I_{11}	I_{10}	I_9	I_8	1	N+45
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	1	N+46
Q_{15}	Q_{14}	Q_{13}	Q_{12}	Q_{11}	Q_{10}	Q_9	Q_8	1	N+47
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	N+48
udCompParam								1	Octet M
I_{15}	I_{14}	I_{13}	I_{12}	I_{11}	I_{10}	I_9	I_8	1	M+1
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	1	M+2
Q_{15}	Q_{14}	Q_{13}	Q_{12}	Q_{11}	Q_{10}	Q_9	Q_8	1	M+3
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	M+4
…									
I_{15}	I_{14}	I_{13}	I_{12}	I_{11}	I_{10}	I_9	I_8	1	M+45
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	1	M+46
Q_{15}	Q_{14}	Q_{13}	Q_{12}	Q_{11}	Q_{10}	Q_9	Q_8	1	M+47
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	M+48
…									

D.2 IQ Sample for Little Endian Byte Order

When the optional “little endian byte order” format is used, the complex IQ data order is changed relative to that shown in Annex D.1. The Q data will be sent before I data, while within IQ data, the low byte is sent before high byte. All other conventions are observed same as (Annex D.1) big endian format. The complex data fields which are applied to the “little endian byte order” are identified in related clauses. The usage of “little endian byte ordering” can be negotiated between the O-DU and O-RU via the M-Plane.

The cells in the following Tables indicate the bit ordering for the following IEs

- I samples denoted by $I_{btwidth-1}$…I_0
- Q samples denoted by $Q_{bitwidth-1}$…Q_0

Table D.12. Bit order of IQ data samples in little endian mode (6-bit IQ bit width example)

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
…									
									Octet N
Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	I_5	I_4	1	N+1
I_3	I_2	I_1	I_0	…				1	N+2
…									
			…	Q_5	Q_4	Q_3	Q_2	1	N+17
Q_1	Q_0	I_5	I_4	I_3	I_2	I_1	I_0	1	N+18
0	0	0	0	E_3	E_2	E_1	E_0	1	Octet M
Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	I_5	I_4	1	M+1
I_3	I_2	I_1	I_0	…				1	M+2
…									
			…	Q_5	Q_4	Q_3	Q_2	1	M+17
Q_1	Q_0	I_5	I_4	I_3	I_2	I_1	I_0	1	M+18
…									

Table D.13. Bit order of IQ data samples in little endian mode (7-bit IQ bitwidth example)

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
…									
									Octet N
Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	I_6	1	N+1
I_5	I_4	I_3	I_2	I_1	I_0	…		1	N+2
…									
	…	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	1	N+20
Q_0	I_6	I_5	I_4	I_3	I_2	I_1	I_0	1	N+21
0	0	0	0	E_3	E_2	E_1	E_0	1	Octet M
Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	I_6	1	M+1
I_5	I_4	I_3	I_2	I_1	I_0	…		1	M+2
…									
	…	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	1	M+20
Q_0	I_6	I_5	I_4	I_3	I_2	I_1	I_0	1	M+21
…									

Table D.14. Bit order of IQ data samples in little endian mode (8-bit IQ bit width example)

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
…									
									Octet N
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	N+1
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	1	N+2
…									
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	N+23
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	1	N+24
0	0	0	0	E_3	E_2	E_1	E_0	1	Octet M
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	M+1
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	1	M+2
…									
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	M+23
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	1	M+24
…									

Table D.15. Bit order of IQ data samples in little endian mode (9-bit IQ bitwidth example)

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
…									
									Octet N
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	N+1
Q_8	I_7	I_6	I_5	I_4	I_3	I_2	I_1	1	N+2
I_0	I_8	…						1	N+3
…									
					…	Q_7	Q_6	1	N+25
Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	Q_8	I_7	1	N+26
I_6	I_5	I_4	I_3	I_2	I_1	I_0	I_8	1	N+27
0	0	0	0	E_3	E_2	E_1	E_0	1	Octet M
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	M+1
Q_8	I_7	I_6	I_5	I_4	I_3	I_2	I_1	1	M+2
I_0	I_8	…						1	
…									
					…	Q_7	Q_6	1	M+25
Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	Q_8	I_7	1	M+26
I_6	I_5	I_4	I_3	I_2	I_1	I_0	I_8	1	M+27
…									

Table D.16. Bit order of IQ data samples in little endian mode (10-bit IQ bitwidth example)

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
…									
									Octet N
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	N+1
Q_9	Q_8	I_7	I_6	I_5	I_4	I_3	I_2	1	N+2
I_1	I_0	I_9	I_8	…				1	N+3
…									
			…	Q_7	Q_6	Q_5	Q_4	1	N+28
Q_3	Q_2	Q_1	Q_0	Q_9	Q_8	I_7	I_6	1	N+29
I_5	I_4	I_3	I_2	I_1	I_0	I_9	I_8	1	N+30
0	0	0	0	E_3	E_2	E_1	E_0	1	Octet M
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	M+1
Q_9	Q_8	I_7	I_6	I_5	I_4	I_3	I_2	1	M+2
I_1	I_0	I_9	I_8	…				1	M+3
…									
			…	Q_7	Q_6	Q_5	Q_4	1	M+28
Q_3	Q_2	Q_1	Q_0	Q_9	Q_8	I_7	I_6	1	M+29
I_5	I_4	I_3	I_2	I_1	I_0	I_9	I_8	1	M+30
…									

Table D.17. Bit order of IQ data samples in little endian mode (11-bit IQ bitwidth example)

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
…									
									Octet N
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	N+1
Q_{10}	Q_9	Q_8	I_7	I_6	I_5	I_4	I_3	1	N+2
I_2	I_1	I_0	I_{10}	I_9	I_8	…		1	N+3
…									
	…	Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	1	N+31
Q_1	Q_0	Q_{10}	Q_9	Q_8	I_7	I_6	I_5	1	N+32
I_4	I_3	I_2	I_1	I_0	I_{10}	I_9	I_8	1	N+33
0	0	0	0	E_3	E_2	E_1	E_0	1	Octet M
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	M+1
Q_{10}	Q_9	Q_8	I_7	I_6	I_5	I_4	I_3	1	M+2
I_2	I_1	I_0	I_{10}	I_9	I_8	…		1	M+3
…									
	…	Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	1	M+31
Q_1	Q_0	Q_{10}	Q_9	Q_8	I_7	I_6	I_5	1	M+32
I_4	I_3	I_2	I_1	I_0	I_{10}	I_9	I_8	1	M+33
…									

Table D.18. Bit order of IQ data samples in little endian mode (12-bit IQ bitwidth example)

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
…									
									Octet N
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	N+1
Q_{11}	Q_{10}	Q_9	Q_8	I_7	I_6	I_5	I_4	1	N+2
I_3	I_2	I_1	I_0	I_{11}	I_{10}	I_9	I_8	1	N+3
…									
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	N+34
Q_{11}	Q_{10}	Q_9	Q_8	I_7	I_6	I_5	I_4	1	N+35
I_3	I_2	I_1	I_0	I_{11}	I_{10}	I_9	I_8	1	N+36
0	0	0	0	E_3	E_2	E_1	E_0	1	Octet M
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	M+1
Q_{11}	Q_{10}	Q_9	Q_8	I_7	I_6	I_5	I_4	1	M+2
I_3	I_2	I_1	I_0	I_{11}	I_{10}	I_9	I_8	1	M+3
…									
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	M+34
Q_{11}	Q_{10}	Q_9	Q_8	I_7	I_6	I_5	I_4	1	M+35
I_3	I_2	I_1	I_0	I_{11}	I_{10}	I_9	I_8	1	M+36
…									

Table D.19. Bit order of IQ data samples in little endian mode (13-bit bitwidth mantissa example)

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
…									
									Octet N
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	N+1
Q_{12}	Q_{11}	Q_{10}	Q_9	Q_8	I_7	I_6	I_5	1	N+2
I_4	I_3	I_2	I_1	I_0	I_{12}	I_{11}	I_{10}	1	N+3
I_9	I_8	…						1	N+4
…									
					…	Q_7	Q_6	1	N+36
Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	Q_{12}	Q_{11}	1	N+37
Q_{10}	Q_9	Q_8	I_7	I_6	I_5	I_4	I_3	1	N+38
I_2	I_1	I_0	I_{12}	I_{11}	I_{10}	I_9	I_8	1	N+39
0	0	0	0	E_3	E_2	E_1	E_0	1	Octet M
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	M+1
Q_{12}	Q_{11}	Q_{10}	Q_9	Q_8	I_7	I_6	I_5	1	M+2
I_4	I_3	I_2	I_1	I_0	I_{12}	I_{11}	I_{10}	1	M+3
I_9	I_8	…						1	M+4
…									
					…	Q_7	Q_6	1	M+36
Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	Q_{12}	Q_{11}	1	M+37
Q_{10}	Q_9	Q_8	I_7	I_6	I_5	I_4	I_3	1	M+38
I_2	I_1	I_0	I_{12}	I_{11}	I_{10}	I_9	I_8	1	M+39
…									

Table D.20. Bit order of IQ data samples in little endian mode (14-bit IQ bitwidth example)

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
…									
									Octet N
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	N+1
Q_{13}	Q_{12}	Q_{11}	Q_{10}	Q_9	Q_8	I_7	I_6	1	N+2
I_5	I_4	I_3	I_2	I_1	I_0	I_{13}	I_{12}	1	N+3
I_{11}	I_{10}	I_9	I_8	…				1	N+4
…									
…				Q_7	Q_6	Q_5	Q_4	1	N+39
Q_3	Q_2	Q_1	Q_0	Q_{13}	Q_{12}	Q_{11}	Q_{10}	1	N+40
Q_9	Q_8	I_7	I_6	I_5	I_4	I_3	I_2	1	N+41
I_1	I_0	I_{13}	I_{12}	I_{11}	I_{10}	I_9	I_8	1	N+42
0	0	0	0	E_3	E_2	E_1	E_0	1	Octet M
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	M+1
Q_{13}	Q_{12}	Q_{11}	Q_{10}	Q_9	Q_8	I_7	I_6	1	M+2
I_5	I_4	I_3	I_2	I_1	I_0	I_{13}	I_{12}	1	M+3
I_{11}	I_{10}	I_9	I_8	…				1	M+4
…									
…				Q_7	Q_6	Q_5	Q_4	1	M+39
Q_3	Q_2	Q_1	Q_0	Q_{13}	Q_{12}	Q_{11}	Q_{10}	1	M+40
Q_9	Q_8	I_7	I_6	I_5	I_4	I_3	I_2	1	M+41
I_1	I_0	I_{13}	I_{12}	I_{11}	I_{10}	I_9	I_8	1	M+42
…									

Table D.21. Bit order of IQ data samples in little endian mode (15-bit IQ bitwidth example)

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
…									
									Octet N
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	N+1
Q_{14}	Q_{13}	Q_{12}	Q_{11}	Q_{10}	Q_9	Q_8	I_7	1	N+2
I_6	I_5	I_4	I_3	I_2	I_1	I_0	I_{14}	1	N+3
I_{13}	I_{12}	I_{11}	I_{10}	I_9	I_8	…		1	N+4
…									
…		Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	1	N+42
Q_1	Q_0	Q_{14}	Q_{13}	Q_{12}	Q_{11}	Q_{10}	Q_9	1	N+43
Q_8	I_7	I_6	I_5	I_4	I_3	I_2	I_1	1	N+44
I_0	I_{14}	I_{13}	I_{12}	I_{11}	I_{10}	I_9	I_8	1	N+45
0	0	0	0	E_3	E_2	E_1	E_0	1	Octet M
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	M+1
Q_{14}	Q_{13}	Q_{12}	Q_{11}	Q_{10}	Q_9	Q_8	I_7	1	M+2
I_6	I_5	I_4	I_3	I_2	I_1	I_0	I_{14}	1	M+3
I_{13}	I_{12}	I_{11}	I_{10}	I_9	I_8	…		1	M+4
…									
…		Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	1	M+42
Q_1	Q_0	Q_{14}	Q_{13}	Q_{12}	Q_{11}	Q_{10}	Q_9	1	M+43
Q_8	I_7	I_6	I_5	I_4	I_3	I_2	I_1	1	M+44
I_0	I_{14}	I_{13}	I_{12}	I_{11}	I_{10}	I_9	I_8	1	M+45
…									

Table D.**22**. Bit order of **IQ data samples in little endian mode (16-bit IQ bit width example)**

0 (msb)	1	2	3	4	5	6	7 (lsb)	Number of Octets	
…									
									Octet N
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	N+1
Q_{15}	Q_{14}	Q_{13}	Q_{12}	Q_{11}	Q_{10}	Q_9	Q_8	1	N+2
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	1	N+3
I_{15}	I_{14}	I_{13}	I_{12}	I_{11}	I_{10}	I_9	I_8	1	N+4
…									
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	N+45
Q_{15}	Q_{14}	Q_{13}	Q_{12}	Q_{11}	Q_{10}	Q_9	Q_8	1	N+46
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	1	N+47
I_{15}	I_{14}	I_{13}	I_{12}	I_{11}	I_{10}	I_9	I_8	1	N+48
0	0	0	0	E_3	E_2	E_1	E_0	1	Octet M
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	M+1
Q_{15}	Q_{14}	Q_{13}	Q_{12}	Q_{11}	Q_{10}	Q_9	Q_8	1	M+2
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	1	M+3
I_{15}	I_{14}	I_{13}	I_{12}	I_{11}	I_{10}	I_9	I_8	1	M+4
…									
Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	1	M+45
Q_{15}	Q_{14}	Q_{13}	Q_{12}	Q_{11}	Q_{10}	Q_9	Q_8	1	M+46
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	1	M+47
I_{15}	I_{14}	I_{13}	I_{12}	I_{11}	I_{10}	I_9	I_8	1	M+48
…									

Annex E (informative): OFDM Phase Compensation

Consider the time interval $t_{start,l}^{\mu} \leq t < t_{start,l}^{\mu} + \left(N_u^{\mu} + N_{CP,l}^{\mu}\right)T_c$ as defined in clause 5.3.1 of [4], which corresponds to the transmission of the lth OFDM symbol, with baseband waveform $s_l^{(p,\mu)}(t)$.

Suppose that the transmitter performs up-conversion of the signal $s_l^{(p,\mu)}(t)$ to RF centre frequency f_0, and the receiver performs down-conversion from RF centre frequency f_1. Unlike LTE, it is possible that the transmitter and receiver have different centre frequencies, $f_0 \neq f_1$, which motivates the introduction of OFDM phase compensation in NR.

Ignoring channel distortions and noise for simplicity, the baseband signal at the receiver in the same time interval can be written

$$s_l^{(p,\mu)}(t) \cdot e^{j2\pi(f_0-f_1)t} \ = \ s_l^{(p,\mu)}(t) \cdot e^{j\theta_l} \cdot e^{j2\pi(f_0-f_1)\left(t-t_{CP,l}^{\mu}\right)}$$

where $\theta_l = 2\pi(f_0 - f_1)t_{CP,l}^{\mu}$ and $t_{CP,l}^{\mu} = t_{start,l}^{\mu} + N_{CP,l}^{\mu}T_c$ is the time at the end of the cyclic prefix of OFDM symbol l.

We can see from the equation above that each OFDM symbol is subjected to a phase shift θ_l, which can result in large phase changes between successive OFDM symbols. This may cause problems for receiver processes such as channel estimation and frequency error tracking, which expect the channel to vary smoothly between symbols.

The solution adopted by 3GPP in RAN WG1 meeting AH#18-01 [5][6] is to apply a phase pre-compensation term $e^{-j2\pi f_0 t_{CP,l}^{\mu}}$ at the transmitter and a phase post-compensation term $e^{j2\pi f_1 t_{CP,l}^{\mu}}$ at the receiver. These two terms together provide the required correction $e^{-j\theta_l}$. The phase pre-compensation requirement for the transmitter is captured in clause 5.4 of [1]. Note that the phase compensation depends only on the starting time of each OFDM symbol, and is common for all OFDM symbols transmitted using a given numerology μ, regardless of which NR physical channels they belong to, except for PRACH.

Annex F (informative): Beamforming Attributes Frame of Reference

The beamforming attributes involving pointing angle (bfAzPt and bfZePt) are defined following 3GPP 38.901 [i.1], clause 7.1. Two coordinate systems are defined, the Global Coordinate System (GCS) and local coordinate system (LCS). The GCS applies across multiple BS and UT locations, while the LCS applies to a single array antenna. The GCS is defined as shown below in Figure F.**1**. In the GCS, $\boldsymbol{\phi}$ refers to the azimuth angle and $\boldsymbol{\theta}$ refers to the zenith angle.

The LCS is defined by a 3-parameter rotation of the GCS. The rotation with parameters $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$ is shown in Figure F.**2** (left). Note that the rotation parameters are defined as follows:

- α is defined as the bearing angle (sector pointing angle)
- β is defined as the downtilt angle
- γ is defined as the slant angle

A two-dimensional array antenna is defined such that the x'-axis is broadside to the array antenna. If an antenna architecture such as in 3GPP 38.901 [i.1] clause 7.3 is used, the horizontal direction is defined as y' and the vertical direction is defined as z'.

All parameters which relate to “peak” refer to the principal beam pointing in a particular direction. For a given beam configuration, the peak direction is the angle corresponding to maximum gain. Thus, bfAzPt specifies the pointing angle of maximum gain.

The pointing parameter bfAzPt is defined as ϕ'. The pointing parameter bfZePt is defined as θ'. The angles are shown in Figure F.2 (right).

The beamwidth parameters bfAz3dd and bfZe3dd are defined as the as the angular widths at which the beam falls to 3 dB below the peak beam gain on both sides of the peak direction, in azimuth and zenith, respectively.

The sidelobe parameters bfAzSl and bfZeSl are defined as the suppression level of the highest sidelobe relative to the peak gain in the azimuth and zenith principal planes, respectively.

