

1 **IEEE P1914.3™/D2.1** 2 **Draft Standard for Radio over** 3 **Ethernet Encapsulations and** 4 **Mappings**

5 Sponsor

6 **Standards Development Board**
7 of the
8 **IEEE Communications Society**

9 Approved <XX MONTH 20XX>

10 **IEEE-SA Standards Board**

11 Copyright © 2017 by the Institute of Electrical and Electronics Engineers, Inc.
12 Three Park Avenue
13 New York, New York 10016-5997, USA

14 All rights reserved.

15 This document is an unapproved draft of a proposed IEEE Standard. As such, this
16 document is subject to change. **USE AT YOUR OWN RISK!** Because this is an
17 unapproved draft, this document must not be utilized for any conformance/compliance
18 purposes. Permission is hereby granted for IEEE Standards Committee participants to
19 reproduce this document for purposes of international standardization consideration. Prior
20 to adoption of this document, in whole or in part, by another standards development
21 organization, permission must first be obtained from the IEEE Standards Activities
22 Department (stds.ipr@ieee.org). Other entities seeking permission to reproduce this
23 document, in whole or in part, must also obtain permission from the IEEE Standards
24 Activities Department.

25 IEEE Standards Activities Department
26 445 Hoes Lane
27 Piscataway, NJ 08854, USA
28

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied **“AS IS.”**

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation, or every ten years for stabilization. When a document is more than five years old and has not been reaffirmed, or more than ten years old and has not been stabilized, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon his or her independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration. A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal interpretation of the IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Recommendations to change the status of a stabilized standard should include a rationale as to why a revision or withdrawal is required. Comments and recommendations on standards, and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854
USA

Authorization to photocopy portions of any individual standard for internal or personal use is granted by The Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

1 Introduction

2 This introduction is not part of IEEE P1914.3

3 *[/[/Editor's note: This standard TBD ...]*

4 Notice to users

5 Laws and regulations

6 Users of IEEE Standards documents should consult all applicable laws and regulations.
 7 Compliance with the provisions of any IEEE Standards document does not imply
 8 compliance to any applicable regulatory requirements. Implementers of the standard are
 9 responsible for observing or referring to the applicable regulatory requirements. IEEE does
 10 not, by the publication of its standards, intend to urge action that is not in compliance with
 11 applicable laws, and these documents may not be construed as doing so.

12 Copyrights

13 This document is copyrighted by the IEEE. It is made available for a wide variety of both
 14 public and private uses. These include both use, by reference, in laws and regulations, and
 15 use in private self-regulation, standardization, and the promotion of engineering practices
 16 and methods. By making this document available for use and adoption by public authorities
 17 and private users, the IEEE does not waive any rights in copyright to this document.

18 Updating of IEEE documents

19 Users of IEEE Standards documents should be aware that these documents may be
 20 superseded at any time by the issuance of new editions or may be amended from time to
 21 time through the issuance of amendments, corrigenda, or errata. An official IEEE document
 22 at any point in time consists of the current edition of the document together with any
 23 amendments, corrigenda, or errata then in effect. In order to determine whether a given
 24 document is the current edition and whether it has been amended through the issuance of
 25 amendments, corrigenda, or errata, visit the IEEE-SA Website at
 26 <http://standards.ieee.org/index.html> or contact the IEEE at the address listed previously.
 27 For more information about the IEEE Standards Association or the IEEE standards
 28 development process, visit the IEEE-SA Website at <http://standards.ieee.org/index.html>.

29 Errata

30 Errata, if any, for this and all other standards can be accessed at the following URL:
 31 <http://standards.ieee.org/findstds/errata/index.html>. Users are encouraged to check this
 32 URL for errata periodically.

1 **Interpretations**

2 Current interpretations can be accessed at the following URL:
3 <http://standards.ieee.org/findstds/interps/index.html>.

4 **Patents**

5 Attention is called to the possibility that implementation of this standard may require use
6 of subject matter covered by patent rights. By publication of this standard, no position is
7 taken by the IEEE with respect to the existence or validity of any patent rights in connection
8 therewith. If a patent holder or patent applicant has filed a statement of assurance via an
9 Accepted Letter of Assurance, then the statement is listed on the IEEE-SA website
10 <http://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may
11 indicate whether the Submitter is willing or unwilling to grant licenses under patent rights
12 without compensation or under reasonable rates, with reasonable terms and conditions that
13 are demonstrably free of any unfair discrimination to applicants desiring to obtain such
14 licenses.

15 Essential Patent Claims may exist for which a Letter of Assurance has not been received.
16 The IEEE is not responsible for identifying Essential Patent Claims for which a license
17 may be required, for conducting inquiries into the legal validity or scope of Patents Claims,
18 or determining whether any licensing terms or conditions provided in connection with
19 submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable
20 or nondiscriminatory. Users of this standard are expressly advised that determination of the
21 validity of any patent rights, and the risk of infringement of such rights, is entirely their
22 own responsibility. Further information may be obtained from the IEEE Standards
23 Association.

24

1 Participants

2 At the time this draft standard was submitted to the IEEE-SA Standards Board for approval,
3 the following is a place holder:

4 *, Working Group Chair*

5 *, Editor*

6 The following individuals submitted technical contributions or commented on the draft
7 standard at various stages of the project development.

8

9

10 Name 11

12

13 The following members of the <individual/entity> balloting committee voted on this
14 standard. Balloters may have voted for approval, disapproval, or abstention.

15

16 *(to be supplied by IEEE)*

17

18 Balloter1	21 Balloter4	24 Balloter7
19 Balloter2	22 Balloter5	25 Balloter8
20 Balloter3	23 Balloter6	26 Balloter9

27

28

29 When the IEEE-SA Standards Board approved this standard on <XX MONTH 20XX>, it
30 had the following membership:

31 *(to be supplied by IEEE)*

32

<Name>, Chair

33

<Name>, Vice Chair

34

<Name>, Past President

35

<Name>, Secretary

36

37 SBMember1
38 SBMember2
39 SBMember3
40 SBMember4
41 SBMember5
42 SBMember6
43 SBMember7
44 SBMember8
45 SBMember9

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

<Name>, NRC Representative

<Name>, DOE Representative

<Name>, NIST Representative

<Name>

IEEE Standards Program Manager, Document Development

<Name>

IEEE Standards Program Manager, Technical Program Development

1	Contents	
2		
3	1 OVERVIEW	12
4	1.1 Scope	12
5	1.2 Purpose	12
6	2 NORMATIVE REFERENCES	14
7	3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS.....	15
8	3.1 Acronyms and abbreviations	15
9	3.2 Definitions	17
10	3.3 Typeface conventions	17
11	3.4 RoE Ethernet type	17
12	3.5 Bit, octet ordering and numerical presentation	18
13	4 RADIO OVER ETHERNET (ROE).....	19
14	4.1 Radio over Ethernet (RoE) architecture.....	19
15	4.1.1 Topologies.....	21
16	4.1.2 Underlying network (Informative).....	22
17	4.1.3 Synchronizing nodes (Informative).....	23
18	4.2 RoE objects	23
19	4.2.1 Ethernet link.....	24
20	4.2.2 RoE mappers / RoE de-mappers	24
21	4.2.3 CPRI ports.....	25
22	4.3 RoE traffic types.....	25
23	4.4 RoE common frame format.....	25
24	4.4.1 Sub type (subType) field	25
25	4.4.2 Flow identifier (flowID) field	26
26	4.4.3 Length (length) field.....	27
27	4.4.4 Ordering information (orderInfo)	27
28	4.4.4.1 Sequence number (seqNum) field	27
29	4.4.4.2 Timestamp (timeStamp) field	30
30	4.4.5 Payload field.....	31

1	5	TIMING AND SYNCHRONIZATION CONSIDERATIONS.....	32
2	5.1	General assumptions	32
3	5.2	RoE presentation time	32
4	5.3	Presentation time measurement points	32
5	6	ROE PARAMETERS	34
6	6.1	Parameter introduction	34
7	6.1.1	Parameter defaults	34
8	6.1.2	Parameter set nomenclature.....	34
9	6.1.3	Parameter exchange and enumeration.....	35
10	6.2	Parameter lists.....	35
11	6.2.1	Encryption	35
12	6.2.2	Compression	36
13	6.2.3	Ethernet link parameters	36
14	6.2.4	CPRI port parameters.....	36
15	6.2.5	RoE mapper / RoE de-mapper parameters	37
16	6.2.6	RoE mapper parameters	37
17	6.2.7	RoE de-mapper parameters.....	40
18	7	ROE MAPPERS	43
19	7.1	Structure-agnostic RoE mapper	43
20	7.1.1	Use of RoE control packets	44
21	7.1.2	Structure-agnostic data packet.....	44
22	7.1.2.1	Sub type (subType) field	44
23	7.1.2.2	Flow identifier (flowID) field	44
24	7.1.2.3	Ordering information (orderInfo) field	44
25	7.1.2.4	Length field	45
26	7.1.2.5	Payload field.....	45
27	7.2	Structure-aware RoE mapper.....	46
28	7.2.1	Container definition.....	47
29	7.2.2	Structure-aware CPRI data packet.....	48
30	7.2.2.1	Sub type (subType) field	48
31	7.2.2.2	Flow identifier (flowID) field	48
32	7.2.2.3	Ordering information (orderInfo) field	48
33	7.2.2.4	Length field	48
34	7.2.2.5	Payload field.....	48
35	7.2.3	Control data and the control process	49
36	7.2.4	CPRI C-plane handling	50
37	7.2.5	Synchronization and L1 protocol fields.....	51