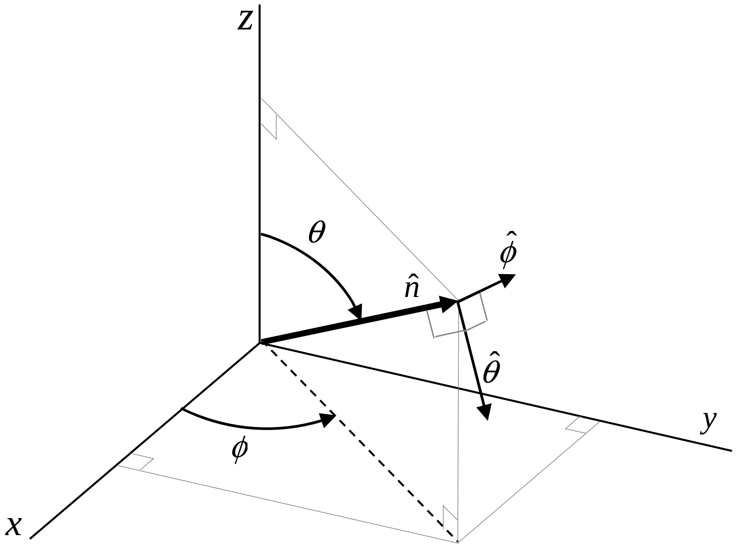


Figure F.1 : Global Coordinate System Definition

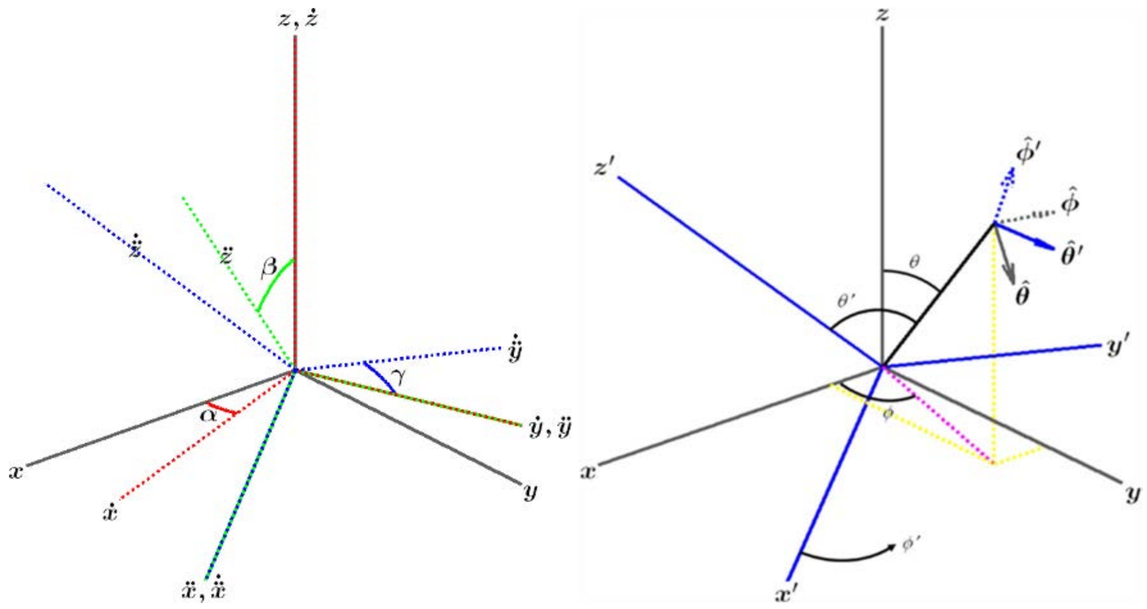


Figure F.2 : Orienting the LCS (blue) with respect to the GCS (gray) by a sequence of 3 rotations (left); Definition of spherical coordinates and unit vectors in both the GCS and LCS (right)

Annex G (informative): LAA Algorithms and Examples

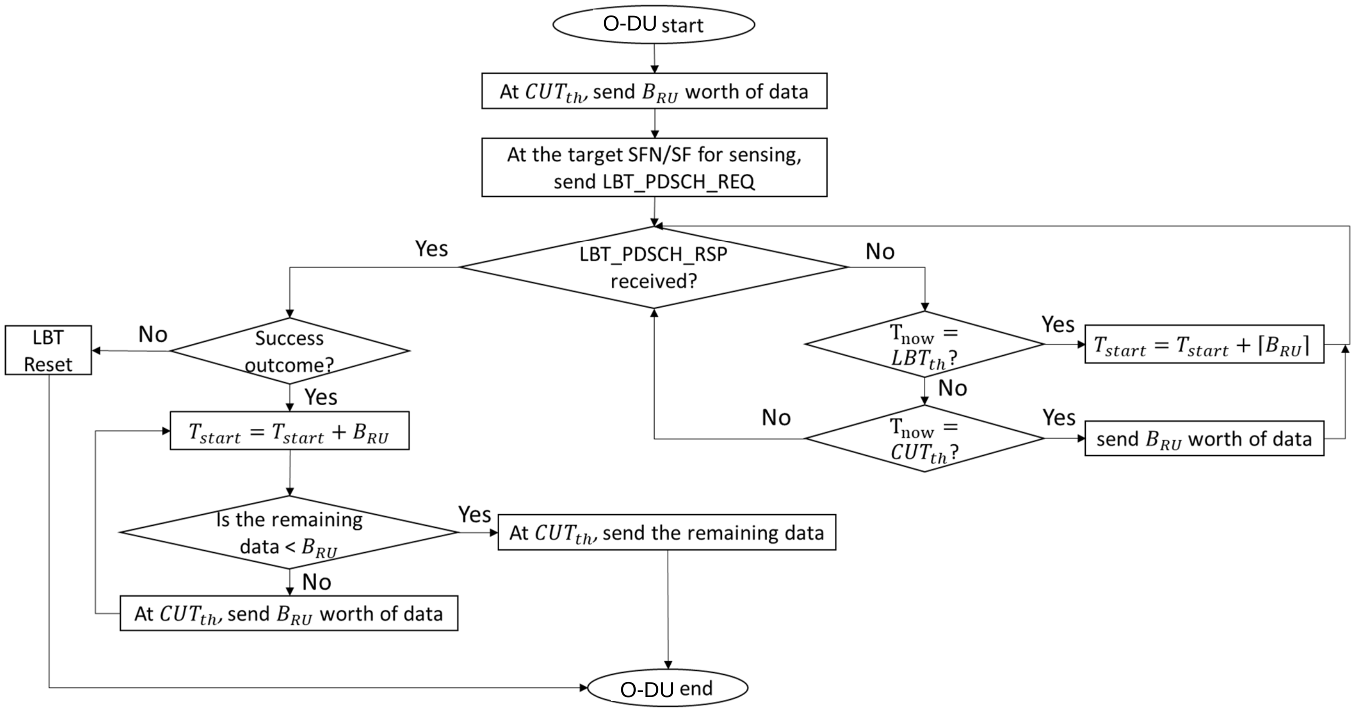

O-DU start
At CUT_{th}, send B_{RU} worth of data
At the target SFN/SF for sensing, send LBT_PDSCH_REQ
LBT_PDSCH_RSP received?
Yes
No
Success outcome?
No
LBT Reset
Yes
$T_{start} = T_{start} + B_{RU}$
Is the remaining data < B_{RU}
Yes
At CUT_{th}, send the remaining data
No
At CUT_{th}, send B_{RU} worth of data
$T_{now} = LBT_{th}$?
Yes
$T_{start} = T_{start} + \lceil B_{RU} \rceil$
No
$T_{now} = CUT_{th}$?
Yes
send B_{RU} worth of data
No
O-DU end

Figure G.1 : PDSCH Transmission Algorithm O-DU flowchart

where $\lceil a \rceil$: is the ceiling of number “a” to the nearest x, where x is 1 ms for normal SF and 0.5 ms for partially-filled SF.

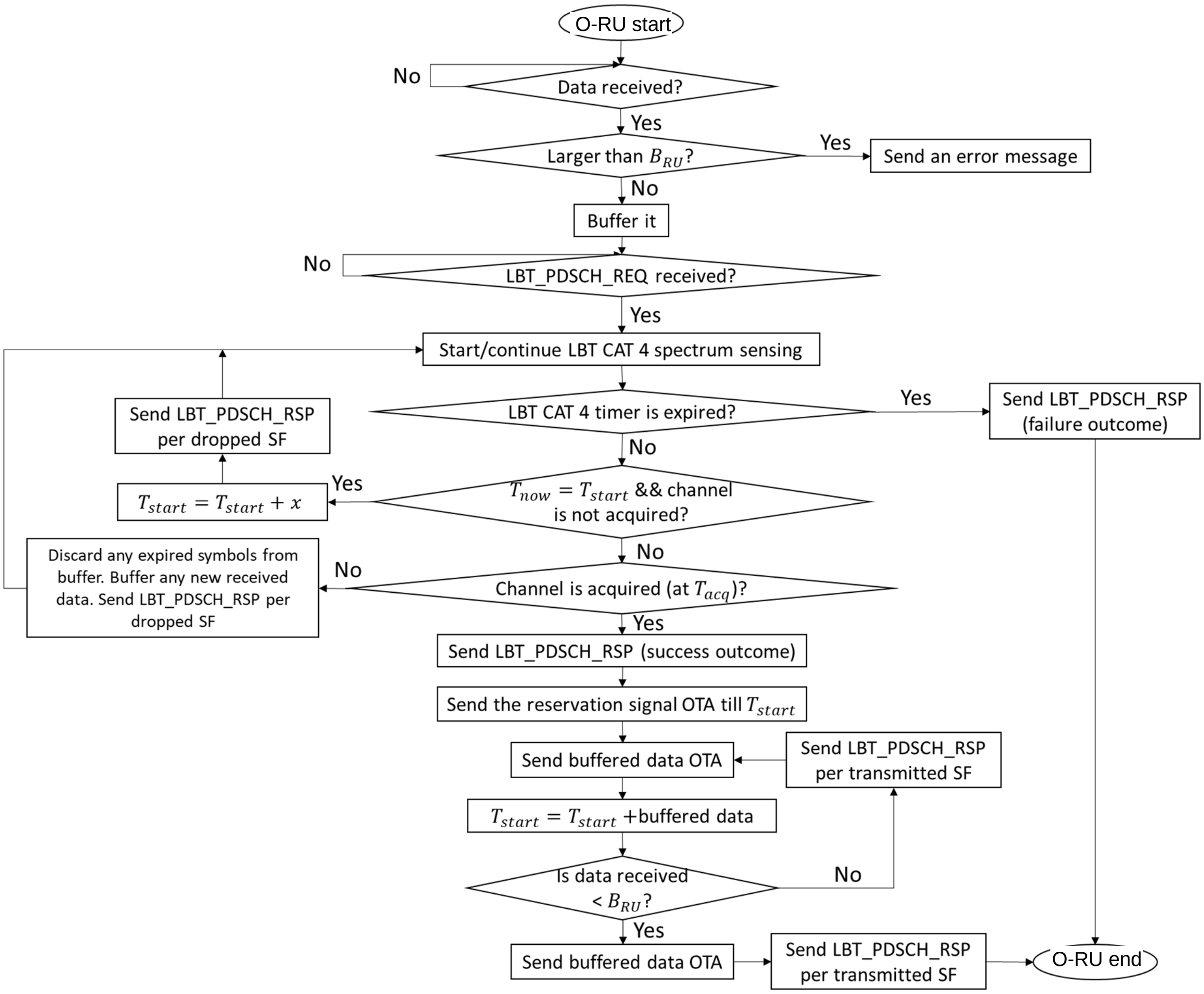

O-RU start
Data received?
No
Yes
Larger than B_{RU}?
Yes
Send an error message
No
Buffer it
LBT_PDSCH_REQ received?
No
Yes
Start/continue LBT CAT 4 spectrum sensing
LBT CAT 4 timer is expired?
Yes
Send LBT_PDSCH_RSP (failure outcome)
Send LBT_PDSCH_RSP per dropped SF
No
$T_{start} = T_{start} + x$
Yes
$T_{now} = T_{start}$ && channel is not acquired?
No
Discard any expired symbols from buffer. Buffer any new received data. Send LBT_PDSCH_RSP per dropped SF
No
Channel is acquired (at T_{acq})?
Yes
Send LBT_PDSCH_RSP (success outcome)
Send the reservation signal OTA till T_{start}
Send buffered data OTA
Send LBT_PDSCH_RSP per transmitted SF
$T_{start} = T_{start}$ +buffered data
Is data received < B_{RU}?
No
Yes
Send buffered data OTA
Send LBT_PDSCH_RSP per transmitted SF
O-RU end

Figure G.2 : PDSCH Transmission Algorithm O-RU flowchart

For DRS Transmission Procedure, the O-DU flowchart covers both cases, where the O-RU does 25 us sensing or LBT CAT 4

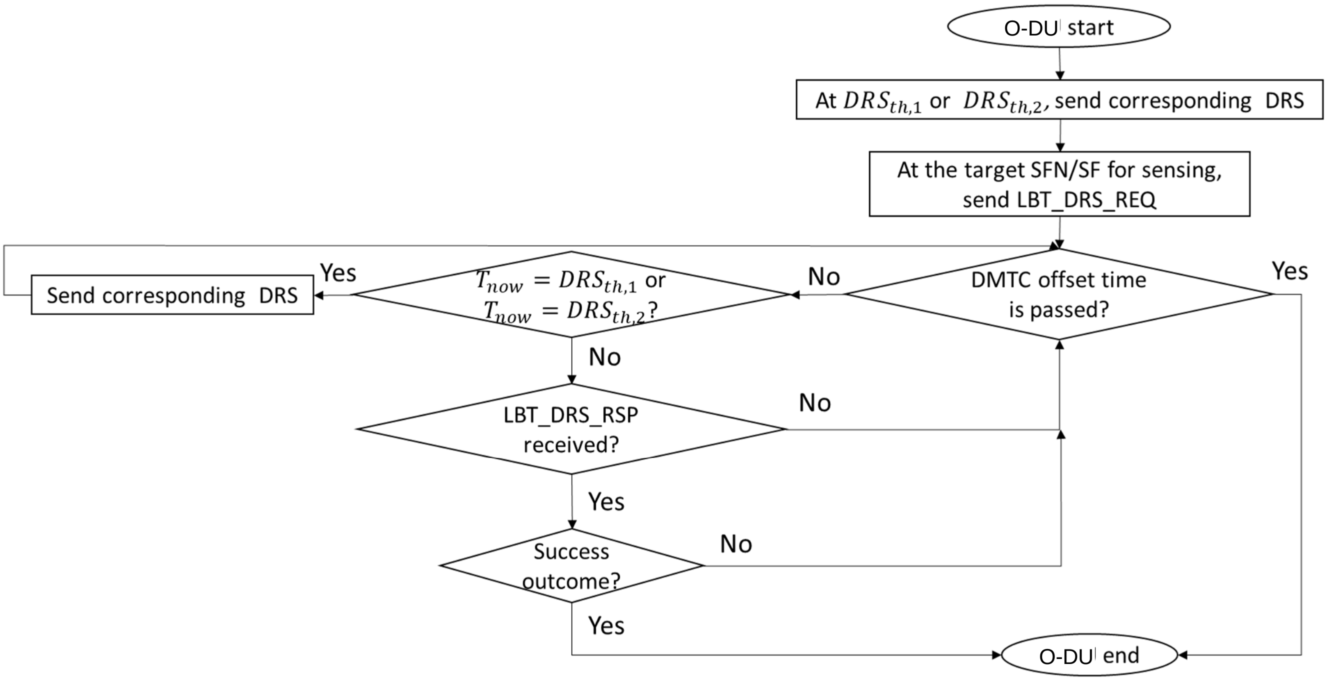

O-DU start
At $DRS_{th,1}$ or $DRS_{th,2}$, send corresponding DRS
At the target SFN/SF for sensing, send LBT_DRS_REQ
DMTC offset time is passed?
Yes
No
$T_{now} = DRS_{th,1}$ or $T_{now} = DRS_{th,2}$?
Yes
Send corresponding DRS
No
LBT_DRS_RSP received?
No
Yes
Success outcome?
No
Yes
O-DU end

Figure G.3 : DRS Transmission Procedure O-DU flowchart

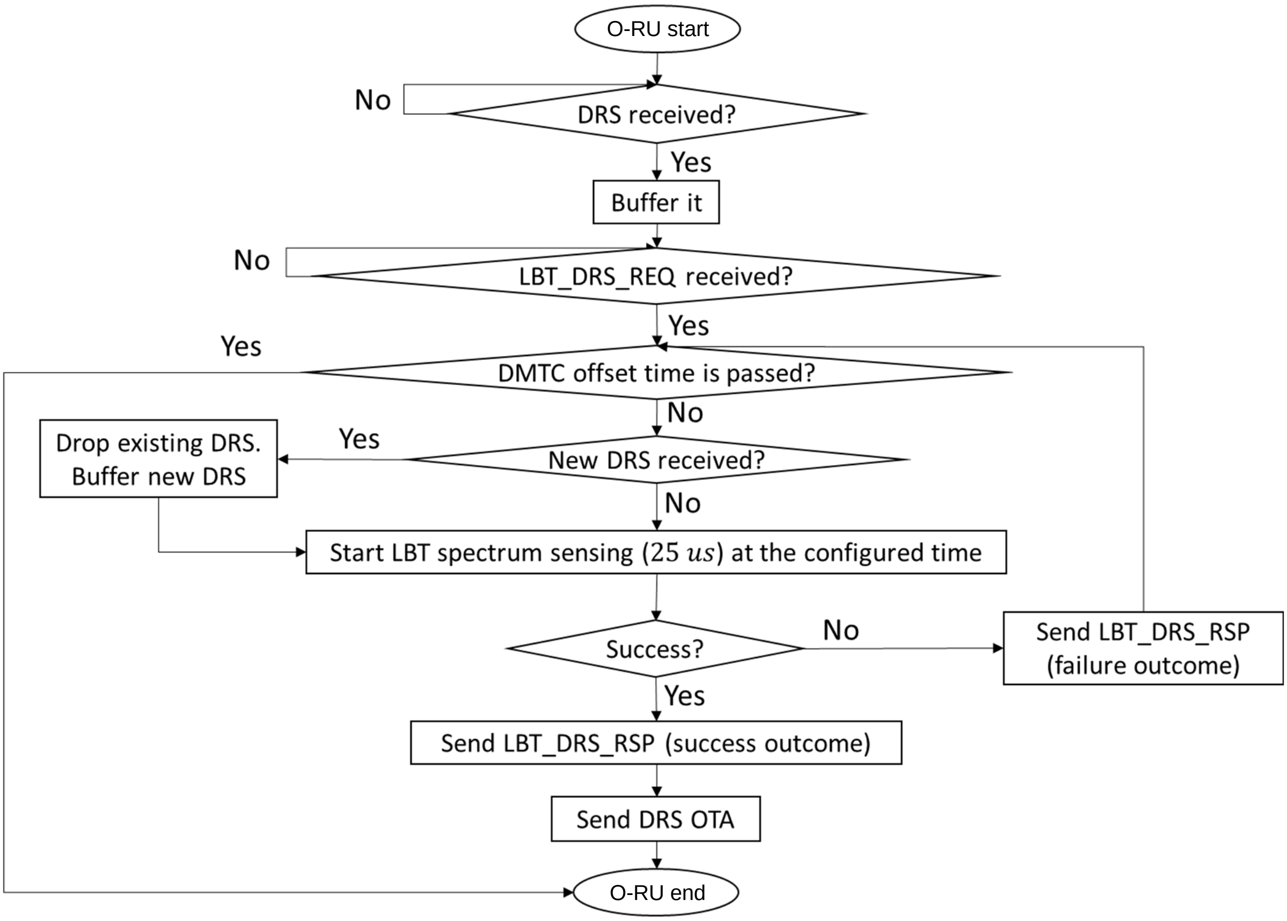

O-RU start
DRS received?
No
Yes
Buffer it
LBT_DRS_REQ received?
No
Yes
DMTC offset time is passed?
Yes
No
New DRS received?
Yes
Drop existing DRS. Buffer new DRS
No
Start LBT spectrum sensing (25 us) at the configured time
Success?
No
Send LBT_DRS_RSP (failure outcome)
Yes
Send LBT_DRS_RSP (success outcome)
Send DRS OTA
O-RU end

Figure G.4 : DRS Transmission Procedure O-RU flowchart – 25 µs sensing

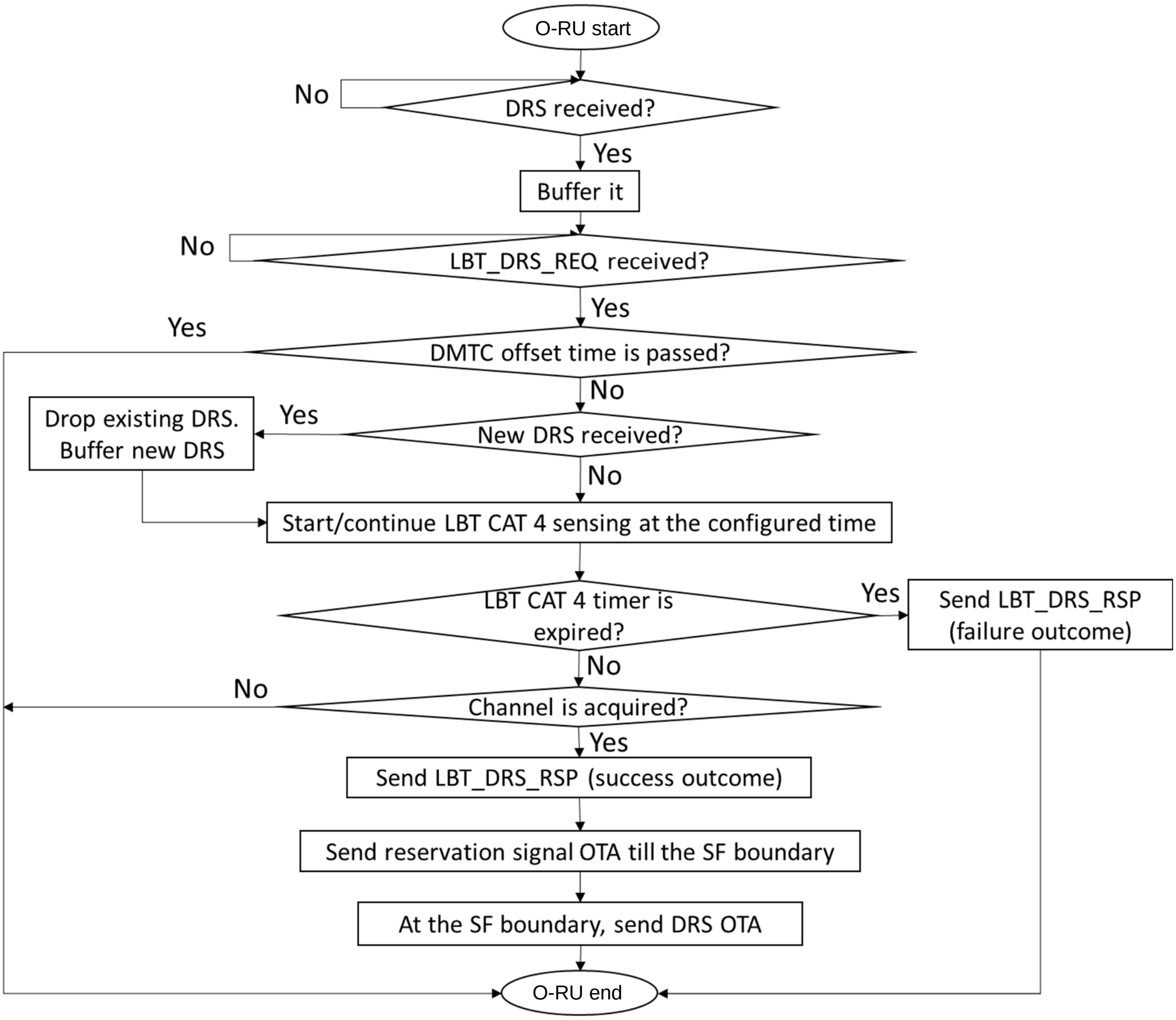

O-RU start
No
DRS received?
Yes
Buffer it
No
LBT_DRS_REQ received?
Yes
Yes
DMTC offset time is passed?
No
Drop existing DRS. Buffer new DRS
Yes
New DRS received?
No
Start/continue LBT CAT 4 sensing at the configured time
LBT CAT 4 timer is expired?
Yes
Send LBT_DRS_RSP (failure outcome)
No
No
Channel is acquired?
Yes
Send LBT_DRS_RSP (success outcome)
Send reservation signal OTA till the SF boundary
At the SF boundary, send DRS OTA
O-RU end

Figure G.5 : DRS Transmission Procedure O-RU flowchart – LBT Cat 4

Annex H (informative): S-Plane detailed frequency and phase error budget analysis, and future ITU-T clock types and classes reference

H.1 Reference documents

Clause 2 lists the standards which are referenced within this annex.

H.2 Frequency and time error budget analysis

This clause provides the informative analysis to support budget allocation in clause 11.3.2 for a Full Timing Support network (as per ITU-T G.8271.1 for the limits, ITU-T.G.8273.2 for the clock definition). The analysis serves 2 purposes:

Considering T-BC Class B and C switches (as per ITU-T G. 8273.2) in a ITU-T G.8271.1 compliant network, the number of allowed switches to satisfy the allocated network limit is computed in detail as an example.

NOTE: the following configurations are outside the scope of this annex, and are therefore For Further Study:

- ITU-T G.8271.1 compliant networks using class D T-BCs.
- ITU-T G.8271.1 compliant networks using T-TCs instead of T-BCs.
- Non-ITU-T G.8271.1-compliant networks, such as ITU-T G.8271.2 ones.
- O-RU using SyncE/eSyncE. In the example below, only PTP is considered as synchronization source for O-RU.

Each network element in the fronthaul clock chain generates time error (including constant cTE and dynamic dTE_H, dTE_L), which will accumulate through the entire clock chain and be present at the O-RU UNI, as described in ITU-T G.8271.1 Appendix IV. This Annex consider the accumulation of centered, symmetrical noise. In particular, accumulated dynamic time error will cause O-RU subordinate clock FFO (fractional frequency error) after clock recovery and filtering. Given O-RU shall meet the 3GPP air interface frequency accuracy target (±50ppb), O-RU filtering is needed to filter the accumulated dynamic time error and reduce the frequency error down to an acceptable level. The allowed network limit (i.e. dynamic time error), reasonable O-RU filter bandwidth and acceptable frequency error after filtering are the result of a compromise exercise as shown in the following analysis.

The value of the O-RU filtering bandwidth is a key compromise, combined with the local oscillator thermal sensitivity:

- The higher filtering bandwidth, the faster frequency correction of the local oscillator thermal sensitivity and therefore the lower temporary accumulated time error under thermal variations, but the poorer efficiency in low pass filtering the dynamic noise seen on the UNI
- The lower filtering bandwidth, the better efficiency in low pass filtering the dynamic noise seen on the UNI, but the poorer frequency correction of the local oscillator thermal sensitivity and therefore the higher temporary accumulated time error under thermal variations.

Frequency error budget for Network limit (LLS-C1 and LLS-C2) :

Based on the above compromise explanation, a practical expectation of O-RU filtering max BW is set to 75mHz to start the analysis.

Table H.1 : O-DU Frequency Error Budget

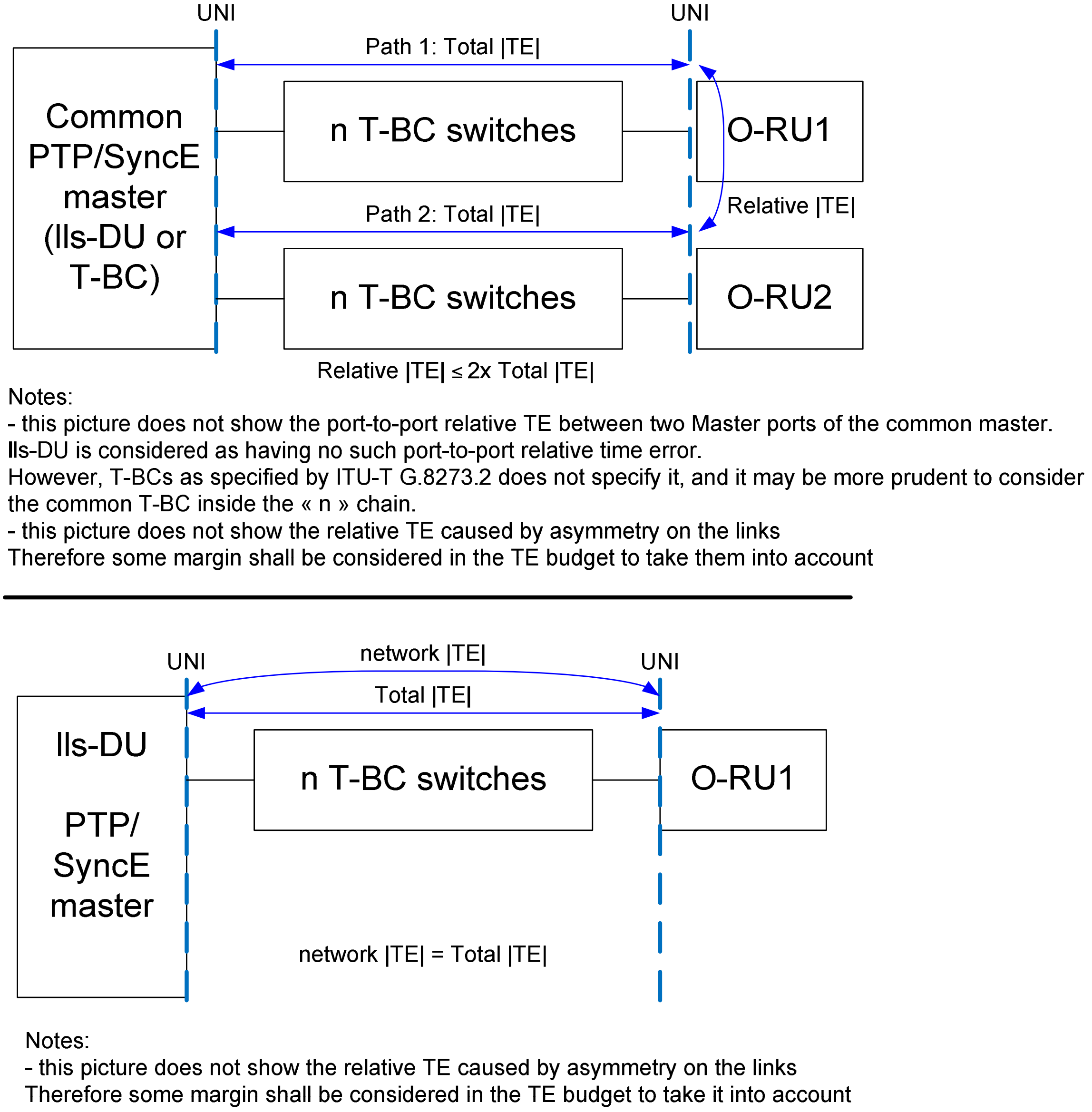
O-DU class	A	B
• Consider O-DU PTP/SyncE master frequency error budget = …………….. (refer to note 1 in clause 11.3.2.1)	±15 ppb	±5 ppb
• Consider O-RU total frequency error budget based on O-DU frequency error budget taken away from the 3GPP air interface (±50ppb) budget = …………	±35ppb	±45ppb
• Further split the O-RU total frequency error budget as follows as an example of O-RU design:		

o FFO (O-RU subordinate clock) =……………………………………………… o FFO (O-RU internal additive frequency noise) = ………………………	±21ppb ±14ppb.	±27ppb ±18ppb
• With FFO (O-RU subordinate clock) value and filter BW = 75mHz, based on ITU-T SG15 Q13 C1730, Geneva, 5 – 16 December 2011: ***FFO (in ppb) = ±2*π * $\lvert dTE_{L+H}\rvert$(in ns)*filter BW(in Hz)*** ⇨ FFO (O-RU subordinate clock) = 2π * $\lvert dTE_{L+H}\rvert$ * filter BW ⇨ $\lvert dTE_{L+H}\rvert$ = FFO (O-RU subordinate clock)/(2π * filter BW) = …………………… which is the max allowed network noise limit (between O-DU UNI and O-RU UNI) guaranteeing FFO at the output of the O-RU filter with 75mHz BW. Note that after this network noise limit is agreed in O-RAN spec, it is up to O-RU vendor implementation to select filter BW (not necessarily 75mHz) to trade off the internal budget split between FFO (O-RU subordinate clock) and FFO (O-RU internal additive frequency noise) as long as the O-RU total frequency error budget (±35ppb or ±43ppb) is still met.	±45ns	±57ns
• Based on G.8271.1 Appendix IV guidance to calculate accumulated error: ⇨ Total dynamic noise = RMSsum (dTE_{L+H}) ⇨ $\lvert dTE_{L+H}\rvert$ = RMSsum ($\lvert dTE_{L+H}\rvert$ of all nodes excluding O-RU's T-TSC) Consider the model of clock chain of n T-BC clocks (between O-DU UNI to RU UNI) ITU-T G.8273.2 (class B) switch: $\lvert dTE_{L}\rvert$ = 20ns, $\lvert dTE_{H}\rvert$ = 35ns ⇨ $\lvert dTE_{L+H}\rvert$ limit = sqrt [$n*20^2 + 35^2$] ns = ……………………………………… ⇨ n = ($\lvert dTE_{L+H}\rvert^2 - 35^2)/20^2$, ………………………………………………………… the maximum number of class B T-BCs in each chain (excluding O-DU) Consider the model of clock chain of n T-BC clocks (between O-DU UNI to RU UNI) ITU-T G.8273.2 (class C) switch: : $\lvert dTE_{L}\rvert$ = 5ns, $\lvert dTE_{H}\rvert$ = 10ns (Note 1) ⇨ $\lvert dTE_{L+H}\rvert$ limit = sqrt [$n*5^2 + 10^2$] ns = ……………………………………… ⇨ n = ($\lvert dTE_{L+H}\rvert^2 - 10^2)/5^2$, ………………………………………………………… the maximum number of class C T-BCs in each chain (excluding O-DU) Note 1: This $\lvert dTE_{H}\rvert$ limit is not yet specified by ITU-T G.8273.2 and is therefore an estimation.	±45ns 2 ±45ns >>10	±57ns 5 ±57ns >>10

Time error budget for network limit (LLS-C1 and LLS-C2) :

Using existing class B T-BCs, and considering no time error contribution by the fiber asymmetry nor from two master ports of the same T-BC, then:

Following G.8271.1 Appendix IV guidance to calculate accumulated error with the following clock chain models:



Notes:
- this picture does not show the port-to-port relative TE between two Master ports of the common master. lls-DU is considered as having no such port-to-port relative time error.
However, T-BCs as specified by ITU-T G.8273.2 does not specify it, and it may be more prudent to consider the common T-BC inside the « n » chain.
- this picture does not show the relative TE caused by asymmetry on the links
Therefore some margin shall be considered in the TE budget to take them into account

Notes:
- this picture does not show the relative TE caused by asymmetry on the links
Therefore some margin shall be considered in the TE budget to take it into account

Figure H.1 : clock chain model for analysis

As per ITU-T G.8271.1 Appendix IV:

Total |TE| = sum (|cTE| of n nodes) + RSS sum (|dTE_L| of n nodes and |dTE_H| of last node)

$$= n*|cTE| + \text{sqrt}\,(n*|dTE_L|^2 + |dTE_H|^2)$$

However, the O-RU's time error budget (cTE, $|dTE_L|^2 + |dTE_H|^2$) has already been taken into account in the budget (see tables below), so only the T-BCs in the network are included in "n" and their contribution is limited to cTE and dTE_L. As a result, the above general formula can be further simplified for the chain of n T-BCs excluding the O-RU's T-TSC:

Total $|TE_L| = n*|cTE| + \text{sqrt}\,(n*|dTE_L|^2)$,

Where a node is based on T-BC Class B switch with the following noise generation specification:

Constant time error = |cTE| = 20ns for class B, 10ns for class C

Low-band dynamic error = $|dTE_L|$ = 20ns for class B, 5ns for class C (considering centered noise)

Table H.2 : O-RU Time Error Budget

For O-RU type= and limit = to meet category (notes 1, 2)	Enhanced 60ns Cat A	Regular 100ns Cat B	Enhanced 190ns Cat B	Regular 95ns Cat C	Enhanced 140ns Cat C
n*|cTE| + sqrt (n*|dTE$_L$| ²) < limit per branch	30ns	50ns	95ns	95ns	140ns
maximum n value, number of class B T-BCs on each branch from common network element (either O-DU or T-BC) to O-RU: Remaining relative TE margin to be assigned to fiber asymmetry and relative TE between two ports of the common network element	0 60ns	1 20ns	2 53ns	2 12ns	4 20ns
maximum n value, number of class C T-BCs on each branch from common network element (either O-DU or T-BC) to O-RU: Remaining relative TE margin to be assigned to fiber asymmetry and relative TE between two ports of the common network element	1 15ns	3 12ns	7 12ns	7 12ns	11 14ns
NOTE 1: Proposed maximum n values are conservative and leave at least 10ns margin for fiber asymmetry or relative TEL between two ports of the branching clock. It is therefore recommended to limit the number of fronthaul clocks to this value, although an additional one may still allow meeting the expected limits. NOTE 2: For IEEE802.1CM cat A and cat B, the limit corresponds to a relative TEL between two O-RU input ports (the end of two branches), and therefore the limit per branch is half. For IEEE802.1CM cat A and cat B, the limit corresponds to a relative TEL between the O-DU output and any O-RU input ports, and therefore the limit per branch is the same value.					

Frequency error budget for Network limit (LLS-C3) :

- Based on the above compromise explanation, a practical expectation of O-RU filtering max BW is set to 75mHz to start the analysis
- Based on G.8272, PRTC/T-GM MTIE (during lock) specification can be used to describe PRTC/T-GM dynamic noise generation:

Table H.3 : Wander Generation (MTIE) for PRTC-A

MTIE limit (us)	Observation interval (s)
$0.275 \times 10^{-3}\,\tau + 0.025$	$0.1 < \tau \le 273$
0.10	$\tau > 273$

Table H.4 : Wander Generation (MTIE) for PRTC-B

MTIE limit (us)	Observation interval (s)
$0.275 \times 10^{-3}\,\tau + 0.025$	$0.1 < \tau \le 54.5$
0.04	$\tau > 54.5$

⇨ Given O-RU filtering max BW = 75mHz, it corresponds to observation interval τ = 1/(π*75mHz) = 4.2s.
From the above table, MTIE limit (with τ = 4.2s) = 26.2ns pk-pk for both PRTC-A and PRTC-B.
From this MTIE number, the value of dTE$_L$ is computed as 13.1ns.

⇨ Besides MTIE, which can be treated as dynamic noise during lock condition, there is additional consideration of PRTC/T-GM during holdover condition. Potential semi-static frequency drift could happen during holdover, ±2ppb is reserved based on ITU-T G.8271.1 Appendix V PRTC failure scenario (b) which permits 400ns holdover limit for short period of 5 minutes.