1	7.2.6	Handling of Fast C&M	51
2	7.2.7	Handling of Slow C&M	52
3	7.2.8	Handling of other AxC_Ctrl and vendor specific words.....	52
4	7.3	Native RoE time domain packet mapper.....	56
5	7.3.1	Native RoE time domain packet	56
6	7.3.1.1	Sub type (subType) field	56
7	7.3.1.2	Flow identifier (flowID) field	56
8	7.3.1.3	Ordering information (orderInfo) field	56
9	7.3.1.4	Length field	56
10	7.3.1.5	Payload field.....	56
11	7.3.2	Control data	57
12	7.4	Native RoE frequency domain packet mapper	57
13	7.4.1	(i)FFT definition	57
14	7.4.2	Native RoE frequency domain packet.....	58
15	7.4.2.1	Sub type (subType) field	58
16	7.4.2.2	Flow identifier (flowID) field	58
17	7.4.2.3	Ordering information (orderInfo) field	58
18	7.4.2.4	Length field	58
19	7.4.2.5	Payload field.....	58
20	7.4.3	Control data	59
21	7.4.4	PRACH definition.....	59
22	7.4.5	Native RoE PRACH packet	60
23	7.4.5.1	Sub type (subType) field	60
24	7.4.5.2	Flow identifier (flowID) field	60
25	7.4.5.3	Ordering information (orderInfo) field	60
26	7.4.5.4	Length field	60
27	7.4.5.5	Payload field.....	60
28	8	ROE CONTROL PACKET HEADER FORMAT.....	62
29	8.1.1	Sub type (subType) field	62
30	8.1.2	opCode field (opCode)	62
31	8.1.3	OAM TLV control packet	62
32	8.1.3.1	Opcode (opCode) field	62
33	8.1.3.2	Flow identifier (flowID) field	63
34	8.1.3.3	Ordering information (orderInfo) field	63
35	8.1.3.4	Length field	63
36	8.1.3.5	Payload.....	63
37	8.1.4	Ctrl_AxC control packet	64
38	8.1.4.1	Opcode (opCode) field	64
39	8.1.4.2	Flow identifier (flowID) field	64
40	8.1.4.3	Ordering information (orderInfo) field	64
41	8.1.4.4	Length field	64

1	8.1.4.5	Payload.....	65
2	8.1.5	VSD packet control packet.....	65
3	8.1.5.1	Opcode (opCode) field.....	65
4	8.1.5.2	Flow identifier (flowID) field.....	65
5	8.1.5.3	Ordering information (orderInfo) field.....	65
6	8.1.5.4	Length field.....	65
7	8.1.5.5	Payload.....	65
8	8.1.6	Timing control packet.....	66
9	8.1.6.1	Opcode (opCode) field.....	66
10	8.1.6.2	Flow identifier (flowID) field.....	66
11	8.1.6.3	Ordering information (orderInfo) field.....	66
12	8.1.6.4	Length field.....	66
13	8.1.6.5	Payload.....	66

14	BIBLIOGRAPHY (INFORMATIVE)	87
----	---	-----------

15

1 Overview

2 1.1 Scope

3 In a wireless telecommunication network, the link and protocol between the radio elements
4 and baseband controllers is known as fronthaul. This document defines the encapsulation
5 and mapping of radio protocols to be transported over Ethernet packets using Radio over
6 Ethernet (RoE). Furthermore, both structure-agnostic and structure-aware definitions are
7 provided for the most common and current radio protocol – Common Public Radio
8 Interface (CPRI^{TM 1, 2}); as well as a native mode.

9 1.2 Purpose

10 The purpose of this standard is to describe the header formats and packet encapsulations
11 required to transport fronthaul protocols over Ethernet. There are broadly three types of
12 encapsulation schemes performed by RoE mappers;

- 13 a) The Structure-agnostic RoE mapper is a simple transparent encapsulation
14 mapping mechanism. This RoE mapper may have some knowledge of the line
15 coding scheme used but it is oblivious to the underlying structure of the protocol
16 being encapsulated.
- 17 b) The Structure-aware RoE mapper is an encapsulation mechanism that is at least
18 partially aware of the structure of the underlying protocol (such as - CPRI) it
19 encapsulates. Knowledge of the structure enables this RoE mapper to reduce
20 the bandwidth.
- 21 c) RoE native encapsulation. This encapsulation mechanism allows for the
22 transfer time domain, frequency domain and PRACH I/Q data. An intermediate
23 protocol such as CPRI is not required. Structure-aware RoE de-mappers receive
24 native RoE encapsulated I/Q data.

25 De-encapsulation is performed by RoE de-mappers and is the reverse operation of
26 encapsulation.

27 This standard does not specify whether or how the Ethernet packets are guaranteed the
28 strict quality of service (QoS) required by the encapsulated radio protocols. IEEE Std

¹ The CPRI specification is available from the CPRI Website <http://www.cpri.info/>.

² CPRI is a trademark of Nokia.

- 1 802.1TMCM or similar techniques can be used to help ensure such QoS guarantees. It is
2 expected that a full implementation of Radio over Ethernet would comprise;
- 3 a) ingress/egress mapping functions that encapsulate/de-encapsulate using the
4 techniques mentioned earlier,
 - 5 b) a networking technology that minimizes delay and frame delay variation (FDV),
 - 6 c) a time/phase and frequency distribution mechanism and
 - 7 d) de-jittering, buffering and retiming the recovered data.
- 8 This specification is only concerned with point a) above,

2 Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

Common Public Radio Interface (CPRITM) Specification CPRI v7.0.

IEEE Std 754-2008TM, IEEE Standard for binary floating-point arithmetic.

IEEE Std 802.3TM, IEEE Standard for Ethernet.

1 **3 Definitions, acronyms, and abbreviations**

2 **3.1 Acronyms and abbreviations**

3	4G/5G	fourth generation and fifth generation wireless networking technologies
4	AxC	antenna carrier
5	BF	basic frame
6	BFN	base frame number
7	BER	bit error rate
8	C-plane	term used in the CPRI specification for the control plane
9	CPRI	common public radio interface
10	DA	destination address
11	E-UTRA	evolved universal terrestrial radio access (3GPP)
12	FCS	frame check sequence
13	FDD	frequency division duplex
14	FDV	frame delay variation
15	GNSS	global navigation satellite system
16	HFN	hyper frame number
17	IP	internet protocol
18	I/Q	in-phase and quadrature
19	LAN	local area network
20	LC _C	logical connection for control packets
21	LC _D	logical connection for data packets
22	LC _T	logical connection for timing packets
23	LSB	least significant bit
24	LTE	long term evolution
25	MIMO	multiple in multiple out

1	MSB	most significant bit
2	NIC	network interface controller
3	OAM	operation, administration and maintenance
4	PPS	pulse per second
5	PRACH	physical random access channel
6	PTP	precision time protocol
7	QoS	quality of service
8	RE	radio equipment
9	REC	radio equipment controller
10	RF	radio frequency
11	RoE	radio over ethernet
12	SA	source address
13	SoF	start of frame
14	TAI	international atomic time (temps atomique international)
15	TDD	time division duplex
16	TLV	type length value
17	ToD	time of day
18	UTC	universal time coordinated
19	VLAN	virtual LAN
20	VSD	vendor specific data

3.2 Definitions

For the purpose of this document, the following terms and definitions apply. The IEEE Standards Dictionary Online should be consulted for terms not defined in this clause.³

- a) RoE object. An RoE mapper, RoE de-mapper, Ethernet link, CPRI port, FFT or PRACH function.
- b) RoE mapper / RoE de-mapper. An RoE mapper performs the task of forming Ethernet packets from I/Q data sources. A RoE de-mapper performs the task of extracting I/Q data from Ethernet packets.
- c) RoE flow. The term flow is used to describe an individual connection between two RoE nodes which shares the same source and destination address (SA/DA).
- d) RoE field. A field refers to a value within an RoE header.
- e) RoE parameter. A parameter is a value which describes how a radio over Ethernet (RoE) connection operates. Parameters are enumerated to facilitate OAM TLV control.

3.3 Typeface conventions

Within this document, names of parameters are shown in bold typeface and prefixed with a period, for example, **.seqNumPMax**. Fields names are shown in bold typeface without a prefixed period, for example **length**. When parameters are described/assigned with their associated RoE object, the RoE object, its index and the parameter is shown in bold typeface, for example, **mapper[1].destID=4**. Multiple indexes can be described together by using a comma separator, for example, **mapper[2,3].lenPack=8**.

3.4 RoE Ethernet type

All RoE packets shall use the EtherType value shown in Table 3-1.

Table 3-1 - RoE EtherType

Purpose	EtherType
RoE packet	0xFC3D

³ IEEE Standards Dictionary Online subscription is available at http://www.ieee.org/portal/innovate/products/standard/standards_dictionary.html.

3.5 Bit, octet ordering and numerical presentation

For Ethernet packets, this document assumes network byte ordering (i.e., big endian). Figure 3-1 illustrates the bit ordering and numbering within an octet. Similarly, Figure 3-2 illustrates the bit and octet ordering, and corresponding numbering within a 32-bit double word.



Figure 3-1 - Bit ordering and numbering within an octet

For structure-aware and native RoE mappers, I/Q samples are transported without their bits interleaved.

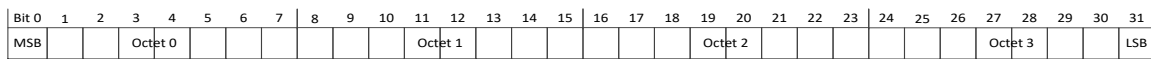


Figure 3-2 - Bit and octet ordering and numbering within a 32-bit double word

The following numerical notations are used in this document:

- Decimal or base 10 value has no specific notation, for example: 763. Every 3 decimal digits are separated by commas, for example: 65,567. Periods are used for decimal points, for example: 3.14159.
- Hexadecimal or base 16 value has a prepended “0x”, for example: 0x02FB. Every 4 hexadecimal digits are separated by a space, for example: 0x1 001F.
- Binary or base 2 value has a trailing “b”, for example: 10 1111 1011b. Every 4 binary digits are separated by a space, for example: 1 0000 0000 0001 1111b.
- ‘NIL’ is the value with all bits set. This translates to -1 for signed values, 0xFF for 8-bit unsigned values, 0xFFFF for 16-bit unsigned values, etc.

4 Radio over Ethernet (RoE)

This clause describes the RoE encapsulation transport format. The following sub-clauses also describe:

- The overall RoE architecture.
- The encapsulation and de-encapsulation functions and locations.
- The RoE mapper functions and locations.
- The underlying assumptions an RoE enabled network has.

4.1 Radio over Ethernet (RoE) architecture

The RoE interface carries 3 types of traffic over physical links. These physical connections carry:

- Logical Connection for Control packets (LC_C). LC_C packets are packets which carry control information. See clause 8.
- Logical Connection for Data packets (LC_D). LC_D packets are packets which carry I/Q data. See sub-clauses 7.1, 7.2 and 7.3.
- Logical Connection for Timing control packets (LC_T). LC_T packets are packets which carry timing control information. See sub-clause 8.1.6. Note that LC_T is a subset of LC_C .

The three logical connections may be carried over the same physical links or over different physical links. This document uses terms node and RoE node, both meaning an RoE capable networking instance that is either the originator or the receiver of an RoE communication link. RoE nodes can be end-points, where RoE traffic is terminated, or pass-through points where the traffic is forwarded to a following node. A given node can support a mixture of terminated and pass-through traffic. There may be zero or more intermediate networking nodes, acting as RoE pass-through points between RoE end-points.

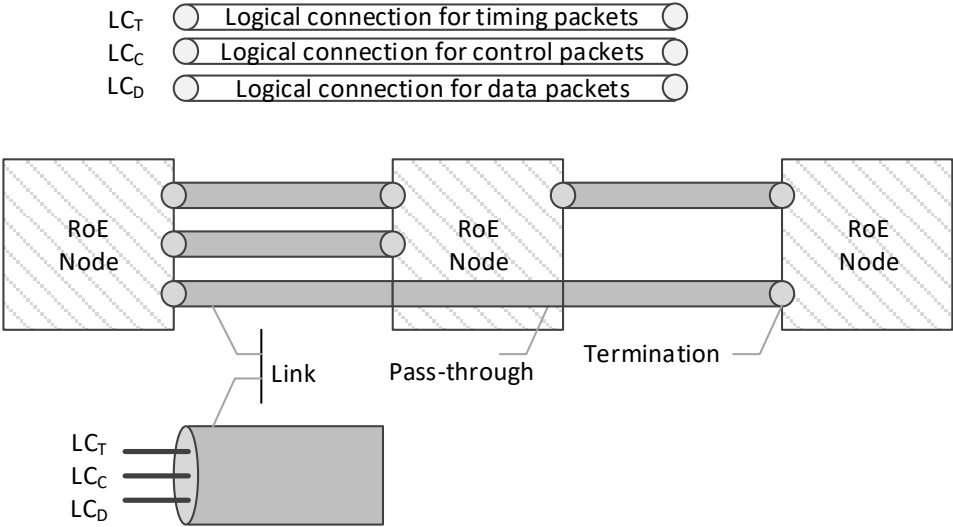


Figure 4-1 - RoE end-points and supported functions

The RoE standard supports scenarios where RoE enabled nodes are connected to legacy CPRI nodes and/or end-points. The RoE encoder/decoder (Enc/Dec) provides this capability using RoE mappers and RoE de-mappers. This allows either the radio equipment controller (REC) or radio equipment (RE) to be either RoE enabled, CPRI enabled or any combination of both.

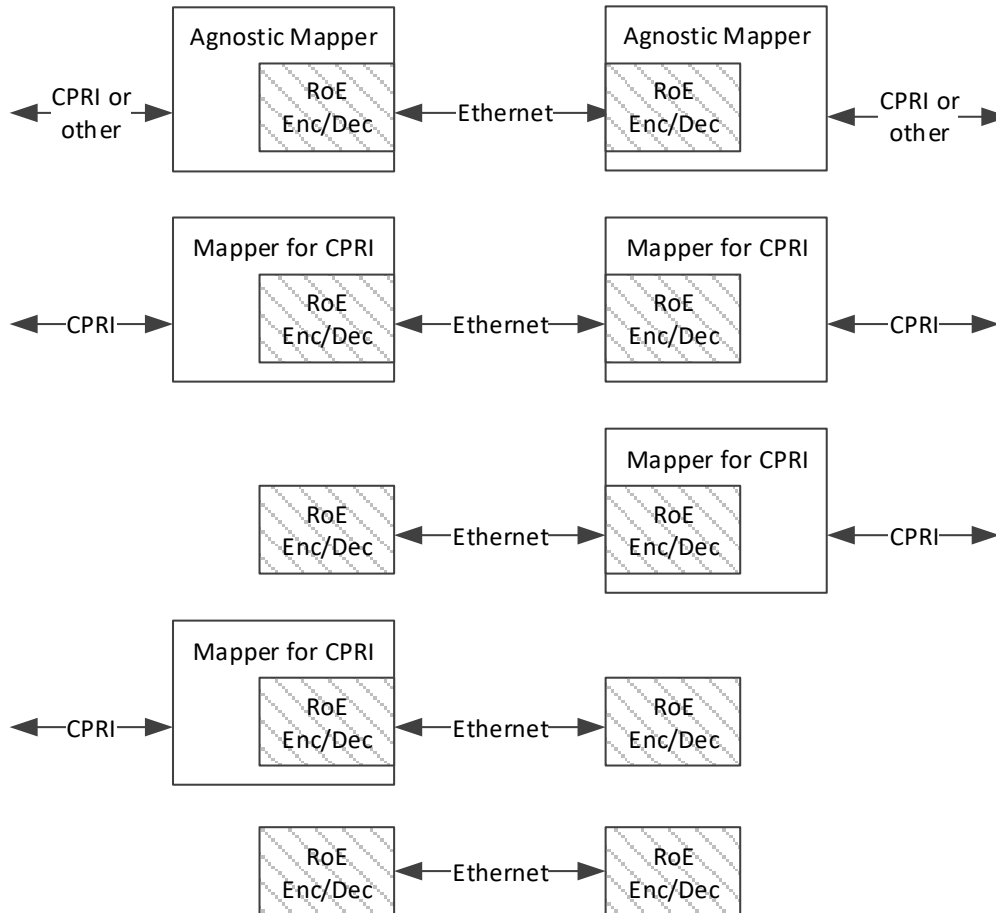
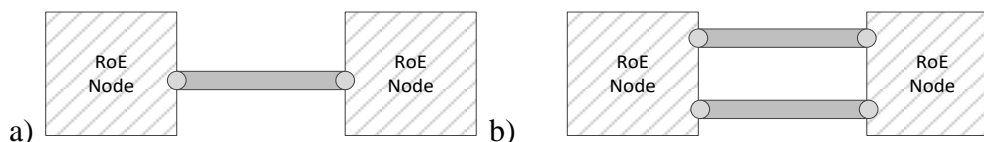