Table H.5 : Network (LLS-C3) Frequency Error Budget

PRTC class	A
Consider PRTC PTP/SyncE master frequency error budget =……………………………………… (refer to note 1 in clause 11.3.2.2)	±2 ppb
Consider O-RU total frequency error budget based on O-DU frequency error budget taken away from the 3GPP air interface (±50ppb) budget = …………………………………………………………	±48ppb
• Further split the O-RU total frequency error budget as follows as an example of O-RU design: o FFO (O-RU subordinate clock) =…………………………………………………………… o FFO (O-RU internal additive frequency noise) = ……………………………………	±30ppb ±18ppb
• With FFO (O-RU subordinate clock) value and filter BW = 75mHz, based on ITU-T SG15 Q13 C1730, Geneva, 5 – 16 December 2011: $\boldsymbol{FFO\ (in\ ppb) = \pm 2*\pi * \lvert dTE_{L+H}\rvert (in\ ns)*filter\ BW(in\ Hz)}$ ⇨ FFO (O-RU subordinate clock) = 2π * $\lvert dTE_{L+H}\rvert$ * filter BW ⇨ $\lvert dTE_{L+H}\rvert$ = FFO (O-RU subordinate clock)/(2π * filter BW) = ………………………………… which is the max allowed network noise limit (between O-DU UNI and O-RU UNI) guaranteeing FFO at the output of the O-RU filter with 75mHz BW. Note that after this network noise limit is agreed in O-RAN spec, it is up to O-RU vendor implementation to select filter BW (not necessarily 75mHz) to trade off the internal budget split between FFO (O-RU subordinate clock) and FFO (O-RU internal additive frequency noise) as long as the O-RU total frequency error budget (±35ppb or ±45ppb) is still met.	±63ns
• Based on G.8271.1 Appendix IV guidance to calculate accumulated error: ⇨ Total dynamic noise = RMSsum (dTE_{L+H}) ⇨ $\lvert dTE_{L+H}\rvert$ = RMSsum ($\lvert dTE_{L+H}\rvert$ of all nodes including PRTC/T-GM but not O-RU's T-TSC) • Consider the model of clock chain of either class A or B PRTC/T-GM and n T-BC switches (between PRTC input to O-RU UNI) and using G.8272 MTIE specification for the PRTC and G.8273.2 dTE_{L+H} specification for the T-BC. ⇨ PRTC/T-GM dynamic noise = MTIE/2 based on max 75mHz O-RU filter BW assumption: …………………………………………………………………………… ⇨ T-BC Class B switch dynamic noise = $\lvert dTE_L\rvert$ = 20ns, $\lvert dTE_H\rvert$ = 35ns ⇨ $\lvert dTE_{L+H}\rvert = sqrt(13^2 + n*20^2 + 35^2)$ ns= ……………………………………………………… ⇨ Maximum n= ($\lvert dTE_{L+H}\rvert^2 - 35^2 - 13^2) / 20^2$ = ……………………………………………… the maximum number of class B T-BCs in each chain (after PRTC) ⇨ T-BC Class C switch dynamic noise = $\lvert dTE_L\rvert$ = 5ns, $\lvert dTE_H\rvert$ = 10ns ⇨ $\lvert dTE_{L+H}\rvert = sqrt(13^2 + n*5^2 + 10^2)$ ns= …………………………………………………… ⇨ Maximum n= ($\lvert dTE_{L+H}\rvert^2 - 10^2 - 13^2) / 5^2$ = …………………………………………… the maximum number of class C T-BCs in each chain (after PRTC)	13ns 63ns 6 63ns >>10

Time error budget for network limit (LLS-C3):

G.8271.1 Appendix V (Example of design options) and Appendix XII (Examples of design options for fronthaul and clusters of base stations) provide guidelines on the number of switches that can be deployed in case of LLS-C3 for the different target requirements.

Appendix V is focusing on the absolute Time Error Requirement (Category C), while Appendix XII addresses also relative time error requirements applicable in fronthaul (Category A and B).

H.3 Summary of allowed number of switches:

The maximum allowed number of switches shall be determined based on the smallest allowed number constraint by

- Frequency error budget

- Operator-chosen most constraint time error budget category
- The class of network elements (note that the O-RU classes are examples proposed by IEEE802.1 CM)

Table H.6 : Network Frequency Error Budget

Frequency Error Network limit	LLS-C1 and LLS-C2, class A O-DU	LLS-C1 and LLS-C2, class B O-DU	LLS-C3	Comment
Absolute Frequency error budget between time source and O-RU	2 (class B T-BC) >>10 (class C T-BC)	5 (class B T-BC) >>10 (class C T-BC)	Note 3	Any branch shall not exceed this number of T-BCs from O-DU or PRTC/T-GM to meet 50ppb frequency accuracy at the air interface.

Table H.7 : Network Time Error Budget

Time Error Network limit	LLS-C1 and LLS-C2, class A O-DU	LLS-C1 and LLS-C2, class B O-DU	LLS-C3	Comment
Cat A Relative Time error budget (with enhanced O-RUs) between O-RUs	0 (class B T-BC) 1 (class C T-BC)	0 (class B T-BC) 1 (class C T-BC)	Note 3 Note 4	Any branch shall not exceed this number of T-BCs from common T-BC to meet target relative time error limit at the air interface. Value 0 means that only lls-C1 is supported
Cat B Relative Time error budget (with regular O-RUs) between O-RUs	1 (class B T-BC) 3 (class C T-BC)	1 (class B T-BC) 3 (class C O-RU)	Note 3	
Cat B Relative Time error budget (with enhanced O-RUs) between O-RUs	2 (class B T-BC) 7 (class C T-BC)	2 (class B T-BC) 7 (class C T-BC)	Note 3	
Cat C Absolute Time error budget (with regular O-RUs) between time source and O-RU	2 (class B T-BC) 7 (class C T-BC)	2 (class B T-BC) 11 (class C T-BC)	Note 3	Any branch shall not exceed this number of T-BCs from O-DU or PRTC/T-GM to meet 1500ns absolute time error limit at the air interface. Note 1.
Cat C Absolute Time error budget (with enhanced O-RUs) between time source and O-RU	4 (class B T-BC) (further limit to 2 due to freq. limit) 7 (class C T-BC)	4 (class B T-BC) 11 (class C T-BC)	Note 3	

Note 1 : Only applicable to lls-C1 and lls-C2: As indicated in table 9-3, the maximum Time Error at the output of the O-DU is 1420 ns for lls-C1 and 1325 ns for lls-C2. This limit considers that the input of the O-DU stays within the limits at Reference point C defined by ITU-T G.8271.1 or ITU-T G.8271.2 .

Note 2 : The analysis on the number of switches (for time error budget) is meant to rough estimate and excludes both fiber asymmetry factor and relative TE between two ports of the common network element. It is operator's responsibility to control these two parameters. The analysis for each time error budget (specific category) has some left-over margin that could be used to cover them. If the left-over margin is not enough, the alternative is to reduce the allowed number of switches.

Note 3: network design guidelines for configuration LLS-C3 are provided in ITU-T G.8271.1 Appendix V (addressing IEEE802.1CM synchronization Category C) and Appendix XII (addressing IEEE802.1CM synchronization Category A and B) of G.8271.1. The guidelines in Appendix V includes also indication on allowed number of switches between the PRTC/T-GM and the O-RU. The guidelines in Appendix XII include also indication on allowed number of switches between a clock that is common for the cooperating O-RUs, and these O-RUs.

As an example:

- in order to meet the Cat B requirements with a regular O-RU, the number of T-BC class C switches, after the common switch (itself a class C T-BC), should be 3 or less, while there is no room for additional switches with T-BC class B clocks after the common switch (itself a class B T-BC).

- In order to meet the Cat B requirements with an enhanced O-RU, the number of T-BC class C switches should be 6 or less after the common switch (itself a class C T-BC), and the number of T-BC class B switches after the common switch (itself a class B T-BC), should be 2 or less.

The minor differences between the maximum number of switches supported in LLS-C2 and LLS-C3 are due to different characteristics of the common clock (as indicated in Figure H-1, in LLS-C2, the O-DU has no relative time error between ports specified, while in LLS-C3, the G.8273.2 T-BC takes this into account).

G.8271.1 also presents the case of an alternative deployment with a short clock chain that has a maximum of 4 Class C T-BC, or 1 Class B T-BC between the PRTC/T-GM and the O-RU (see reference network model in Figure II.6 of G.8271.1 with a PRTC-B/T-GM directly connected to the common T-BC). For this case the regular O-RU was considered as it represents the worst-case scenario. This deployment, in addition to meeting IEEE802.1CM synchronization Category C, is also suitable to support IEEE802.1CM synchronization Category B.

Note 4: Cat A requirements concerns co-located O-RUs. It is assumed that the cooperating O-RUs are connected to the same switch (therefore there is no switch after the common T-BC).

Annex I (informative): Precoding and Examples

I.1 Case 1: Tx Diversity 1-CRS Port Ant0, 1 PRB:

At the O-DU

- For single Tx case, $d^{(q)}(0),...,\ d^{(q)}(M_{\text{symb}}^{(q)}-1)$ M_q_sym modulation bits belonging to PDSCH ANT0 are packed into a PRB. All CRS REs for ANT0 are packed into a PRB for transmission and are unpacked at the O-RU.

At the O-RU

- At the O-RU, for single antenna port single Tx a single layer is used and mapping is defined as $x^{(0)}(i)=d^{(0)}(i)$
- O-RU needs to map CRS REs into antenna ports using crsReMask, crssymbolNumber and crsShift. Follow case 5

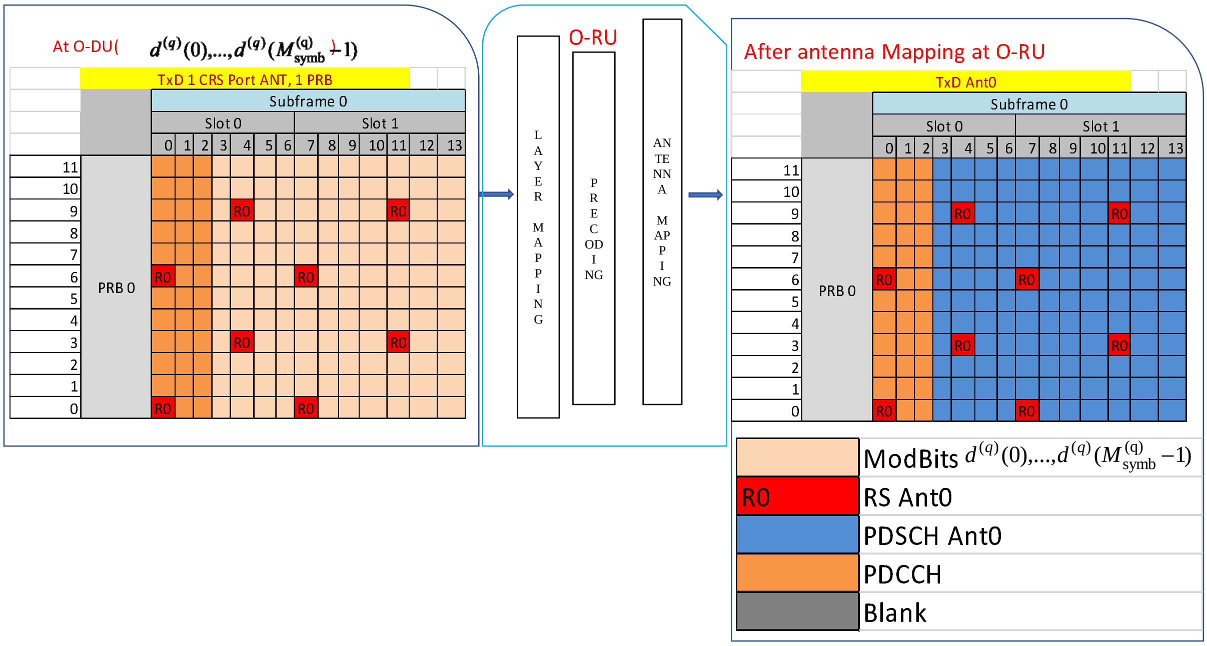


Figure I.1 : Single Tx 1-CRS Port Ant0, 1 PRB

I.2 Case 2: Tx Diversity 2-CRS Port Ant0, Ant1, and 1 PRB

Case 2.1 At the O-DU

- For TxD case, $d^{(q)}(0),...,\ d^{(q)}(M_{\text{symb}}^{(q)}-1)$ M_q_sym modulation bits belonging to PDSCH are packed into a PRB. All CRS REs for ANT0 and ANT1 are packed into the same PRB for transmission and are unpacked at the O-RU.

Case 2.2 At the O-RU

- At the O-RU, for two antenna port TxD 2 layers are used and mapping is defined as $\begin{array}{l} x^{(0)}(i)=d^{(0)}(2i) \\ x^{(1)}(i)=d^{(0)}(2i+1) \end{array}$
- For TxD, information for 2 layers are packed into a PRB for transmission and are unpacked at the O-RU. At the O-RU, after layer mapping and, precoding, CRS REs for 2 antenna ports are mapped to the appropriate RE positions and rest are left blank (gray-shaded REs) as illustrated below. Follow Case 5.

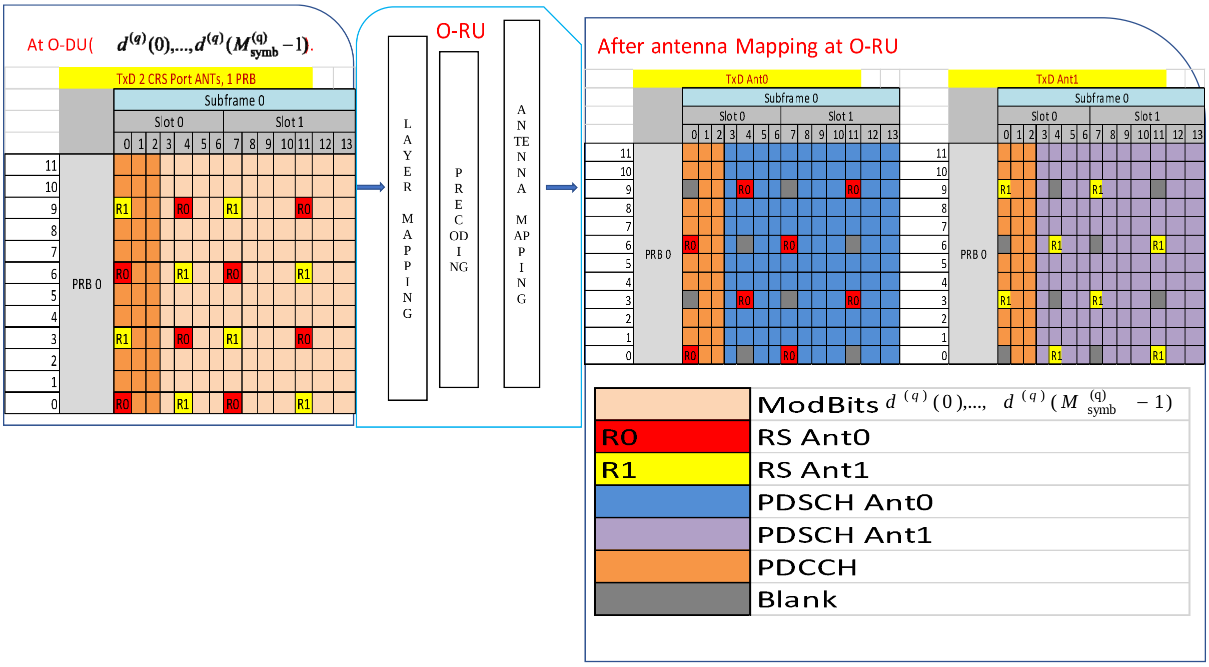

At O-DU($d^{(q)}(0),...,d^{(q)}(M^{(q)}_{symb}-1)$.
TxD 2 CRS Port ANTs, 1 PRB
Subframe 0
Slot 0
Slot 1
PRB 0
R0
R1
O-RU
LAYER MAPPING
PRECODING
ANTENNA MAPPING
After antenna Mapping at O-RU
TxD Ant0
TxD Ant1
ModBits $d^{(q)}(0),...,\ d^{(q)}(M^{(q)}_{symb}-1)$
R0 RS Ant0
R1 RS Ant1
PDSCH Ant0
PDSCH Ant1
PDCCH
Blank

Figure I.2 : Tx Diversity 2-CRS Port Ant0, Ant1, and 1 PRB

I.3 Case 3: Tx Diversity 4-CRS Port Ant0,1,2,3 and 1 PRB

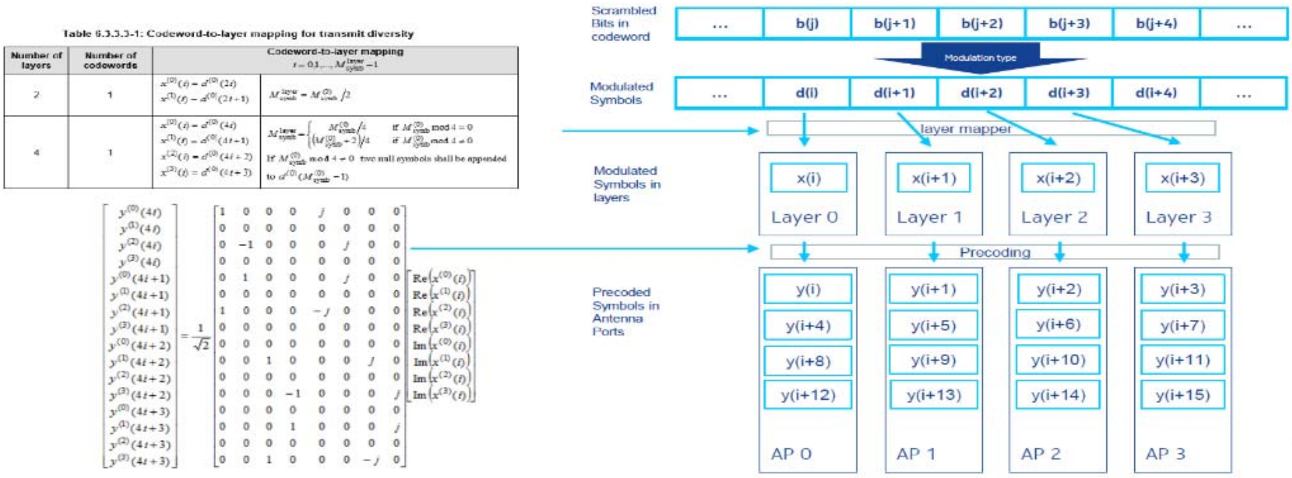


Figure I.3 : Case 3 Layer Mapping

Case 3.1 At the O-DU

- For TxD case, $d^{(q)}(0),...,\ d^{(q)}(M^{(q)}_{symb}-1)$ M_q_sym modulation bits belongs to PDSCH are packed to into a PRB. All CRS RE's for ANT0, ANT1, ANT2 and ANT3 are packed into that same PRB for transmission and are unpacked at the O-RU.

Case 3.2 At the O-RU

- At the O-RU, for two antenna port TxD 4 layers are used and mapping is defined as

$$\begin{aligned} x^{(0)}(i) &= d^{(0)}(4i) \\ x^{(1)}(i) &= d^{(0)}(4i+1) \\ x^{(2)}(i) &= d^{(0)}(4i+2) \\ x^{(3)}(i) &= d^{(0)}(4i+3) \end{aligned}$$

- For TxD, user data RE's for 4 layers are packed into a PRB for transmission and are unpacked at the O-RU. At the O-RU, after layer mapping and precoding, CRS REs for 4 antenna ports are mapped to the appropriate RE position and rest are left blank as illustrated below.