Figure 4-2 - RoE nodes and supported functions

4.1.1 Topologies

RoE supports several different topologies; Figure 4-3 shows the supported topologies. These are:

- a) Single point-to-point
- b) Multiple point-to-point
- c) Point to multi-point
- d) Chain
- e) Ring
- f) Tree



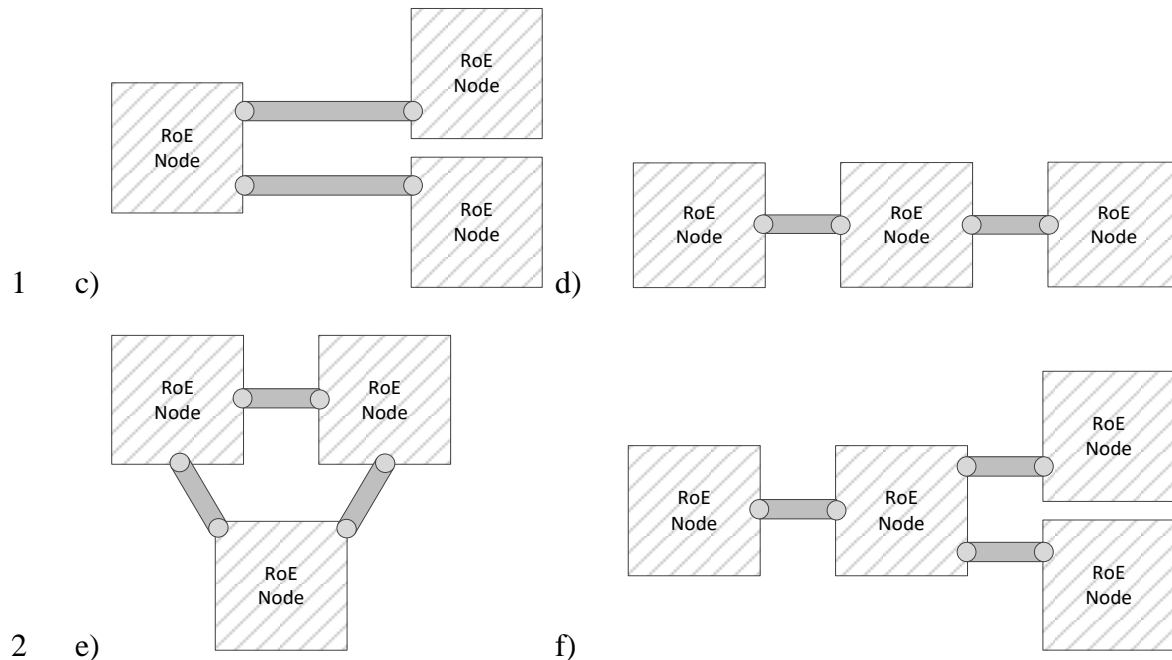


Figure 4-3 - RoE topologies supported

4.1.2 Underlying network (Informative)

For RoE to work successfully, the underlying network needs to be provisioned appropriately. While the need for a network which is time sensitive is clear, this RoE specification does not specify how this is to be achieved. Several other standards, including ITU-T G.8275 and IEEE Std 802.1CMTM address this. In general, the underlying network will need to fulfill the following requirements:

- A network comprised of bridges and point-to-point Ethernet links.
- The number of actual links and nodes are not in scope, however, the delay and the FDV should meet the application's requirements.
- The network will need management for delay and FDV.
- Highly managed network - not oversubscribed (at least for traffic at the same/higher priority as RoE), known peak bandwidth, latencies and FDV known in advance. Network is required to have sufficient bandwidth to carry RoE traffic.
- Support for time of day (ToD) distribution.
- Both the transmitting and receiving end-points need to share the same understanding of the ToD in order for the information to be presented at the desired time.
- No retransmission (BER is no more than 10^{-12}) for RoE traffic.

- 1 h) The maximum one-way transmission time needs to satisfy the application's
- 2 requirements and needs to be less than the time quantity represented by a
- 3 presentation time or sequence number in the RoE orderInfo field.
- 4 i) Ethernet network that maintains the frame source and destination addresses.

5 **4.1.3 Synchronizing nodes (Informative)**

6 The actual method and characteristics of synchronizing RoE nodes is out of scope, but
 7 methods such as Global Navigation Satellite System (GNSS) and IEEE Std 1588v2TM, or
 8 later may be used to synchronize nodes.

9 **4.2 RoE objects**

10 As shown in Figure 4-4, an RoE node has the following objects;

- 11 a) Ethernet Link(s). Each with unique SA/DA pairs and at least one is mandatory for
- 12 all RoE implementations.
- 13 b) RoE mapper / RoE de-mapper. Required for structure-aware mode and typically
- 14 used in structure-agnostic mode. An RoE mapper packetizes data while a RoE de-
- 15 mapper de-packetizes data. Mandatory for all RoE implementations. Figure 4-5
- 16 shows objects related to RoE mappers / RoE de-mappers.
 - 17 1) RoE mapper / RoE de-mapper container. Used in structure-aware mode to
 - 18 describe how to construct/deconstruct a CPRI packet.
 - 19 2) RoE mapper / RoE de-mapper FFT. Used in RoE Native frequency domain
 - 20 mode to describe how to construct/deconstruct an RoE Native frequency
 - 21 domain I/Q data packet.
 - 22 3) RoE mapper / RoE de-mapper PRACH. Used in RoE Native frequency
 - 23 domain mode to describe how to construct/deconstruct an RoE Native
 - 24 frequency domain PRACH packet.
- 25 c) CPRI Port. Required for structure-aware and structure-agnostic RoE mappers.

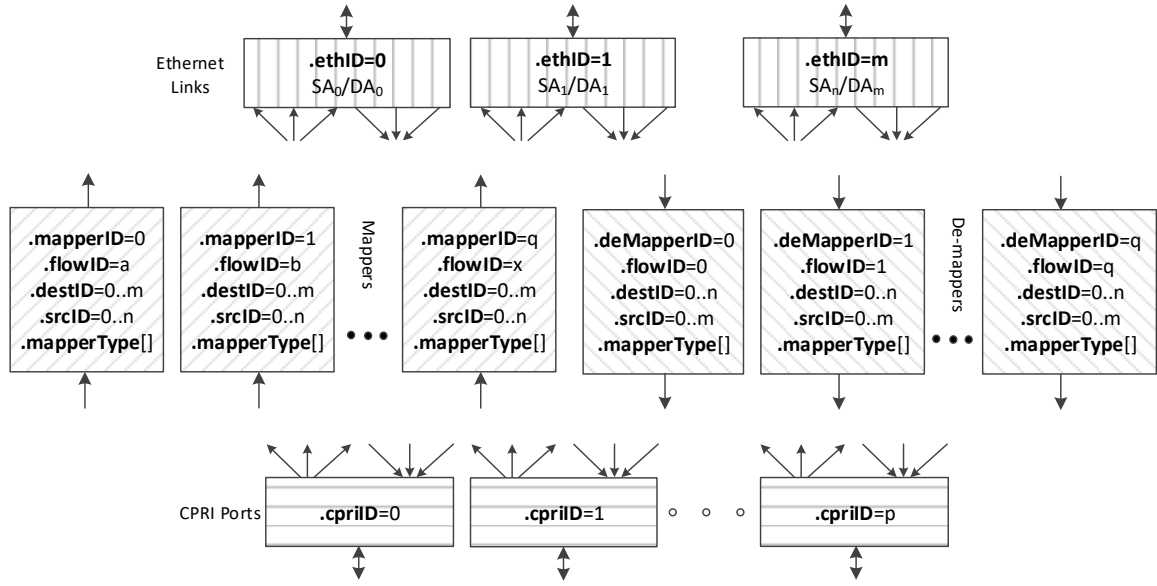


Figure 4-4 – RoE objects and hierarchy

4.2.1 Ethernet link

In this document, an Ethernet link is defined as a connection with a particular source and destination address. An RoE node shall have at least one Ethernet link. An Ethernet link can accept and provide packets of data from multiple flows and RoE mappers. Ethernet links are assumed to be bi-directional; however, it is also allowed for just one direction to be used.