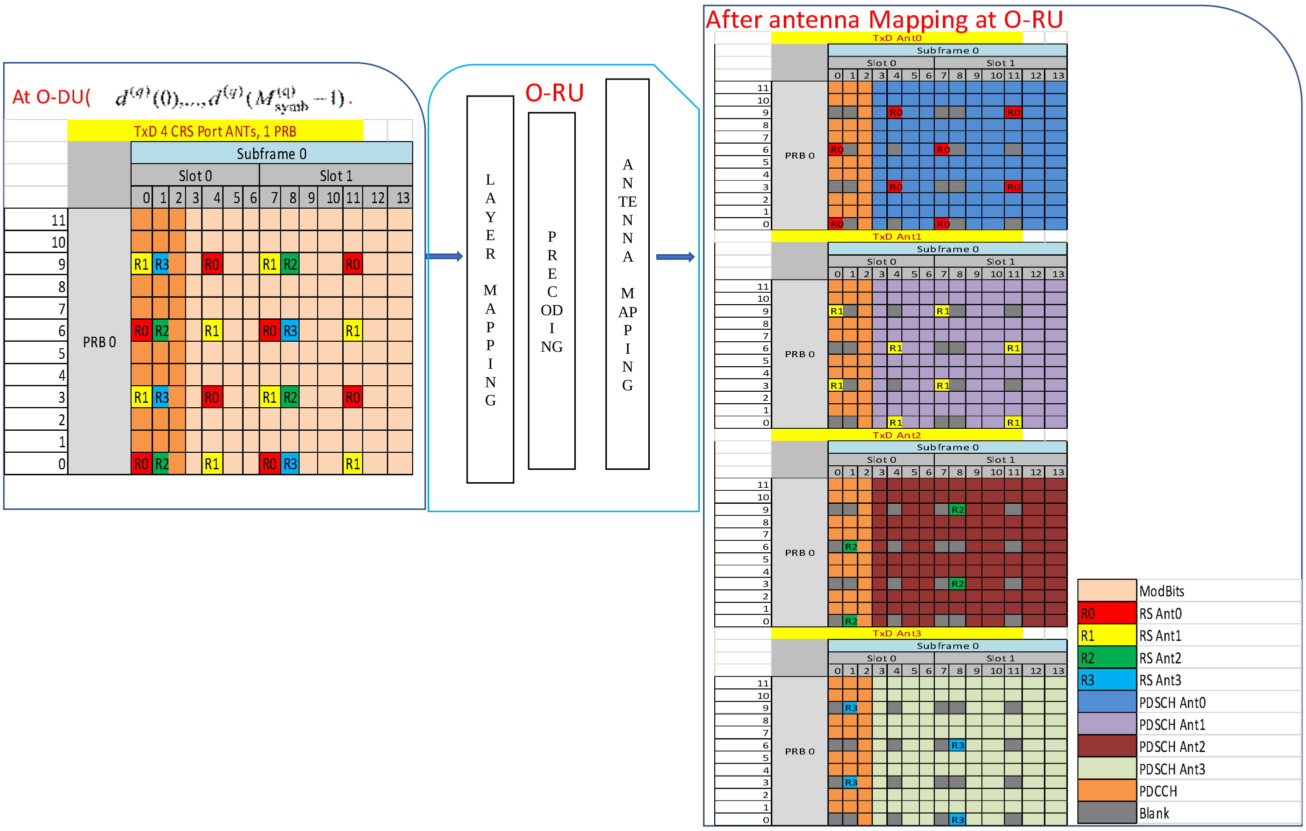


Figure I.4 : Tx Diversity 4-CRS Port Ant0,1,2, 3, and 1 PRB

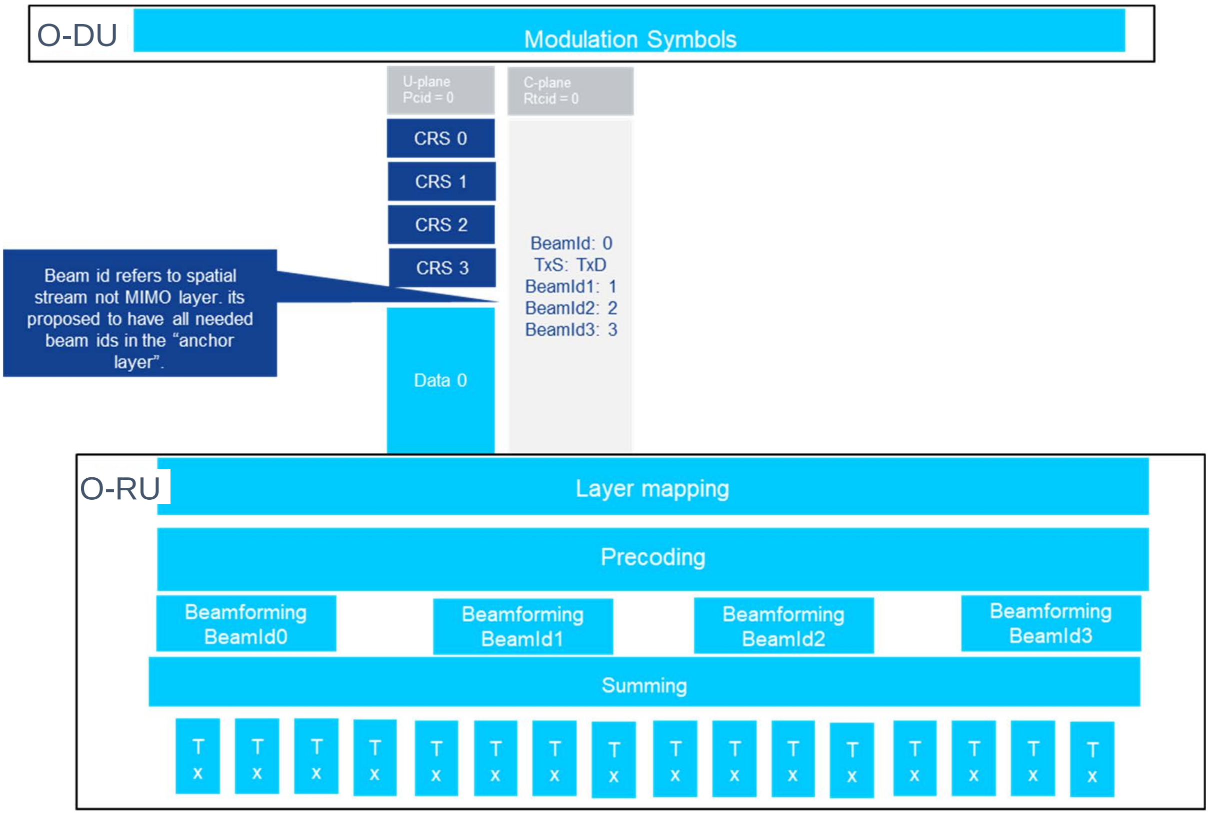


Figure I.5 : TxD – 4 CRS Ports

I.4 Case 4: TM3/TM4 3 Layers, 4 Antenna ports

One symbol from each of layers is linearly mapped to each antenna port.

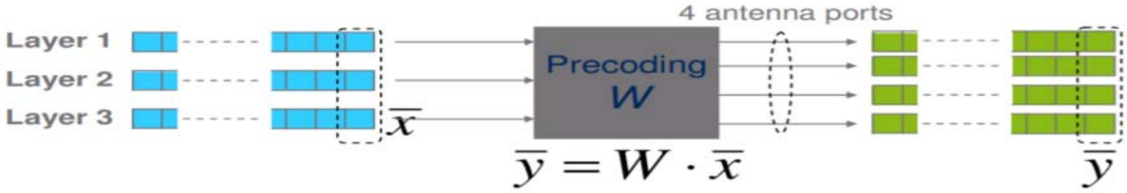


Figure I.6 : Case 4 Layer Mapping

For TM3/TM4, REs belonging to all the antenna ports are mapped to the zeroth layer at the O-DU. The O-RU should consider layer zero's CRS REs for CRS mapping for all the antenna ports using reMask bit field; CRS RE positions in the non-zero layers can be ignored as illustrated below (gray-shared REs in the O-DU represent CRS RE positions that are not populated by the O-DU.

$$x^{(0)}(i) = d^{(0)}(3i)$$

$$x^{(1)}(i) = d^{(0)}(3i+1) \quad M_{\text{symb}}^{\text{layer}} = M_{\text{symb}}^{(0)} / 3$$

$$x^{(2)}(i) = d^{(0)}(3i+2)$$

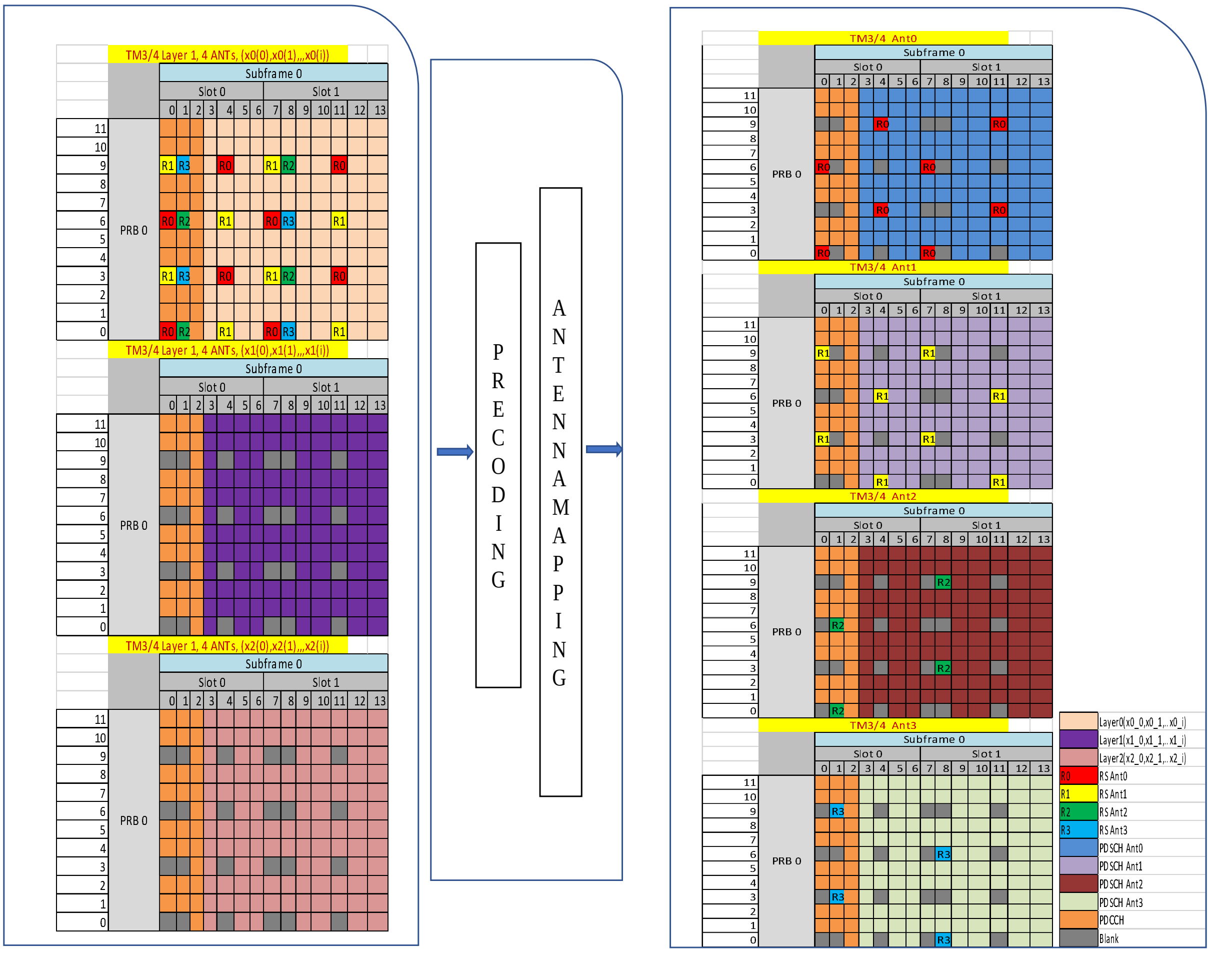


Figure I.7 : TM3/TM4, 3 Layers 4 Antenna Ports

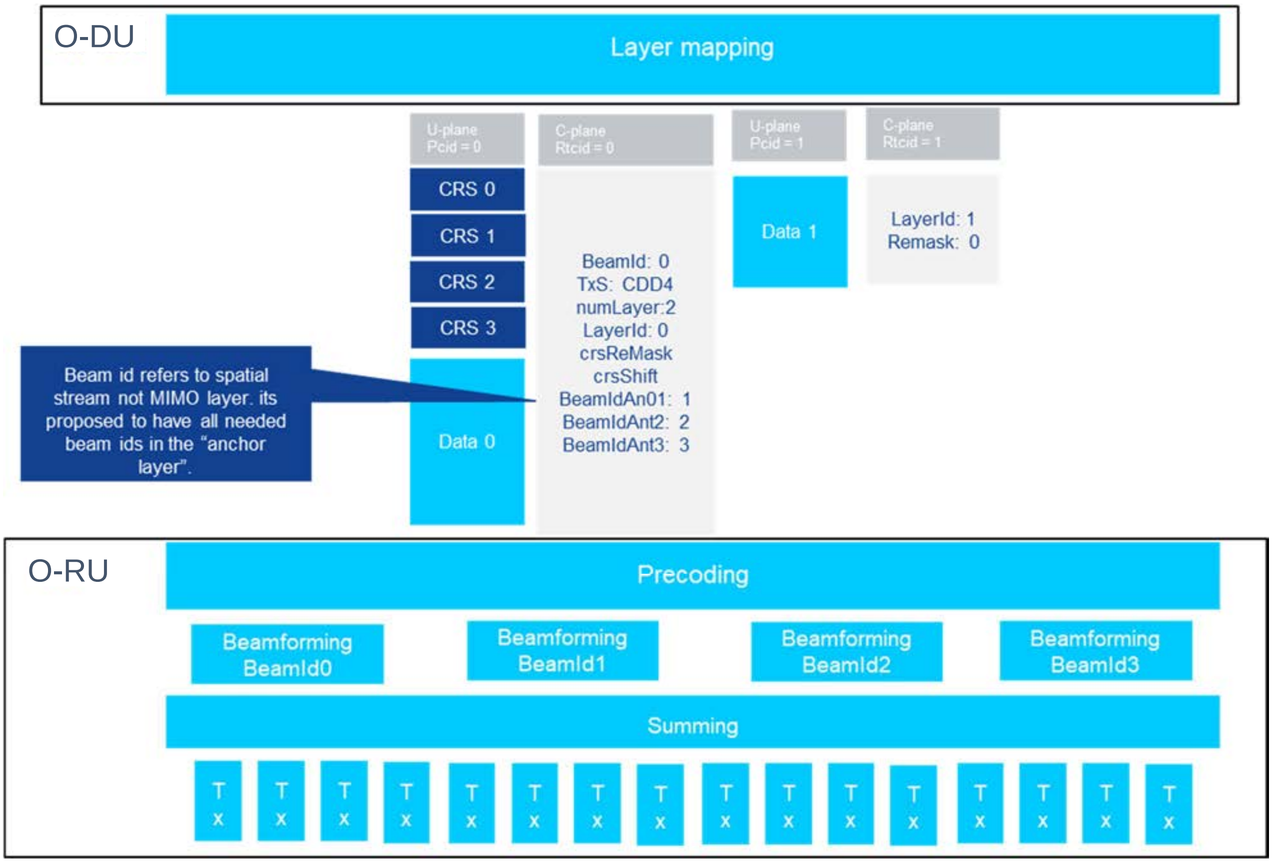


Figure I.8 : TM3 rank 2/4 CRS Ports

I.5 CRS location assignment: reMask Bit position for Layer0, 2 and 4 for all possible vShift cases

NOTE: CRS shift pattern for antenna port 0 when 3 ≤ vshift ≤ 5 is same as 0 ≤ vshift ≤ 3 for antenna port 1 resulting in only three possible vshift for a 2-antenna port case and this holds good for 4-antenna port case as well. Hence the crsReMask is differentiated with the crsShift field. If the O-RU identifies crsShift as 1, then shift in CRS belongs to each antenna have to read properly.

Table I.1 : CRS Location Assignment For layer1

	Layer1				
		crsSymNum 0 and 7		crsSymNum 4 and 11	
vshift	crsShift	crsReMask	PosInd in Bit Ant 0	crsReMask	PosInd in Bit Ant 0
0	0	0000 0100 0001	0,6	0010 0000 1000	3,9
1	0	0000 1000 0010	1,7	0100 0001 0000	4,10
2	0	0001 0000 0100	2,8	0100 0010 0000	5,11
3	0	0010 0000 1000	3,9	0000 0100 0001	6,0
4	0	0100 0001 0000	4,10	0000 1000 0010	7,1
5	0	1000 0010 0000	5,11	0001 0000 0100	8,2

crsReMask is not repeated for all for vShift combinations, hence crsShift is always indicated as 0.

Table I.2 : CRS Location Assignment For layer2

	Layer2					
		crsSymNum 0 and 7			crsSymNum 4 and 11	
vshift	crsShift	crsReMask	PosInd in Bit Ant 0	PosInd in Bit Ant 1	crsReMask	PosInd in Bit Ant 0	PosInd in Bit Ant 1
0	0	0010 0100 1001	0,6	3,9	0010 0100 1001	3,9	0,6
1	0	0100 1001 0010	1,7	4,10	0100 1001 0010	4,10	1,7
2	0	1001 0010 0100	2,8	5,11	1001 0010 0100	5,11	2,8
3	1	0010 0100 1001	3,9	6,0	0010 0100 1001	6,0	3,9
4	1	0100 1001 0010	4,10	7,1	0100 1001 0010	7,1	4,10
5	1	1001 0010 0100	5,11	8,2	1001 0010 0100	8,2	5,11

crsReMask for vShift 0 is same as for vShift 3 case but RE locations are different for Ant 0 and 1. Hence indicating 1 for 3,4,5 vShift cases, thereby O-RU shall pick the positions as interpreted in above table.