4.2.2 RoE mappers / RoE de-mappers

An RoE mapper is a function/process which converts other transport framing formats to an RoE framing format, and a RoE de-mapper performs the opposite function. This specification describes different flavors of RoE mappers. Structure-aware and native RoE mappers separate control data and sample data into different channel. Control data is fed to a block known as the “control process” in this specification (see sub-clause 7.2.3). On the RoE de-mapper direction, the RoE control packets received by the control process are fed to RoE de-mappers with the **.flowID** set to NIL **flowID** and the **.deMapperID** set to appropriate values by the control process. The chief function of an RoE mapper is to select the appropriate input data from the input data stream (e.g. a CPRI port) and then packetize it. The RoE mapper also generates or selects control information. Individual sub-clauses in this specification describe the operation of RoE mappers and RoE de-mappers.

The RoE mapper is responsible for generating the ordering information. The functionality of the ordering information is described in sub-clause 4.4.4.

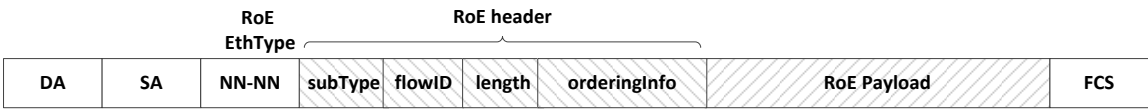
Structure-agnostic RoE mappers do not have control packets as the entire data stream (including control information) is encapsulated as a single LC_D data channel.

1 **4.2.3 CPRI ports**

2 CPRI ports are physical connections. In structure-aware and structure-agnostic mapping
3 modes, an RoE node has CPRI (or CPRI-like) ports. CPRI (or CPRI-like) ports are assumed
4 to be bi-directional, however, it is also allowed for just one direction to be used.

5 **4.3 RoE traffic types**

6 RoE frames are encapsulated in Ethernet packets as shown in Figure 4-5. For the purposes
7 of this document, the SA, DA and FCS are implicit to all RoE packets. The RoE EtherType
8 value is specified in sub-clause 3.4. This document focuses on the fields contained in the
9 RoE header and RoE payload fields.

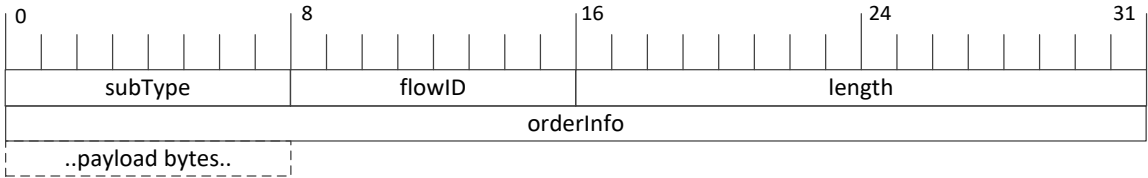


11
12 **Figure 4-5 - RoE encapsulation in Ethernet packets**

13 **4.4 RoE common frame format**

14 This sub-clause documents the octets of the frame (i.e., the RoE header) that are common
15 to all RoE flow data and control packets. Figure 4-6 illustrates the frame format and its
16 fields. The common RoE frame format has the following header fields:

- 17 a) **subType** (sub type) field: 8 bits
- 18 b) **flowID** (flow identifier) field: 8 bits
- 19 c) **length** (length) field: 16 bits
- 20 d) **orderInfo** (timeStamp/seqNum) field: 32 bits
- 21



23 **Figure 4-6 - RoE encapsulation common frame format – the RoE header**

24 The RoE common frame header is placed into the transport protocol payload field, which
25 in this document context is the Ethernet frame payload field.

26 **4.4.1 Sub type (subType) field**

27 The 8-bit **subType** field is used to define the RoE sub type and the type of flow carried by
28 the RoE packets. This document reserves sub types listed in Table 4-1.

1

Table 4-1 – RoE subType values

Binary value	Function	Description
0000 0000b	RoE Control sub type	Control packet between the RoE control node and RoE controlled node.
0000 0001b	Reserved	Reserved.
0000 0010b	RoE Structure-agnostic data sub type	Data payload packet with RoE common frame header and structure-agnostic payload.
0000 0011b	RoE Structure-aware CPRI data sub type	Data payload packet with RoE common frame header and structure-aware CPRI data payload.
0000 0100b	RoE Slow C&M CPRI sub type	C&M payload packet with common RoE frame header and structure-aware CPRI Slow C&M payload.
0000 0101b – 0000 1111b	Reserved	Reserved for future sub types.
0001 0000b	RoE Native time domain data sub type	Time domain data payload packet with RoE common frame header.
0001 0001b	RoE Native frequency domain data sub type	Frequency domain data payload packet with RoE common frame header.
0001 0010b	RoE Native PRACH data sub type	PRACH IQ data payload with common frame header
0001 0011b – 1111 1011b	Reserved	Reserved for future sub types.
1111 1100b – 1111 1111b	Experimental	Reserved for experimental types. The nature and purpose of an experimental subType cannot be known in advance, thus the structure after the common RoE header is defined as opaque. Entities implementing the subType can interpret the message according to their implementation.

2 4.4.2 Flow identifier (flowID) field

3 The **flowID** identifies a specific flow between two end-points. The end-points are defined
 4 as Ethernet packet Source Address (SA) and Destination Address (DA) pair in the context
 5 of this specification. The **flowID** allows multiplexing up to 255 specific flows between two
 6 end-points.

The **flowID** identifier has no routing function and is solely interpreted by the end-points. The **flowID** identified flow may consist of multiple sub-flows (i.e., a group flow). The interpretation of flow content and possible sub-flows is solely controlled by the end-points.

RoE mapper **flowIDs** are allocated numerically, the RoE de-mapper **flowIDs** are assigned through parameters. Both ends of a given flow share the same **flowID**.

This document reserves flow identifier values listed in Table 4-2.

Table 4-2 – RoE flowID values

Binary value	Function	Description
0000 0000b – 1111 1110b	flowID number	Flow identifiers available for use to identify specific flows between two nodes.
1111 1111b	NIL flowID	Reserved flowID indicating that the field shall not be interpreted as any specific flow. The NIL flowID is used for RoE control packets.

4.4.3 Length (length) field

Although an Ethernet Network Interface Controller (NIC) can determine the length of packets through low-level encoding features, these length fields are not implicitly available to higher layers or when packets are sent through other transports. RoE packets shall follow a scheme similar to Internet Protocol (IP) version 6 and include a length field where the value of the length is the total number of octets following the common RoE header. The length does not include the Ethernet FCS bytes.

4.4.4 Ordering information (orderInfo)

Ordering information is assigned to each flow and is presented in one of two methods, a sequence number or time stamp. The **orderInfo** type used for a given flow is determined by a parameter. This parameter is described in Table 6-5 and Table 6-6. This parameter shall be maintained throughout the lifetime of the flow. For data packets, the **orderInfo** value is the sequence number or timestamp which applies at the start of the RoE payload data.

4.4.4.1 Sequence number (seqNum) field

The sequence number field is used to identify the order of successive packets. In order to allow additional flexibility, it is possible to have two independent counters, e.g. one for Hyper Frame Number (HFN) and another for Base Frame Number (BFN).

The sequence number field is generated by the RoE mapper and consists of three independently sized fields, whose sizes depend on two variables p and q. The bits p up to and including 31 constitutes the **p-counter** field and the bits q to p-1 constitutes the **q-counter** field.

Figure 4-7 illustrates the composition of different fields that together form the sequence number field. The following rules shall apply:

a) $0 \leq p < 32$ and

b) $0 \leq q \leq p$ and (if $q=p$ then the **q-counter** is not present).

If **.seqNum** is in use, the **p-counter** field shall exist. The other two fields may exist based on the p and q values. Each RoE flow that uses sequence numbers shall have its own instance of **p-counter** and **q-counter**. Each of the fields is initialized to a known value on the first use of the sequence number. Note that the **.seqNumPMax**, **.seqNumPIncProp**, **.seqNumQMax**, **.seqNumQInc**, **.seqNumQIncProp**, and **.seqNumRsvd** values shall remain unchanged during the lifetime of the RoE flow but each flow may have different values for these parameters.



Figure 4-7 – Sequence number composition

The following parameters listed in Table 4-3 are used to describe the initial values, behavior and number space wrap properties of the sequence number.

Table 4-3 - Sequence number related parameters

enumParam	Parameter	Bits	Default value	Description
0	.seqNumPMax	32	0xFFFF FFFF	Maximum numerical value of the p-counter field so that .seqNumPMax is less than 2^{32-p} .
1	.seqNumPVal	32	0x0000 0000	The counter value of the p-counter field. When the .seqNumPVal > .seqNumPMax then the counter field wraps to a value .seqNumPVal modulo (.seqNumPMax +1). The counter wrap also causes the .seqNumQVal to be incremented by .seqNumQInc based on the .seqNumQIncProp setting.
2	.seqNumPIncProp	3	001b	0x0: No increment. 0x1: Increments by a fixed value indicated by .seqNumPInc on every packet sent.

				<p>0x2: Increment by the RoE payload size on every sent packet.</p> <p>0x3 to 0x6: Reserved.</p> <p>0x7: Experimental</p>
3	.seqNumPInc	32	0x0000 0001	<p>The increment value of .seqNumPVal on every sent packet.</p> <p>Designated value when .seqNumPIncProp equals 0x1 which is fixed during the lifetime of an RoE flow, ignored when .seqNumPIncProp equals any other value.</p>
4	.seqNumQMax	32	0x0000 0000	<p>Maximum numerical value of the q-counter field so that .seqNumQMax is less than 2^{P-q}.</p>
5	.seqNumQVal	32	0x0000 0000	<p>The counter value of the q-counter field. When the .seqNumQVal > .seqNumQMax then the counter field wraps to a value .seqNumQVal modulo (.seqNumQMax+1).</p>
6	.seqNumQIncProp	3	001b	<p>0x0: No increment.</p> <p>0x1: Increment by a fixed value indicated by .seqNumQInc on every .seqNumPVal wrap event.</p> <p>0x2 to 0x6: Reserved.</p> <p>0x7: Experimental</p>
7	.seqNumQInc	32	0x0000 0001	<p>The increment value of .seqNumQVal on every .seqNumPVal wrap event.</p> <p>Designated value when .seqNumQIncProp equals 0x1, ignored when .seqNumQIncProp equals other values.</p>
8	.seqNumRsvd	32	0x0000 0000	<p>Static value for reserved field, if available. Shall remain unchanged for the lifetime of the flow.</p>

9-255	Reserved			reserved
-------	----------	--	--	----------

4.4.4.2 Timestamp (timeStamp) field

The timestamp field is generated by the RoE mapper. It is 32-bits in size and provides the start-of-frame marker, a condensed sequence number, and the absolute time for presentation of the packet information by the de-mapper at the receiving end-point.

The presentation time is relative to a reference plane, as shown in Figure 5-1 for the mapper and de-mapper, which is itself referenced to a timescale (see sub-clause 5.1). Both the transmitting and receiving end-points must be referenced to the same timescale for the information to be presented at the appropriate time. Which timescale to use is out-of-scope of this standard.

The ToD used for the presentation time mechanism shall be tracked with a 24-bit nanosecond counter and a 5-bit fractional nanosecond counter. The presentation time counters at the associated endpoints shall be aligned such that their 29-bit ToD values are consistent (to a specified uncertainty) with the ToD of a counter that started at zero at the chosen common timescale's epoch and that increments with that timescale's definition of the duration of 1/32 nanoseconds. An example of how this alignment is achieved for a Precision Time Protocol (PTP) timescale is shown in Annex C.

The format of the **timeStamp** field is shown in Figure 4-8.

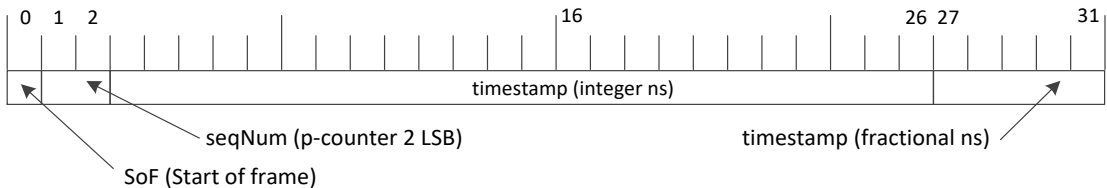


Figure 4-8 - Format of the timeStamp field

Bit 0 is the Start of frame (**SoF**) marker and is an indication of a radio frame boundary. When the **SoF** bit is set to 1, this indicates the start of the payload contained within the packet is the start of the radio frame. Bits 1 and 2 contain the 2 least significant bits of the **p-counter** from the **.seqNum** information.

Bits 3 through to 26 of the **timeStamp** field is the integer nanosecond portion of the presentation timestamp. It counts in units of nanoseconds and the value ranges from 0 ns to 16,777,215 ns (0x0 to 0xFF FFFF respectively).

Bits 27 through to 31 of the **timeStamp** field is the fractional nanosecond portion of the presentation timestamp. It counts in fractions of a nanosecond (0.03125 ns) where 0x0 represents 0 ns and 0x1F represents 0.96875 ns. If sub-nanosecond timestamping is not used, these five bits shall be set to 0 at the sender and shall be ignored at the receiver.

1 The size of the presentation timestamp permits the mapper to specify a presentation time
2 that is up to 16,777,215.96875 ns in the future.

3 Under non-erroneous conditions, the presentation timestamp value shall correspond to the
4 same absolute ToD value at the mapper and the de-mapper to the nanosecond level or better
5 (the use of fractional nanoseconds bits is optional). Examples of the how the presentation
6 timestamp value is generated at the mapper and interpreted at the de-mapper for both non-
7 erroneous and erroneous conditions are given in Annex C.

8 **4.4.5 Payload field**

9 The content, structure and size of the payload field is specific to an RoE sub type and its
10 definition. The payload may contain a flow of In-phase and Quadrature (I/Q) samples for
11 a single antenna carrier or a group of antenna carriers. Both single and group content is
12 identified by a **flowID** between two RoE nodes. Furthermore, when specific RoE mappers
13 are applied the payload field can contain, for example, an individual antenna carrier
14 component flow of a decomposed CPRI basic frame. In a case of RoE control packets, the
15 payload may contain appropriate control and management information, such as in the form
16 of Type Length Value (TLV) messages.

17 The total RoE payload field size shall always be full octets.

18 RoE payloads shall have a minimum size of 64 bytes.

5 Timing and synchronization considerations

In general, for an RoE interface to work, one or both ends of the link need some knowledge of time. In particular, in order to comply with radio standards, the presentation time of the radio signal needs to be tightly controlled. The mechanism for achieving this, however, is outside the scope of this standard.

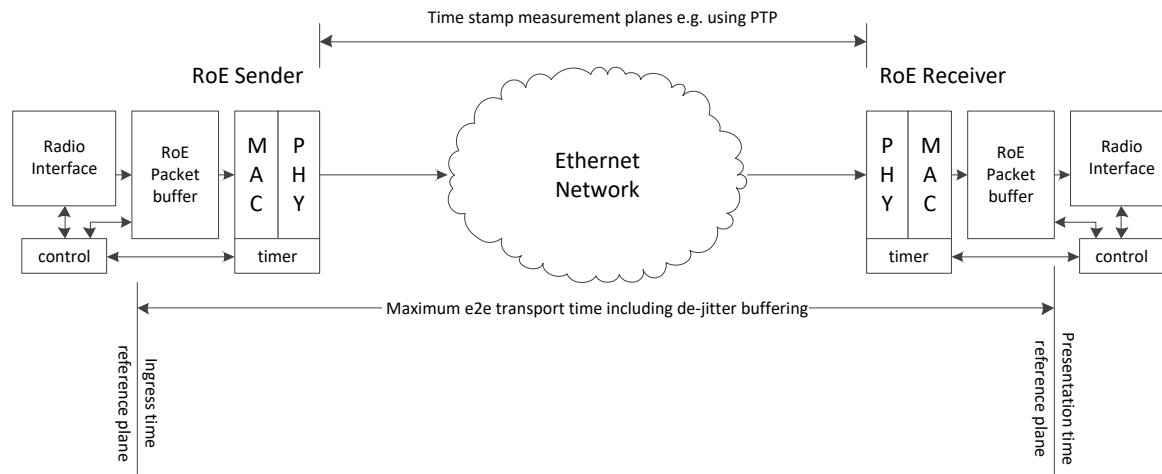


Figure 5-1 - Presentation time measurement points

5.1 General assumptions

The RoE epoch is 1 January 1970 00:00:00 International Atomic Time (TAI), which is 31 December 1969 23:59:51.999918 UTC. It is assumed (but not mandated) that both RoE end-points have an access to a reference time source. The RoE time source, when available, shall be synchronized to TAI.

RoE end-point devices should provide a one Pulse Per Second (1PPS) output as described in clause A.1 or clause A.2 of ITU-T G.8271/Y.1366.

5.2 RoE presentation time

The RoE presentation time is used to phase-align the transfer of data between the RoE end-points. The presentation time is calculated by the RoE sender and represents the time when the RoE packet payload shall be played out from the RoE receiver packet buffer to the consumer of the payload data.