Table I.3 : CRS Location Assignment For layer4

	Layer4						
		crsSymNum 0 and 7			crsSymNum 4 and 11		
vshift	crsShift	crsReMask	PosInd in Bit Ant 0	PosInd in Bit Ant 1	crsReMask	PosInd in Bit Ant 0	PosInd in Bit Ant 1
0	0	0010 0100 1001	0,6	3,9	0010 0100 1001	3,9	0,6
1	0	0100 1001 0010	1,7	4,10	0100 1001 0010	4,10	1,7
2	0	1001 0010 0100	2,8	5,11	1001 0010 0100	5,11	2,8
3	1	0010 0100 1001	3,9	6,0	0010 0100 1001	6,0	3,9
4	1	0100 1001 0010	4,10	7,1	0100 1001 0010	7,1	4,10
5	1	1001 0010 0100	5,11	8,2	1001 0010 0100	8,2	5,11

	Layer4						
		crsSymNum 1			crsSymNum 8		
vshift	crsShift	crsReMask	PosInd in Bit Ant 2	PosInd in Bit Ant 3	crsReMask	PosInd in Bit Ant 3	PosInd in Bit Ant2
0	0	0010 0100 1001	0,6	3,9	0010 0100 1001	0,6	3,9
1	0	0100 1001 0010	1,7	4,10	0100 1001 0010	1,7	4,10
2	0	1001 0010 0100	2,8	5,11	1001 0010 0100	2,8	5,11
3	1	0010 0100 1001	3,9	6,0	0010 0100 1001	3,9	6,0
4	1	0100 1001 0010	4,10	7,1	0100 1001 0010	4,10	7,1
5	1	1001 0010 0100	5,11	8,2	1001 0010 0100	5,11	8,2

crsReMask for vShift 0 is same as for vShift 3 case but RE locations are different for Ant 0 and 1. Hence indicating 1 for 3,4,5 vShift cases, thereby O-RU shall pick the positions as interpreted in above table.

Pseudo code to determine the CRS belongs to ANT port-N

```
// Determine vShift = N_CELL_ID % 6
// Below pseudo code for Num Layer = 4
Switch(crsSymNum)
{
Case 0 || case 7:
    if( vShift  == 0 )
    {
      if (bitSet_0 && bitSet_6)
      {
        mapIQ of 0 and 6 to Ant0 //rest blank
      }
      if (bitSet_3 && bitSet_9)
      {
        mapIQ of 0 and 6 to Ant1 //rest blank
      }
    }
    Break;
Case 1:
    if( vShift  == 0 )
    {
        if (bitSet_0 && bitSet_6)
        {
            mapIQ of 0 and 6 to Ant2
        }
        if (bitSet_3 && bitSet_9)
        {
            mapIQ of 3 and 9 to Ant3
        }
    }
    Break;
Case 4 || case 11:
    if( vShift  == 0 )
    {
        if (bitSet_0 && bitSet_6)
        {
            mapIQ of 0 and 6 to Ant1
        }
        if (bitSet_3 && bitSet_9)
        {
            mapIQ of 0 and 6 to Ant0
        }
    }
    Break;
Case 8:
    if( vShift  == 0 )
    {
        if (bitSet_0 && bitSet_6)
        {
            mapIQ of 0 and 6 to Ant3
        }
        if (bitSet_3 && bitSet_9)
        {
            mapIQ of 3 and 9 to Ant2
        }
    }
    Break;
}
```

Annex J (informative): Beamforming Methods Description

Beams are RF energy directed in specific angular directions in space. Beamforming can generate energy lobes (maxima, or “peaks”) &/or energy nulls (minima, or “valleys”) in the spatial dimension. They can be formed using various methods in the analog domain, the digital domain, or a combination thereof. Beams and beamforming can further be used to re-utilize temporo-spectral (Time-Frequency) resources to achieve Spatial Multiplexing.

O-RAN has four distinct methods supported at the O-RU for beamforming.

1. **Predefined-Beam Beamforming.**

In this method, beam indices (“beamId” values) are conveyed from the O-DU to the O-RU to indicate which beam to apply to the DL or UL data. The beams are expected to be pre-defined within the O-RU and the method of beamforming is not specified. However, the O-DU still needs to know whether the beamforming is frequency-domain, time-domain or “hybrid” so the beamId values can be correctly applied e.g. to not apply different time-domain beamIds to subcarriers in the same OFDM symbol.

2. **Weight-based Dynamic Beamforming.**

In this method, beamforming weights are transmitted across the interface using C-Plane messages.

For frequency-domain beamforming the operation may be considered as follows:

Let $\mathrm{X} = [\mathrm{x}_0\ \mathrm{x}_1\ \dots \mathrm{x}_P]$ represent the frequency domain IQ data in one data section, where P is the number of PRBs in the data section and x is a PRB, $x \in \mathbb{C}^{L*1}$, L is the number of streams.

Let $\mathrm{W} = [\mathrm{w}_0\ \mathrm{w}_1\ \dots \mathrm{w}_K]$ represent the beamforming weights associated with the tx-array, where w is a beamforming weight, $\mathrm{w} \in \mathbb{C}^{K*L}$, K is the number of array elements.

The output after beamforming $\mathrm{Y} = [\mathrm{y}_0\ \mathrm{y}_1\ \dots \mathrm{y}_P]$ is given by

$$\mathrm{y} = \mathrm{w}.\mathrm{x}$$

where y is a beamforming output for the PRB x , $\mathrm{y} \in \mathbb{C}^{K*1}$

The equation above refers to DL beamforming but the same principle is applied for UL as well.

For time-domain beamforming the operation is the similar:

Here, however, instead of being applied per data section the beamforming is applied for the entire OFDM symbol.

Let $\mathrm{X} = [\mathrm{x}_0\ \mathrm{x}_1\ \dots \mathrm{x}_P]$ represent the frequency domain IQ data in one OFDM symbol, where P is the number of PRBs in the OFDM symbol and x is a PRB, $x \in \mathbb{C}^{L*1}$, L is the number of streams.

Let $\mathrm{W} = [\mathrm{w}_0\ \mathrm{w}_1\ \dots \mathrm{w}_P]$ represent the beamforming weights associated with one tx-array, where w is a beamforming weight, $\mathrm{w} \in \mathbb{C}^{K*L}$, K is the number of array elements.

The output after beamforming $\mathrm{Y} = [\mathrm{y}_0\ \mathrm{y}_1\ \dots \mathrm{y}_P]$ is given by

$$\mathrm{y} = \mathrm{w}.\mathrm{x}$$

where y is a beamforming output for the PRB x , $\mathrm{y} \in \mathbb{C}^{K*1}$

The equation above refers to DL beamforming but the same principle is applied for UL as well.

For hybrid beamforming the operation involves the multiplication by frequency-domain weights and time-domain weights.

There are K number of time-domain weights (the is the same as the number of array elements) and K’ number of frequency-domain weights (K’ is less than K, see clause 12.5). The K’ frequency-domain weights are applied for the PRBs in a data section, and the K time-domain weights are applied for all the PRBs in the entire OFDM symbol.

Let $\mathrm{X} = [\mathrm{x}_0\ \mathrm{x}_1\ \ldots\ \mathrm{x}_P]$ represent the frequency domain IQ data in one OFDM symbol, where P is the number of PRBs in the OFDM symbol and x is a PRB, $x \in \mathbb{C}^{L*1}$, L is the number of streams.

Let $\mathrm{W} = [\mathrm{w}_0\ \mathrm{w}_1\ \ldots\ \mathrm{w}_P]$ represent the beamforming weights in one OFDM symbol, where w is a time-domain beamforming weight, $\mathrm{w} \in \mathbb{C}^{K*L}$, K is the total number of array elements.

Let $\mathrm{W}' = [\mathrm{w}'_0\ \mathrm{w}'_1\ \ldots\ \mathrm{w}'_{P',}\ \mathrm{w}'_0\ \mathrm{w}'_1\ \ldots\ \mathrm{w}'_{P'}\ \ldots\ \mathrm{w}'_0\ \mathrm{w}'_1\ \ldots\ \mathrm{w}'_{P'}]$ represent the beamforming weights in one data section, where w' is a frequency-domain beamforming weight, $\mathrm{w}' \in \mathbb{C}^{K'*L}$.

Note that $K = K' * p'$ so the W' vector represents K / K' repetitions of the K' frequency-domain weights.,

The output after beamforming $\mathrm{Y} = [\mathrm{y}_0\ \mathrm{y}_1\ \ldots\ \mathrm{y}_P]$ is given by

$$\mathrm{y} = (\mathrm{w} * \mathrm{w}').\,\mathrm{x}$$

where y is a beamforming output for the PRB x , $\mathrm{y} \in \mathbb{C}^{M*1}$

The equation above refers to DL beamforming but the same principle is applied for UL as well.

3. Attribute-Based Dynamic Beamforming.

In this method, beamforming attributes and/or their indices (if already known to the O-RU) are transmitted across the interface.

Whereas a beam *index* provides a pointer to a beamforming vector already known to the O-RU, and beamforming *weights* specify an important method for *how* to form the beam, beamforming *attributes* specify the *what*, an inherently compact characterization of the desired beam pattern itself, to be formed directionally in space.

These beamforming attributes include:

- bfAzPt: the azimuth beamforming pointing angle in degrees
- bfZePt: the zenith beamforming pointing angle in degrees
- bfAz3dd: the azimuth beamforming 3dB down beam width in degrees
- bfZe3dd: the zenith beamforming 3dB down beam width in degrees
- bfAzSl: the azimuth beamforming sidelobe suppression value in dB
- bfZeSl: the zenith beamforming sidelobe suppression value in dB

Multiple methods of forming the beam per the Beam Attributes are possible and are left as O-RU implementation choices. Some potential schemes are: Beamforming phased array weights (Analog or Digital), Holographic Beamforming, Butler Matrices, Lenses, and other known and emerging techniques. These schemes may also be hybridized with method #4 “Channel-Information-Based beamforming”.

4. Channel-Information-Based Beamforming.

In this method, beamforming weights are calculated at the O-RU based on the channel estimates that are transmitted across the interface.

For UL:

Assuming K users who are jointly scheduled for MU-MIMO in UL, a beamforming matrix $W = [w_1,\ldots,w_M] \in \mathbb{C}^{KxM}$ is applied to the frequency domain IQ data for K users.

Let $\mathrm{H} \triangleq [h_1 \ldots, h_K] \in \mathbb{C}\mathrm{M}^{MxK}$ be the UL channel estimates for the K users.

Multiple methods to calculate the beamforming weights for UL are possible and are left as the O-RU implementation choices. Some potential schemes are:

Zero-forcing : $W = (H^H\ H)^{-1}\, H^H$

Regularized zero-forcing/ MMSE: $W = (H^H H + \xi I_K)^{-1} H^H$,

ξ is the regularization parameter.

Note that the C-Plane Section Extension =8 allows configuration of regularization factor per user in UL.

For DL:

Assuming K layers which are jointly scheduled for MU-MIMO in DL, a beamforming matrix $W = [w_1, ..., w_K] \in \mathbb{C}^{MxK}$ is applied to the frequency domain IQ data for K layers.

Let Let $\mathrm{H_{eff}} \triangleq \left[\mathrm{h}_{\mathrm{eff,1}}^T, \ldots, \mathrm{h}_{\mathrm{eff,K}}^T\right]^T \in \mathbb{C}^{\mathrm{KxM}}$ be the DL channel estimates for the K selected layers among L users each transmitting multiple-layer data where $\mathrm{h_{eff,k}} \in \mathbb{C}^{\mathrm{1xM}}$.

.

Multiple methods to calculate the beamforming weights for DL are possible and are left as the O-RU implementation choices. Some potential schemes are:

Zero-forcing: $W = \frac{1}{\sqrt{\Psi}} \mathrm{H}_{\mathrm{eff}}^H (\mathrm{H_{eff}} \mathrm{H_{eff}}{}^H)^{-1}$

Regularized zero-forcing/ MMSE: $W = \frac{1}{\sqrt{\Psi}} \mathrm{H}_{\mathrm{eff}}^H \left(\mathrm{H_{eff}} \mathrm{H_{eff}}{}^H + \xi \mathrm{I_K}\right)^{-1}$,

ξ is the regularization parameter, and the normalization parameter Ψ can be chosen to satisfy the total power constraint $\{WW^{\mathbf{H}}\} \leq M$.

Note that the C-Plane Section Extension =8 allows configuration of regularization factor per user in DL.

Assuming L users' channel matrices each transmitting multiple layers' data considering transmit antenna switching (TAS), so $\mathrm{H_{eff}}$ can be selected by using a proper sub-spacing function $f(\cdot)$ among L user's channel matrices H_i where $i = 1, \ldots, \mathrm{L} \in \mathbb{C}^{\mathrm{N}_i \mathrm{xM}}$ and N_i is the number of antennas of *i*-th user. Define K_i as the number of selected layers for *i*-th user, and then $\mathrm{K} = \sum_{i=1}^{\mathrm{L}} \mathrm{K}_i$ and $\mathrm{H_{eff}} = \left[\mathrm{H}_{\mathrm{eff,K_1}}^T \ldots, \mathrm{H}_{\mathrm{eff,K_L}}^T\right]^T$ where $\mathrm{H}_{\mathrm{eff,K}_i} = f(\mathrm{H}_i) \in \mathbb{C}^{\mathrm{K}_i \mathrm{xM}}$ and $\mathrm{K}_i \leq \mathrm{N}_i$.

Annex K (informative): Layers of Array Elements

K.1 General

Figure K.1 shows an example of a rectangular array with 12 rows and 4 columns of array elements. There are four data converters (not shown in the figure). Each data converter connects to all the 48 array elements (also known as the full-connection model in 3GPP). The array element contains 4 gain and phase control elements each connecting to one of the data converters. The gain and phase control element is used to apply time domain (TD) beamforming weights. This type of connection creates four overlapping arrays (“array-layers”) by reusing the same array elements.

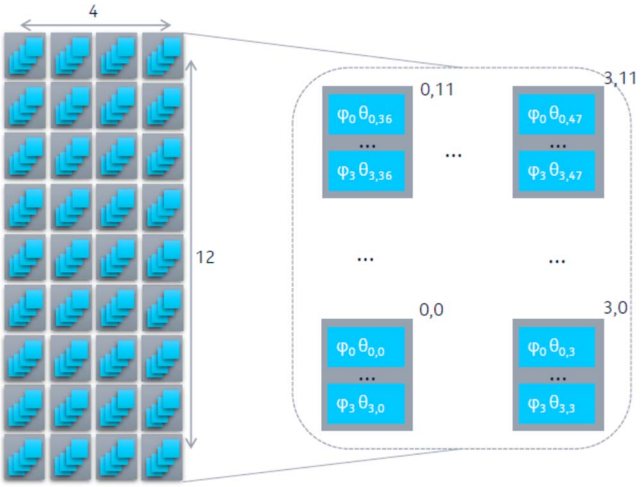


Figure K.1

On the transmit side each DAC can correspond to one layer, which is then connected to one of the overlapping Tx arrays. Hence the structure of the tx-array so constructed can support 4 layers. The 48 elements corresponding to layer q = 0…3 receive the same input from DAC q and this input can be beamformed with a 48 element TD beamforming vector $(\theta_{q,0}, \theta_{q,47})$and a Frequency Domain (FD) beamforming weight φ_q corresponding to layer q. Each array element hence can receive 4 equivalent beamforming weights $\varphi_0\theta_{0,0}to\varphi_3\theta_{3,0}$ corresponding to 4 layers.

K.2 Use Case A

Use case A is to send eAxC via 48 elements. In this case, the eAxC is beamformed in frequency domain (FD) by $\varphi_0$and 48 complex weights corresponding to time domain (TD) using$(\theta_{0,0}, \theta_{0,47})$ see figure K.2. In this case, the O-RU has 4 simple tx-arrays with 48 elements each.

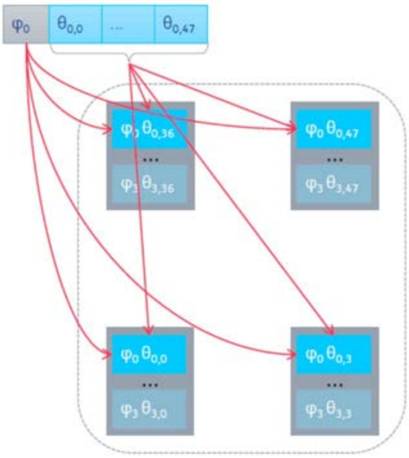


Figure K.2

One should note in the above case that the 48 array elements each have one power amplifier which is shared across the four tx-arrays, conveyed over the M-plane as the array elements being “shared” according to clause 12.5.4

K.3 Use Case B

In this use case, one eAxC signal is sent over all the 4x48 elements. This can be accomplished by using 4 digital weights in FD and 4x48 TD weights (figure K.3). In this case, O-RU has one tx-array with 4x48 elements.

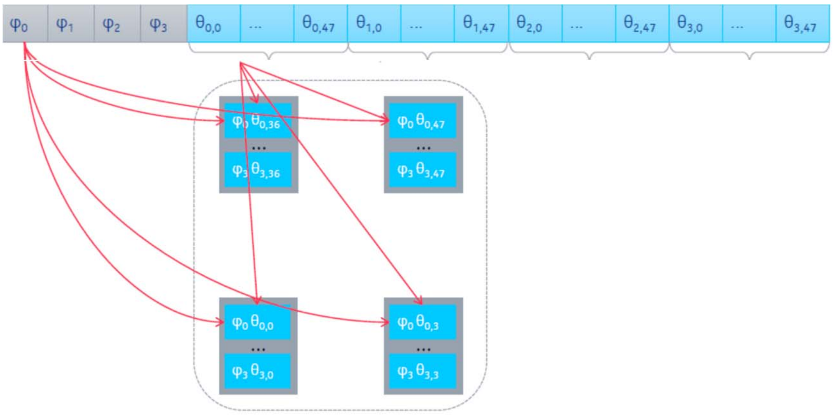


Figure K.3

One should note in the above case that the 48 array elements each have one power amplifier which is shared across the four array-layers *q*=0..3, conveyed over the M-plane as independent-power-budget being false according to clause 12.5.2

Annex L (informative): Considerations when operating in non-ideal transport environments

L.1 General

The Intra-PHY lower layer fronthaul split has been optimized for operation using transport systems that are able to support tight latency and bandwidth requirements. However, this does not preclude operation of the lower layer split in environments where the transport systems are not able to meet tight latency and bandwidth requirements. This appendix is informative and describes those additional capabilities that may be included in an O-DU and/or O-RU to assist when operating in such circumstances.

Many of the following discussion points and examples assume an FDD cell configuration. FDD examples are used here as the timing aspects are easier to conceptualize. The same principles and approaches apply to TDD cells.

L.2 Deployment Scenarios and Performance Considerations

L.2.1 Low Impact

There is a class of low mobility deployment scenarios characterized by slow fading, which provide a relatively stable radio channel at small time scales. When combined with low UE density, these scenarios see low (or no) performance impact from non-ideal fronthaul vRAN deployments.

Example deployment use cases include indoor femto cells, fixed wireless access, and outdoor pico cells in non-dense urban settings.

In such scenarios the benefit from HARQ is negligible, thus the performance impact of disabling HARQ retransmissions or using predictive HARQ is negligible. Further, the impact of a diminished UE attach ramp rate is negligible as well because UE ramp rate is not a performance design attribute for such deployments.

L.2.2 Medium Impact

There is a second class of low mobility deployment scenarios where again, radio channels are predominately slow fading, but where UE density may be higher. In these scenarios, the UE attach ramp rate may be a performance attribute of higher importance and thus a larger impact from non-ideal fronthaul may be expected.

Example deployment use cases include venue deployments (e.g. stadiums or arenas), shopping malls, or airport terminals.

In such scenarios inter-cell interference may become significant and the benefit from HARQ may be greater and thus a noticeable degradation of system throughput would be observed when using non-ideal fronthaul. In addition, delays in NAS attach may become noticeable to users depending on the user behavior given the increased UE density.