5.3 Presentation time measurement points

Figure 5-1 illustrates the measurements planes for the RoE presentation time. When an RoE sender calculates the presentation time at the RoE receiver, it has to take the entire end to end delay between the RoE sender and receiver reference planes into account. The end to end delay consists of the networking delay (i.e., the transit time), processing and

- 1 enough to compensate for FDV introduced by the network and internal processing at both
- 2 end-points.
- 3 The method for measuring the end to end delay is implementation-specific and
- 4 deployment-specific.

6 RoE parameters

RoE parameters are communicated during link establishment and define how the RoE node shall operate. These parameters set and control the internal configuration of the node and the RoE mapper/ RoE de-mapper functionality.

6.1 Parameter introduction

6.1.1 Parameter defaults

In order to minimize the link establishment traffic, default parameter values are assumed unless specifically set.

6.1.2 Parameter set nomenclature

RoE defines some distinct parameters sets for different object types; these are defined as follows;

- a) Ethernet link – **eth[ethID].parameter name = parameter value.**
- b) CPRI port – **cpri[cpriID].parameter name = parameter value.**
- c) Mapper – **mapper[mapperID].parameter name = parameter value.**
 - 1) Mapper container - **mapper[mapperID].[containerID]. parameter name = parameter value.**
 - 2) Mapper FFT – **mapper[mapperID].[fftID].parameter name = parameter value.**
 - 3) Mapper FFT PRACH - **mapper[mapperID].[fftID].[PRACH].parameter name = parameter value.**
- d) De-mapper – **demapper[demapperID].parameter name = parameter value.**
 - 1) De-mapper container - **demapper[demapperID].[containerID].parameter name = parameter value.**
 - 2) De-mapper FFT – **demapper[demapperID].[fftID].parameter name = parameter value.**
- e) Sequence number parameters – these parameters belong to RoE mapper and RoE de-mapper objects.
 - 1) – **mapper[mapperID].parameter name = parameter value.**
 - 2) – **demapper[demapperID].parameter name = parameter value.**

6.1.3 Parameter exchange and enumeration

Parameters are either programmed/communicated out of band using some management protocol or exchanged using Operations Administration and Management (OAM) TLV packets, see sub-clause 8.1.3. To form an OAM TLV packet, object types and their parameters are enumerated. Object types are enumerated in Table 6-1 by the **enTLV** field. Parameters are enumerated in the parameter definition tables in the sub-clauses in sub-clause 6.2 by the **enParam** field.

Table 6-1 – Object type enumeration

enTLV	Object type	Description
0	Ethernet link	Ethernet link object type.
1	CPRI port	CPRI port object type.
2	Mapper	RoE mapper object type.
3	De-mapper	RoE de-mapper object type.
4	Mapper container	RoE mapper container object type, belonging to an RoE mapper.
5	De-mapper container	RoE de-mapper container object type, belonging to an RoE de-mapper.
6	Mapper FFT	RoE mapper FFT object type, belonging to an RoE mapper.
7	De-mapper FFT	RoE de-mapper FFT object type, belonging to an RoE de-mapper.
8	Mapper PRACH	RoE mapper PRACH object type, belonging to an RoE mapper FFT.
9-63	Reserved	Reserved
64	RoE 1914.1 TLV	Service OAM. This object type allows TLVs described in IEEE Std 1914.1 TM to be uniquely enumerated for parameter exchange.
65-127	Reserved	Reserved

6.2 Parameter lists

6.2.1 Encryption

RoE allows for the encryption and decryption of links and flows. The underlying encryption technique and mechanism by which this is performed is outside of the scope of this standard. The default is no encryption.

6.2.2 Compression

RoE allows for the compression and decompression of links and flows. The underlying compression technique and mechanism by which this is performed is outside of the scope of this standard. The default is no compression.

6.2.3 Ethernet link parameters

A given RoE node can support up to 65,536 Ethernet links and each link is assigned a unique identifier. These link parameters are set during link establishment.

Table 6-2 – Ethernet link parameters

enParam - Parameter	Bits	Name	Default	Description
0 - Identifier	16	.ethID	0	Each Ethernet port in a given node has a unique identifier.
1-255				Reserved

6.2.4 CPRI port parameters

A given RoE node can support up to 65,536 physical CPRI ports and each port is assigned a unique identifier. Each port can be assigned one of 15 encryption and compression schemes. These port parameters are set during link establishment.

Table 6-3 – CPRI port parameters

enParam - Parameter	Bits	Name	Default	Description
0 - Identifier	16	.cpriID	0	Each CPRI port in a given node has a unique identifier.
1 – Port speed	64	.cpriSpeed	0	The speed of the CPRI port in Mbps, expressed as a IEEE Std 754 TM double precision, binary 64 floating point number.
2 – CPRI version	8	.cpriVer	0	Protocol version at location Z.2.0 i.e., control word 2.
3 – HDLC rate	3	.cpriHDLC	0	HDLC bit rate at location Z.66.0 i.e., control word 66.
4 – L1 location	5	.cpriL1	0	L1 signaling at location Z.130.0 i.e., control word 130.
5 – Ethernet pointer	6	.cpriEth	0	Ethernet pointer at location Z.194.0 i.e., control word 194.
6-255				Reserved

6.2.5 RoE mapper / RoE de-mapper parameters

RoE mapper and RoE de-mapper parameters share the same **.enParam** enumeration space, however, not all parameters are relevant to all RoE mappers and RoE de-mappers. describes this relevance.

Table 6-4 – RoE mapper/de-mapper parameter relevance

	Mapper / de-mapper type			
enParam - Parameter	Structure-agnostic	Structure-aware	Native RoE time domain	Native RoE frequency domain
0 - Identifier	Y			
1 - Flow	Y			
2 - Source link	Y			
3 - Destination Ethernet link	Y			
4 - orderInfo type	Y			
5 – Mapper Type	Y			
6 - Encryption	Y			
7 - Compression	Y			
8 - Packet length	Y			
9 – Accepted Time Window	Y			
10 – Sample Width	N	Y		
11-31	Reserved			
32 - Number of containers	N	Y	N	
33 - Number of control containers	N	Y	N	
34 – Number of FFT objects	N			Y
35 – Number of PRACH objects	N			Y
36-255	Reserved			

6.2.6 RoE mapper parameters

This sub-clause describes the generic parameters required for an RoE mapper. Different types of RoE mappers require different parameters; these are described in detail in following sub-clauses. A given RoE node can support up to 255 RoE mappers and each RoE mapper is assigned a unique identifier. For packets being sent by a mapper, the RoE packet common header **flowID** field is populated with the **mapper[.flowID]** parameter value corresponding to the receiver's de-mapper. As such, this parameter may be non-unique to the mapper and different mappers can have overlapping **flowIDs**.

Each RoE mapper has an associated destination Ethernet link and source Ethernet link/CPRI port. This parameter defines the wiring of the RoE mapper.

In the case of the structure-aware RoE mapper, the number of containers shall be defined.

An RoE mapper includes a functional block which packetizes the data. The length of the packet is controlled by a packet length parameter.

Each RoE de-mapper can be assigned either no encryption and compression or one of 15 encryption and compression schemes.

These RoE mapper parameters are set during link establishment.

Table 6-5 – RoE mapper parameters

enParam - Parameter	Bits	Name	Default	Description
0 -Identifier	8	.mapperID	0	Each RoE mapper in a given node has a unique identifier.
1 - Flow	8	.flowID	0	This is assigned by the master controller to map to the receiver's deMapperID . RoE mapper flowIDs are not required to be unique.
2 - Source link	16	.srcID	0	Identifies the source Ethernet link/CPRI port.
3 - Destination Ethernet link	16	.destID	0	Identifies the destination Ethernet link.
4 - orderInfo type	1	.orderInfoType	0	0 indicates seqNum is used. 1 indicates timeStamp is used.
5 – RoE mapper Type	4	.mapperType	0	0 indicates structure-agnostic simple tunneling mode. 1 indicates structure-agnostic mode & remove line encoding. 2 indicates structure-aware mode. 3 indicates native time domain mode. 4 indicates native frequency domain mode. 5-15 reserved.