L.2.3 High Impact

There also exists a class of deployments which are characterized by high mobility and/or fast fading. In these scenarios the performance impacts of using non-ideal fronthaul may be significant.

Example deployment use cases include freeway coverage macro cells, dense urban macro or small cells, or rural large ISD macros.

In these scenarios, HARQ often provides a benefit and thus prediction algorithms will likely result in a noticeable performance impact. In addition, the impact of delayed CSI feedback will further reduce system efficiency. Similarly, depending on the UE behavior (i.e. traffic movement) UE attach ramp rate may be a metric of higher importance.

L.3 HARQ

Long latency fronthaul links may break traditional Hybrid ARQ timelines. The following clauses discuss the timelines in detail and cover techniques which can be used to minimize the impact of the timeline being broken.

L.3.1 Synchronous HARQ

Since FDD LTE offers 8 HARQ processes for each UE, one UE can be scheduled for PDSCH traffic in up to 8 different subframes simply by associating a different HARQ process to each subframe. Hence, the most demanding scenario in terms of processing timeline is given by a “full-buffer” UE continuously scheduled in consecutive subframes, which results in 8 subframes (i.e., 8 ms) of timeline budget for each HARQ process. Since the standard mandates that each ACK/NACK HARQ response is sent in uplink 4 subframes after the relevant downlink transmission, effectively only 4 ms of timeline budget are left for eNB-side operations.

The eNB is typically expected to complete all tasks within the 4-ms timeline budget, with many eNB implementations taking more than 3 ms in the worst load conditions. It is this argument that has generated the common perception according to which all sources of “extra” latency cannot exceed 0.5-1.0 ms, see figure L.1.

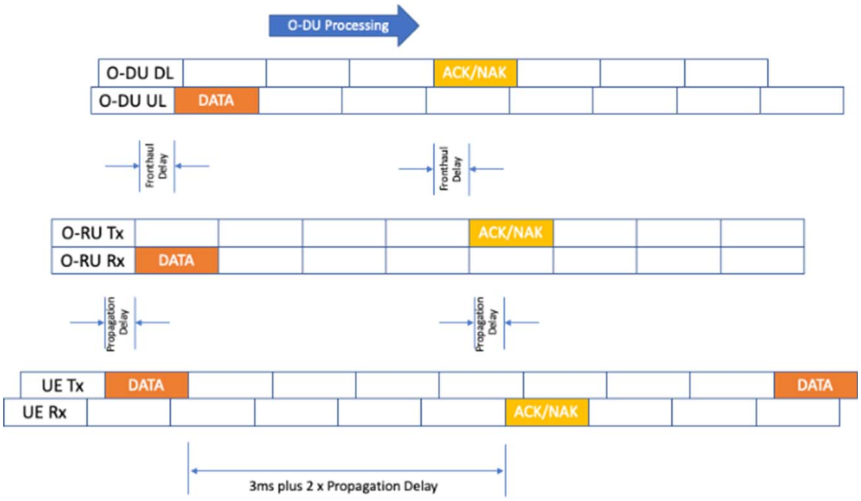


Figure L.1 : Synchronous Up-link HARQ Processing Budget

In the strictest interpretation of this argument, the sub-ms upper bound includes occasional, short-term spikes, which effectively rules out virtual implementations of LTE L1 on commodity operative systems (OSs) and forces adoption of real-time OSs. Even in the broader interpretation, this argument rules out disruptive deployments such as split-7 architectures over high-latency fronthaul.

The following analysis challenges these often-cited assumptions, showing that, while they make sense for many legacy deployments, different scenarios exist in which forcing tight latency constraints is unnecessary and may be counterproductive to the exploration of innovative low-cost deployment scenarios.

L.3.1.1 HARQ Interleaving

One option often mentioned in the context of latency-resistant LTE implementations is “HARQ (process) interleaving”, which in essence requires waiting for the ACK/NACK response to be available before reusing a certain HARQ process. This concept was first introduced in [REF: https://www.bell-labs.com/our-research/publications/200870/].

Using this approach, the tolerated latency for O-DU processing time plus round trip frounthaul delay can be increased to 3 + n8 milli-seconds, where n is number of autonoms HARQ responses sent with the New Data Indication (NDI) not toggled, ensuring that the UE re-transmits the data until it can be successfully ACK’s/NACK’d by the O-DU.

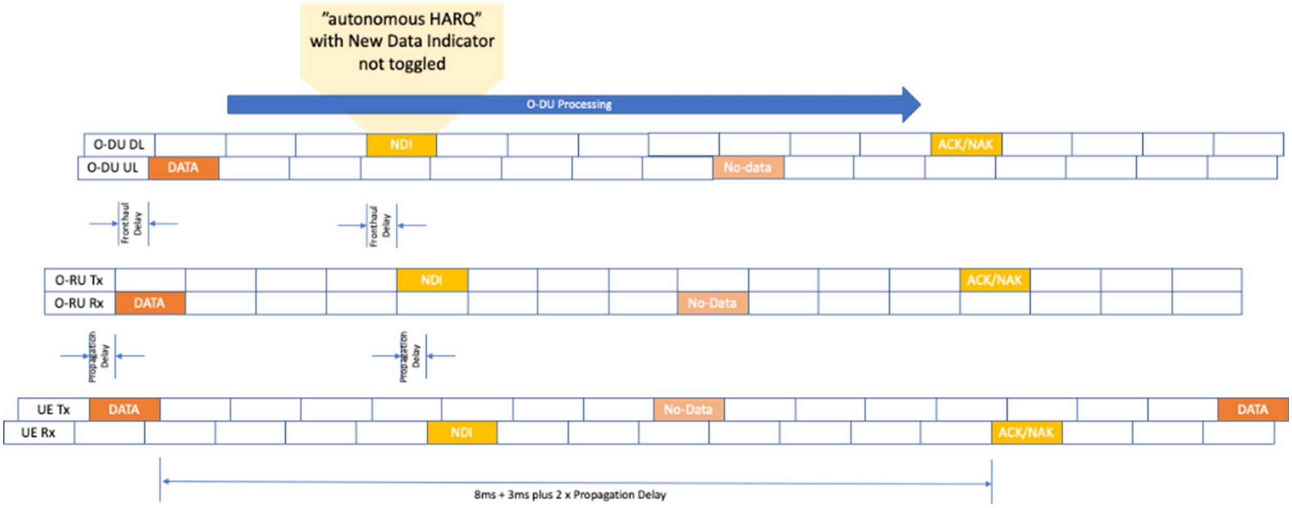


Figure L.2: Interleaved HARQ Operation

In the worst case in terms of timeline budget -- that of a single, full-buffer UE -- this option is very inefficient, resulting in only 8 every (8+n) subframes being used when the overall latency exceeds the 8-ms budget by n ms. For example, 8 ms of extra latency would lead to about 50% of the subframes being unused. Hence, while possibly appealing in deployments with latency typically below the desired limit except for a few occasional spikes, this option is certainly not viable for deployments in which high latency is the norm, such as split-7 architectures over high-latency fronthaul.

L.3.1.2 Predictive HARQ

Another option consists of, effectively, turning off HARQ, relying on upper layers for retransmissions (e.g., RLC-AM, or transport/application layer for RLC-TM and RLC-UM). In this technique, the eNB schedules each user as it would if it had received a timely ACK response for each PDSCH transmission, which results in no subframe left unnecessarily unused.

When the late HARQ response is actually received, two cases are possible. In one case, when the response is ACK, nothing else needs to be done since "prediction of ACK" was correct -- the eNB did everything correctly, without wasting time waiting for the actual response.

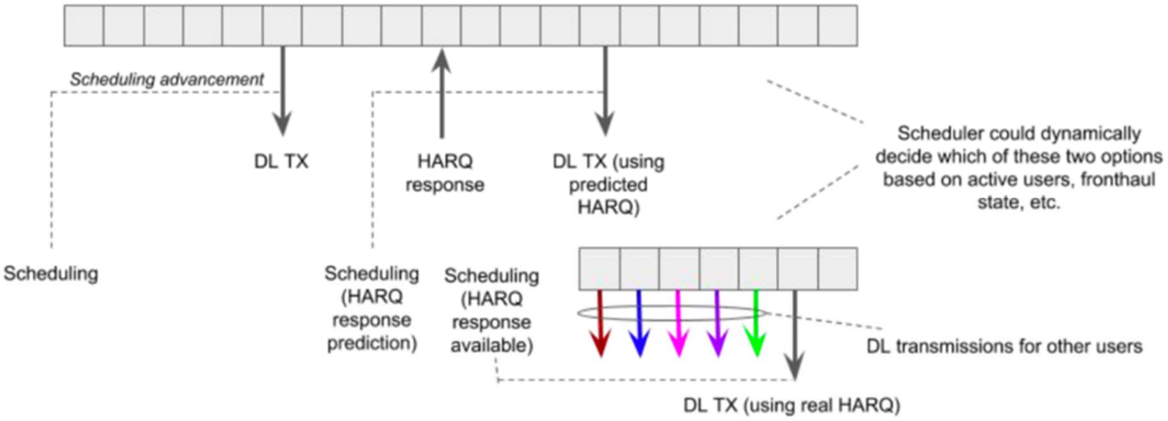

Scheduling advancement
DL TX
HARQ response
DL TX (using predicted HARQ)
Scheduler could dynamically decide which of these two options based on active users, fronthaul state, etc.
Scheduling
Scheduling (HARQ response prediction)
Scheduling (HARQ response available)
DL transmissions for other users
DL TX (using real HARQ)

Figure L.3: DL Predictive HARQ Operation

In the other case, when the response is NACK and the prediction was wrong, nothing else needs to be done, since this event is equivalent to an ACK-for-NACK decoding error, hence equivalent to an event that is typically caused by radio problems (e.g., fading, interference) against which LTE already has built-in robustness.

This simple technique performs well when NACK events are rare, most notably in peak-rate conditions, when NACK events are essentially absent even at maximum MCS. With proper tuning of the link adaptation parameters (e.g., lower target block error rate), it is possible to achieve full-buffer throughput performance within 10-15% of what an optimal, low-latency eNB can achieve in any given static/pedestrian radio conditions.

It is evident that, for these low-mobility scenarios, ruling out high-latency deployments is unnecessary: Many use cases exist in which a 10-15% throughput degradation does not justify more expensive infrastructures.

More advanced algorithms may be required for optimization of scenarios in which full-buffer throughput is not the key metric (e.g., RRC messages not protected by RLC-AM). For example, “prediction of NACK” may be be used until an ACK response is actually received for a certain critical message.

While discussions of advanced techniques are beyond the scope of this clause -- and the scope of fronthaul protocol specifications -- it is critical to remark that innovative eNB vendors have the opportunity to unlock new use cases, using the longer latency O-DU and O-RU categories.

L.3.2 Asynchronous HARQ Considerations

In the LTE down-link and in both 5G down-link and up-link, HARQ has been defined to be asynchronous. This means that a separate HARQ process identifier is included in the HARQ messages, thus avoiding the processing time limitations associated with synchronous up-link LTE operation.

L.4 RACH Considerations

The LTE attach procedure starts with UEs performing a preamble transmission on the random access channel to identifying itself to the network. This call flow is usually described as “message 1” through “message 4” as:

1) RACH preamble from UE

2) Random Access Response (RAR) from MAC

3) PUSCH message from UE containing CCCH or MAC signaling

4) Contention resolution Identity MAC CE for contention-based RACH procedure

Following are the timers used in RACH procedure (specified as per 3GPP TS 36.331) apart from the PRACH configuration.

```
       ra-ResponseWindowSize              ENUMERATED {
                                            sf2, sf3, sf4, sf5, sf6, sf7,sf8, sf10},
       mac-ContentionResolutionTimer      ENUMERATED {
                                            sf8, sf16, sf24, sf32, sf40, sf48, sf56, sf64}
```

PRACH configuration allows the RACH occasions for UEs to be available in every UL subframe to every 20 UL subframes.

Even if the above timers are set to maximum value, timers would pose an issue with RACH handling for long latency fronthaul transport links. In particular, the ra-ResponseWindowSize timer maximum value is 10ms (with timer starting 3 subframes after RACH transmission) from RACH preamble transmission occasion. During this time the vRAN system needs to handle the steps including

1) Complete L1 processing of RACH detection,

2) Sending decoded RACH message to MAC (including fronthaul transport latency)

3) MAC processing to allocate DL (msg2, RAR) and UL (msg3 grant) resources

4) Relaying of messages back to L1 (including fronthaul transport latency)

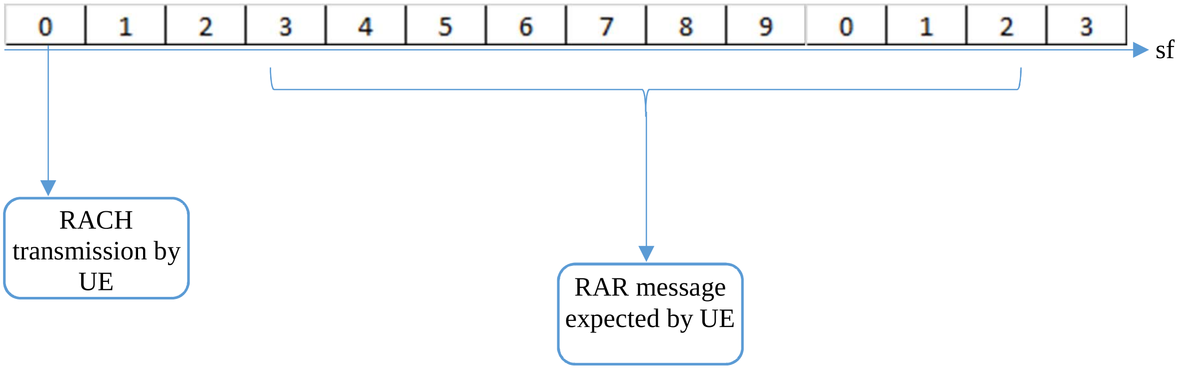


Figure L.4: Traditional RACH Timing Diagram

Therefore, in this process, there will be a 2-way delay (first in UL for RACH reception at MAC and second for sending RAR from MAC to L1) apart from the L1/MAC processing time. Without adjust to accommodate long latency fronthaul, this would lead to RACH procedure failure and UE will reattempt the RACH again (and would lead to same problem again) leading to UE not able to access the network.

With high fronthaul latency, alternative options are required to handle the RACH procedure. Two such options are discussed below.

L.4.1 Non-Ideal Fronthaul RACH Designs

When designing the RACH process for non-ideal, long latency fronthaul the following points are considered:

1) The method should work assuming existing 3GPP specification

2) There should not be any impact to UE and 3GPP compliant UE should work with the method

3) Currently PRACH Format-0 is considered in RACH analysis to consider FH latency requirements. Support of different Preamble formats need to be considered in future versions

4) Current clause considers the msg2 handling to meet the FH latency requirements

5) Contention Based RACH procedure is considered in current document. Contention Free RACH procedure needs to be considered in future versions

6) BI (Backoff indicator) not considered

7) HARQ timings for Msg3 and msg4 HARQ are considered in clause L.3.

8) FDD Duplexing mode is considered

L.4.1.1 Option-1: Semi-persistent Resource allocation

Option 1 presents a semi-persistent allocation of resources that allows the O-DU to make RACH decisions without waiting for the reception of a RACH preamble. In this alternative, MAC allocates three types of resource for RAR messages:

1) PDCCH resources for RAR

a) RA-RNTI (0 – 9) being fixed and one-to-one mapping for RACH occasions in FDD mode

b) Common search space resources for PDCCH carrying RAR (corresponding RA-RNTI)

c) Based on configured RACH configuration (RACH occasions), MAC reserves the PDCCH resources corresponding to RAR message in all required TTIs (For example, n+3rd TTI from every RACH preamble occasion).

d) The reservation of resources will be done at MAC regardless of the RACH triggered by UE

e) DCI Format 1A

2) PDSCH resources for RAR

a) Configured number of RACH to be entertained by MAC in one subframe

b) Based on “Number of RACH” to be processed, MAC estimates the size of RAR PDU

c) MAC reserves the PDSCH resources in all required TTIs similar to PDCCH

d) Layer-1 updates the TC-RNTI (A pool of TC-RNTI available with L1 for CBR RACH) in RAR along with RAP-ID (Preamble ID) and TA value in RAR message (decoding of MAC PDU sent by MAC)

e) Layer-1 maintains the TC-RNTI pool

3) PUSCH resources for Msg3

a) Based on “Number of RACH” to be processed in a subframe, MAC reserves the PUSCH resources

4) All the above information (reserved by MAC) to L1 at cell configuration time. MAC will directly receive the PUSCH (Rx_PUSCH.Indication) with TCRNTI (and corresponding subframe number) and TA value applied in RAR

5) Considering maximum value of contention resolution timer, MAC will then normally schedule msg4.

6) After successful msg4 procedure, MAC will start using C-RNTI

7) In case of contention failure or UE release, MAC will inform Layer-1 to free the TC-RNTI

Disadvantage: This would waste PDCCH/PDSCH/PUSCH resources if there is no RACH procedure triggered. In addition hard coding of the timing advance is an acceptable approach for small cell deployments, but in macro cells this will not work.

L.4.1.2 Option-2: RACH retransmission estimation

The following description addresses the flow driving message 2 scheduling in a basic configuration -- different, more complex flows exist.

For simplicity, the description refers to an FDD LTE deployment with RACH configured with the greatest possible values for the maximum number of preamble retransmissions (50), the preamble periodicity (20 subframes) and the RAR window size (10 subframes).

- The O-DU stack detects that a RACH preamble was transmitted at UL subframe t.
- If the scheduling in advance (which is a function of the fronthaul latency) is such that the relevant message 2 can be transmitted over the air at DL subframe t+12 or earlier, the "regular" flow is possible. In particular, a message 2 with content corresponding to the preamble detected in UL subframe t is transmitted by DL subframe t+12.
- Else, the relevant message 2 cannot be delivered within the valid RAR window, in which case the terminal that transmitted a preamble at UL subframe t is mandated to transmit another preamble at UL subframe t+20, for which the valid RAR window ends at DL subframe t+32. In this case, O-DU skips the scheduling of the message 2 corresponding to the preamble transmitted at UL subframe t (since it would not be received within the valid RAR window) and directly schedules for DL subframe t+32 the message 2 corresponding to the preamble transmitted at UL subframe t+20, even before receiving and processing the samples for said UL subframe.

Note that the flow described above assumes that the "effective fronthaul round-trip delay" is lower than 32 ms, that is, assumes that it is possible for O-DU to process the samples relative to UL subframe t and to compute the relevant response, and for the O-RU(s) to transmit said response at DL subframe t+32.

A simple extension of this flow makes it viable to complete the RACH procedure in the presence of an "effective fronthaul round-trip delay" as high as 52 ms, or even 72 ms, but deployments over such high-latency fronthauls are not recommended.

Some details are omitted in the description above to maintain the flow presentation as simple as possible. In practice, the latency-resistant RACH procedure is made slightly more complex by the fact that the terminal choses the preamble ID randomly at each retransmission, and that the content of message 2 has to include the preamble ID of the last transmission.

In the example above, at the time of scheduling message 2 for DL subframe t+32 (which refers to the preamble transmitted in UL subframe t+20), O-DU doesn't know what preamble ID to include in that message, since it has not processed yet the samples relative to said preamble transmission. The simplest method to this is to populate message 2 with the last seen preamble.

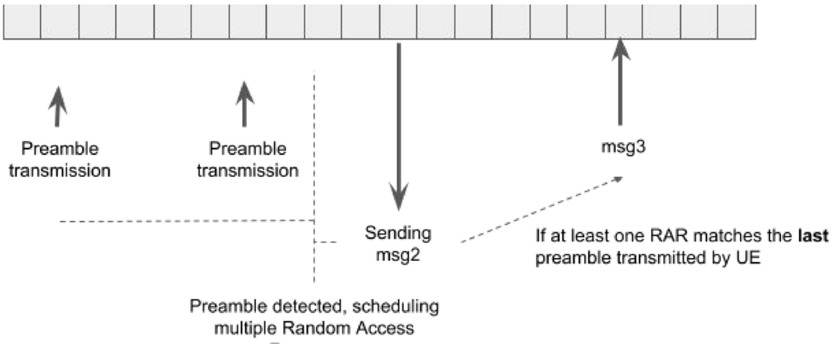

Preamble transmission
Preamble transmission
msg3
Sending msg2
If at least one RAR matches the **last** preamble transmitted by UE
Preamble detected, scheduling multiple Random Access

Figure L.5: Long Latency RAR “Collision” Operation

By limiting the number of available preambles, the likelihood two subsequent UE RACH attempts using the same preamble can increased sufficiently to ensure successful RACH in a timely manner.

L.5 Other Latency Related Considerations

Supporting high latency requires careful system design to ensure seamless device connectivity while maintaining spectrum efficiency. In this clause, we highlight other system aspects that are impacted by high fronthaul latency. Some of the issues can be resolved with upper layer protocols while some may not be relevant for certain deployment use cases. Proposed solutions for these issues are for future study to see if they require any additional information to be transmitted over the fronthaul interface.

- Link adaptation and scheduling
  - Link adaptation is implemented at layer 2, where the eNB adapts in real-time the UEs’ MCS based on the CQI. Scheduling (e.g., proportional fair scheduling) depends heavily on the channel quality between the eNB and the UE. Although this depends on the channel model, it has to be done on a relatively short time-scale. Link adaptation and scheduling have to be conservative for non-ideal fronthaul with split 7 since the channel can change significantly in the time period. This may reduce the benefit of frequency selective scheduling and high mobility support since it takes longer for the scheduler to respond to the channel variations under fading conditions. However, this may not be an issue for low mobility scenarios in a small cell environment.
- UL power control
  - Another challenging issue for non-ideal fronthaul is UL power control. This also becomes challenging under high mobility and large latencies since it takes longer time to feedback this information to the scheduler to adjust the UE transmit power. However, for use cases with limited mobility, this may not be an issue.
- Beamforming and MU-MIMO
  - Beamforming is becoming an important use case for interference mitigation and for multi-user MIMO support for LTE and NR for macros. This requires close interaction with the scheduler in order to

suppress interference and pair users. If there is significant latency on the fronthaul, the beamforming and multi-user MIMO support becomes challenging under non-ideal conditions. This can be explored further. However, there are use cases with 2TRX/4TRX radios where beamforming support is not critical.

- COMP and advanced receiver support
  - One of the main advantages of split 7 is centralized processing for features such as UL COMP and advanced receivers since signals from multiple base-stations can be combined and processed jointly. However, under split 7 non-ideal fronthaul, these gains are not straightforward to attain in general since the latencies on the different links can have significant variations. Hence, innovation is needed to help achieve these gains. However, there can be mechanisms to exploit these gains via other means (e.g. solutions developed for inter-site COMP which can accommodate more latency). This is also not a requirement for all use cases.
- UE attach latency
  - The UE attach process involves 5 steps of message exchanges between the eNB and the UE. While there are messages also dependent on the core network and RRC latency, most of these messages are handled by the MAC/PHY layers and this has an impact on UE attach time, specially under large latencies. Again, this may not be relevant for low mobility use cases such as small cells.

To summarize, there are several other factors to consider for high latency support. However, such factors need not be relevant for all use cases and can be explored further in future versions.

L.6 Bandwidth Limitation Considerations

In addition to accommodating extra latency in the fronthaul transport link, implementations may also support variable bandwidth as another aspect of non-ideal fronthaul transport. At a high level, two bandwidth related scenarios can exist; peak rate may be limited, or bandwidth resources may be shared and thus fluctuate over time. In either case, fronthaul implementations can be designed to handle these conditions.

To support limited or variable bandwidth transport links, the O-DU will need to produce estimates of the uplink and downlink available bandwidth at periodic intervals e.g. on a per slot or per TTI interval. The O-DU can then use these bandwidth estimates to inform the scheduling decisions.

In particular a bandwidth estimation function can simply return available bandwidth values in terms of bits per second in the next time interval. Alternatively, available bandwidth could also be expressed as a function of the following parameters including but not limited to:

- Real-time IQ sample variable-bit-width
- IQ Compression
- Variable bit width per channel
- MCS / Constellation of data
- Beam count / compression
- Beam forming related signaling overhead
- C-Plane signaling overhead (reMask, symInc support, etc)
- Number of PRBs scheduled

An example API implementation may take as inputs some or all o the above parameters and return simply *numPrbu* available in the next interval.

The design goal in such implementations is to keep the cell up and active but limit the air interface user plane bandwidth to match the available fronthaul resources. Fronthaul data associated with air interface broadcast and control/reference signaling occupies a small fraction of the bandwidth needed to achieve full cell throughput. An example implementation of a fronthaul aware scheduler could prioritize broadcast and control signaling, allocating PDSCH and PUSCH resources only after fronthaul bandwidth resources for all broadcast and control signaling has been removed from the available bandwidth estimate.

The exact fronthaul aware scheduling algorithm used is out of scope for this specification.

Annex M (informative): Use Case of Selective Transmission and Reception

M.1 General

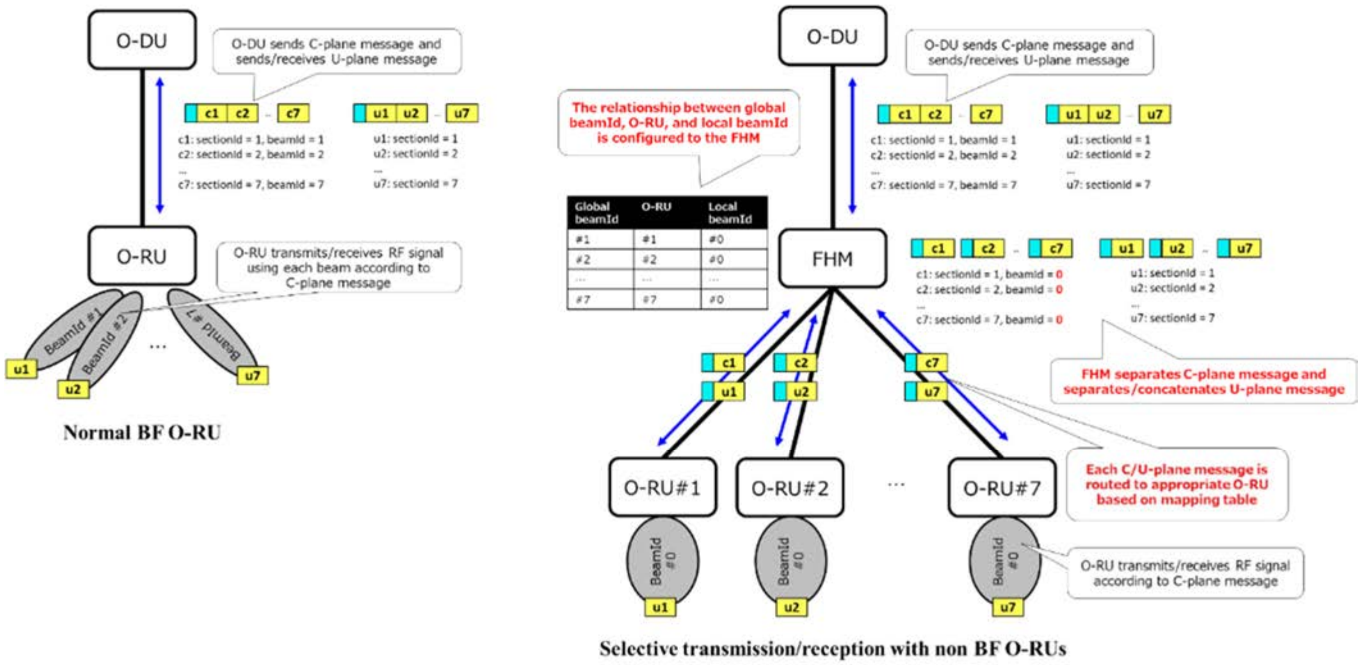
In order to clarify how to transmit/receive signal when selective transmission and reception is used, some concrete examples of selective transmission and reception with beamId are illustrated in this clause.

1) Selective transmission and reception with non beamforming O-RUs and 1-to-1 mapping of global beamId and local beamId

2) Selective transmission and reception with beamforming O-RUs and 1-to-N mapping of global beamId and local beamId

M.2 Selective transmission and reception with non beamforming O-RUs and 1-to-1 mapping of global beamId and local beamId

In this clause, it is assumed that the total number of beams is seven. Figure M.1 shows comparison of C/U-plane processing and RF signal transmission/reception procedure between normal beamforming O-RU case with seven beam and selective transmission and reception with seven non beamforming O-RUs case.



O-DU sends C-plane message and sends/receives U-plane message

c1: sectionId = 1, beamId = 1
c2: sectionId = 2, beamId = 2
...
c7: sectionId = 7, beamId = 7

u1: sectionId = 1
u2: sectionId = 2
...
u7: sectionId = 7

O-RU transmits/receives RF signal using each beam according to C-plane message

The relationship between global beamId, O-RU, and local beamId is configured to the FHM

Global beamId	O-RU	Local beamId
#1	#1	#0
#2	#2	#0
...	...	...
#7	#7	#0

O-DU sends C-plane message and sends/receives U-plane message

c1: sectionId = 1, beamId = 1
c2: sectionId = 2, beamId = 2
...
c7: sectionId = 7, beamId = 7

u1: sectionId = 1
u2: sectionId = 2
...
u7: sectionId = 7

c1: sectionId = 1, beamId = 0
c2: sectionId = 2, beamId = 0
...
c7: sectionId = 7, beamId = 0

u1: sectionId = 1
u2: sectionId = 2
...
u7: sectionId = 7

FHM separates C-plane message and separates/concatenates U-plane message

Each C/U-plane message is routed to appropriate O-RU based on mapping table

O-RU transmits/receives RF signal according to C-plane message

Figure M.1: C/U-plane processing and RF signal transmission/reception procedure comparison (non BF O-RU case)

First of all, the mapping table between global beamId/O-RU/local beamId is configured via M-Plane during start-up procedure. Table M-1 shows the mapping table for this example scenario. The mapping table is assumed to be created by operator based on what an operator would like to operate.

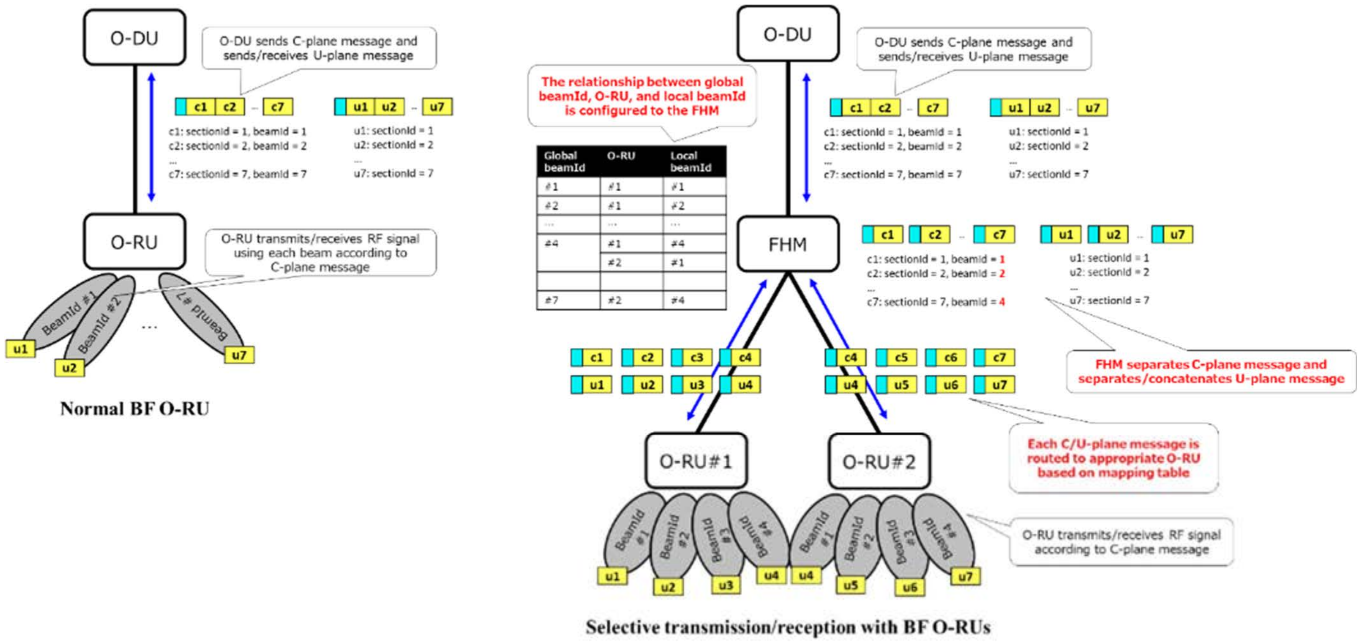
Table M.1: Global beamId/O-RU/local beamId mapping table

Global beamId	O-RU	Local beamId
Beam ID #1	O-RU #1	Beam ID #0
Beam ID #2	O-RU #2	Beam ID #0
Beam ID #3	O-RU #3	Beam ID #0
Beam ID #4	O-RU #4	Beam ID #0
Beam ID #5	O-RU #5	Beam ID #0
Beam ID #6	O-RU #6	Beam ID #0
Beam ID #7	O-RU #7	Beam ID #0

When SSB is transmitted, O-DU sends C-Plane message with beamId corresponds to a transmitting SSB before U-Plane transmission. FHM reads C-Plane header and routes it to corresponding O-RU(s). After that, U-Plane packet is routed in same manner and then O-RU transmits SSB. When the UE receives SSB, UE can obtain each SSB ID by decoding PBCH included in the SSB. If UE is before attaching, UE will transmit PRACH preamble on the PRACH preamble occasion correspondig to the best SSB ID. Since PRACH occasion is linked to SSB ID and SSB ID is linked to global beamId one by one, O-DU knows which beam shall be used to receive PRACH preamble on each PRACH occasion. Therefore, O-DU can sned C-Plane packet with appropriate beamId to receive PRACH preamble on each PRACH occasion. The C-Plane packet is routed to O-RU(s) by FHM and then the O-RU(s) try to receive PRACH preamble and sends U-Plane packet to FHM. After UE attaching, other shared channel and reference signal are transmitted/received as same manner.

M.3 Selective transmission and reception with beamforming O-RUs and 1-to-N mapping of global beamId and local beamId

In this clause, it is assumed that the total number of beams is seven. Figure M.2 shows comparison of C/U-plane processing and RF signal transmission/reception procedure between normal beamforming O-RU case with seven beam and selective transmission and reception with two beamforming O-RUs case. In this case, each beamforming O-RU for selective transmission and reception has 4 beams. Local beamId #4 of O-RU #1 and local beamId #1 of O-RU #2 are mapped to the same global beamId #4.



O-DU sends C-plane message and sends/receives U-plane message

c1: sectionId = 1, beamId = 1
c2: sectionId = 2, beamId = 2
...
c7: sectionId = 7, beamId = 7

u1: sectionId = 1
u2: sectionId = 2
...
u7: sectionId = 7

O-RU transmits/receives RF signal using each beam according to C-plane message

Normal BF O-RU

The relationship between global beamId, O-RU, and local beamId is configured to the FHM

Global beamId	O-RU	Local beamId
#1	#1	#1
#2	#1	#2
...	...	...
#4	#1	#4
	#2	#1
#7	#2	#4

O-DU sends C-plane message and sends/receives U-plane message

c1: sectionId = 1, beamId = 1
c2: sectionId = 2, beamId = 2
...
c7: sectionId = 7, beamId = 7

u1: sectionId = 1
u2: sectionId = 2
...
u7: sectionId = 7

c1: sectionId = 1, beamId = 1
c2: sectionId = 2, beamId = 2
...
c7: sectionId = 7, beamId = 4

u1: sectionId = 1
u2: sectionId = 2
...
u7: sectionId = 7

FHM separates C-plane message and separates/concatenates U-plane message

Each C/U-plane message is routed to appropriate O-RU based on mapping table

O-RU transmits/receives RF signal according to C-plane message

Selective transmission/reception with BF O-RUs

Figure M.2: C/U-plane processing and RF signal transmission/reception procedure comparison (BF O-RU case)

First of all, the mapping table between global beamId/O-RU/local beamId is configured via M-Plane during start-up procedure. Table M-2 shows the mapping table for this example scenario. The mapping table is assumed to be created by operator based on what an operator would like to operate.

Table M.2: Global beamId/O-RU/local beamId mapping table

Global beamId	O-RU	Local beamId
Beam ID #1	O-RU #1	Beam ID #1
Beam ID #2	O-RU #1	Beam ID #2
Beam ID #3	O-RU #1	Beam ID #3
Beam ID #4	O-RU #1	Beam ID #4
	O-RU #2	Beam ID #1
Beam ID #5	O-RU #2	Beam ID #2
Beam ID #6	O-RU #2	Beam ID #3
Beam ID #7	O-RU #2	Beam ID #4

As shown in Table M.2, local beamId #4 of O-RU #1 and local beamId #1 of O-RU #2 are mapped to the same global beamId #4. This means O-RU #1 and O-RU #2 partially construct shared cell area which is coverd by beam ID #4 of O-RU #1 and beam ID #1 of O-RU #2. Therefore copy and combine function is applied for the signal transmit to/receive from this area. The detailed procedure of fronthaul interface and physical layer for transmitting SSB, receiving PRACH preamble, and for transmitting/receiving other shared channel are same as described in clause M.2.

Change History

date	Version	Information about changes
2019.03.11	01.00	First version 01.00
2019.08.02	02.00	Document version 02.00 approved by WG4
2020.03.26	03.00	Document version 03.00 approved by WG4
2020.07.30	04.00	Document version 04.00 approved by WG4
2020.11.19	05.00	Document version 05.00 approved by WG4
2021.03.12	06.00	Document version 06.00 approved by WG4
2021.07.20	07.00	Document version 07.00 approved by WG4
2021.10.18	07.00.01	Created initial version, with change-marks from previous content. Implemented CRs: NOK-0085, NOK-0086
2021.10.27	07.00.02	Resolution of rewview comments: #1,3, 8-61, 63-66, 68, and 69
2021.10.28	07.01	Incorporated review comments #5, 62, 67 and 71. Document approved by WG4

History

Document history		
V7.0.2	September 2022	Publication