6 - Encryption	4	.encrypt	0	Selects/enables encryption on a given flow. 0x0 is no encryption.	
7 - Compression	4	.compress	0	Selects/enables compression on a given flow. 0x0 is no compression.	
8 - Packet length	16	.lenPack	0	Identifies the amount of data to include per packet payload.	
				.mapperType	.lenPack units
				0,1,3 & 4	Octets.
				2	Container sets.
9 – Accepted Time Window	29	.acceptTimeWindow	0	<p>A 29-bit value that is used with the orderInfo timeStamp and the arrival time of the timestamped radio data at de-mapper to determine if an RoE packet experienced excess transit delay.</p> <p>A received RoE packet with (an orderInfo timeStamp minus currentTime which is either negative or greater than the .acceptTimeWindow constitutes an error condition. The currentTime is the RoE node’s current ToD truncated to a similar 29-bit value as the timeStamp. The use of this field is for error checking only. See Annex C.</p>	
10 – Sample Width	8	.sampleWidth	16	Indicates the number of bits in each I portion and in each Q portion of an I/Q sample. By default, 16-bit I and 16-bit Q width is assumed.	
11-31				Reserved	
32 - Number of containers	8	.numContainers	0	Defines the number of containers associated with this RoE mapper.	

33 - Number of control containers	8	.numCtrlConts	0	Defines the number of control containers associated with this RoE mapper.
34 – Number of FFT objects	8	.numFFT		Defines the number of FFT objects.
35 – Number of PRACH objects	8	.numPRACH		Defines the number of PRACH objects.
36-255				Reserved

6.2.7 RoE de-mapper parameters

This sub-clause describes the generic parameters required for a RoE de-mapper. Different types of RoE de-mappers require different parameters; these are described in detail in following sub-clauses. A given RoE node can support up to 255 RoE de-mappers. Every de-mapper has a unique **.deMapperID**. For packets being received by the de-mapper, the sender mapper will populate the RoE common header **flowID** with the target de-mapper's **.deMapperID** value.

Each RoE de-mapper has an associated source Ethernet link and destination Ethernet link/CPRI port. This parameter defines the wiring of the RoE de-mapper.

In the case of the structure-aware RoE de-mapper, the number of containers shall be defined.

A RoE de-mapper includes a functional block which de-packetizes the data. The length of the packet is controlled by a packet length parameter.

Each RoE de-mapper can be assigned either no encryption and compression or one of 15 encryption and compression schemes.

These RoE de-mapper parameters are set during link establishment.

Table 6-6 – RoE de-mapper parameters

enParam - Parameter	Bits	Name	Default	Description
0 - Identifier	8	.deMapperID	0	Each RoE de-mapper in a given node has a unique identifier.
1 - Flow	8	.flowID	0	For an RoE de-mapper, the flowID = .deMapperID .
2 - Source Ethernet link	16	.srcID	0	Identifies the source Ethernet link.

3 - Destination Ethernet link/ CPRI port	16	.destID	0	Identifies the destination Ethernet link/CPRI port.						
4 - orderInfo type	1	.orderInfoType	0	0 indicates seqNum is used. 1 indicates timeStamp is used.						
5 – RoE mapper Type	4	.mapperType	0	0 indicates structure-agnostic simple tunneling mode. 1 indicates structure-agnostic mode & remove line encoding. 2 indicates structure-aware mode. 3 indicates native time domain mode. 4 indicates native frequency domain mode. 4-15 reserved.						
6 – Encryption	4	.encrypt	0	Selects/enables encryption on a given flow. 0x0 is no encryption.						
7 – Compression	4	.compress	0	Selects/enables compression on a given flow. 0x0 is no compression.						
8 – Packet length	16	.lenPack	0	Identifies the amount of data to include per packet payload. <table><tr><td>.mapperType</td><td>.lenPack units</td></tr><tr><td>0,1,3 & 4</td><td>Octets.</td></tr><tr><td>2</td><td>Container sets.</td></tr></table>	.mapperType	.lenPack units	0,1,3 & 4	Octets.	2	Container sets.
.mapperType	.lenPack units									
0,1,3 & 4	Octets.									
2	Container sets.									
9 – Accepted Time Window	29	.acceptTimeWindow	0	A 29-bit value that is used with the orderInfo timeStamp and the arrival time of the timestamped						

				<p>radio data at de-mapper to determine if an RoE packet experienced excess transit delay.</p> <p>A received RoE packet with (an orderInfo) timeStamp minus currentTime which is either negative or greater than the .acceptTimeWindow constitutes an error condition. The currentTime is the RoE node's current ToD truncated to a similar 29-bit value as the timeStamp. See Annex C.</p>
10 – Sample Width	8	.sampleWidth	16	Indicates the number of bits in each I portion and in each Q portion of an I/Q sample. By default, 16-bit I and 16-bit Q width is assumed.
11-31				Reserved
6 - Number of containers	8	.numContainers	0	Defines the number of containers associated with this RoE de-mapper.
12-255				Reserved
33 - Number of control containers	8	.numCtrlConts	0	Defines the number of control containers associated with this RoE de-mapper.
34 – Number of FFT objects	8	.numFFT		Defines the number of FFT objects.
35-255				Reserved

1 7 RoE mappers

2 This clause defines RoE mappers to/from existing radio framing formats to/from RoE
 3 native transport encapsulation format. The different mapping techniques affect the overall
 4 efficiency of the link. If the encoding and packing method of the source of the I/Q stream
 5 is not known, then the parameters required for using a more efficient RoE mapper are not
 6 available. In this case, using the simple tunneling RoE mapper (**.mapperType=0**) is the
 7 only possible choice. When the relevant information on the encoding and packing is known,
 8 a structure-aware RoE mapper (**.mapperType=1**) can be used.

9 7.1 Structure-agnostic RoE mapper

10 The structure-agnostic RoE mapper captures bits from one end of a constant bit rate link,
 11 packetizes the bits into Ethernet packets, sends the packets across the network, and then
 12 recreates the bit stream at the far end of the link. While the constant bit rate data stream is
 13 commonly encoded with the CPRI protocol, it could also be of any other protocol, provided
 14 it is within the range of data rates supported by that equipment.

15 The agnostic RoE mapper has two main modes of operation (defined by the **.mapperType**
 16 parameter).

17 a) Tunnelling mode (**.mapperType=0**), works as a simple Ethernet tunnel. It does not
 18 remove any line coding bits and does not interpret any special characters (such as
 19 K-characters). If the source data is 8b/10b-encoded, the 10-bit symbols present on
 20 the line will be tunnelled by this RoE mapper as 10 bits of data. Similarly, 66-bit
 21 symbols will be sent for 64b/66b-encoded data as 66 bits of data. The entire stream
 22 is simply packetized.

23 b) Line coding aware mode (**.mapperType=1**), removes the line coding bits such as
 24 for CPRI encoded with 8b/10b or 64b/66b. If the source data is 8b/10b-encoded,
 25 the 8-bit symbols present on the line will be tunnelled by this RoE mapper as 8 bits
 26 of data. This mode removes any 8b/10b encoding and extracts/stores **.lenPack**
 27 octets from/to the source stream. Similarly, if the source data is 64b/66b-encoded,
 28 the 64-bit symbols present on the line will be tunnelled by this RoE mapper as 64
 29 bits of data. This mode removes any line encoding and extracts/stores **.lenPack**
 30 octets from/to the source stream For a CPRI type source stream the RoE mapper
 31 does not de-interleave the I/Q samples. Likewise, as part of the decoding, the
 32 scrambling (when supported) shall also be undone.

33 Data that is not evenly divisible into 8-bit octets is not padded; instead, packet payloads
 34 are truncated to the last full octet, and the remainder bits are sent at the beginning of the
 35 next packet. This means, for example, that data from a 66-bit symbol at the end of a
 36 payload may be split across two packets with successive packet numbers.

37 A structure-agnostic RoE mapper can only generate one flow from one source.

1 7.1.1 Use of RoE control packets

2 Since the structure-agnostic RoE mapper encapsulates the entire data stream (with or
3 without line coding), there are no associated control packets for the structure-agnostic RoE
4 mapper. Any control or management information embedded in the data stream is not
5 interpreted by the RoE mapper but is passed through as binary data.

6 7.1.2 Structure-agnostic data packet

7 7.1.2.1 Sub type (subType) field

8 The **subType** field for a structure-agnostic data packet shall be set to the value identified
9 in Table 4-1.

10 7.1.2.2 Flow identifier (flowID) field

11 See sub-clause 0 and sub-clause 6.2.7.

12 7.1.2.3 Ordering information (orderInfo) field

13 The **orderInfo** field uses either a sequence number or time stamp as described in sub-
14 clause 4.4.4. When sequence numbers are used, an unexpected next sequence number value
15 indicates either a missing packet or an out of order packet. Packets received out of order
16 due to network traffic may be output in order if egress buffering is used and the out of order
17 packets are received within the window of egress buffering.

18 The sequence number is used to reflect the framing of the source data. The **orderInfo** used
19 for a given flow is determined by a parameter. This parameter is described in Table 6-5
20 and Table 6-6.

21 The bit rate for the stream is defined when the stream is initiated. The parameter **.lenPack**
22 defines the number of octets which are encapsulated in the RoE payload. This is reflected
23 in the length field in the common header.

24 All tunneling mode (**.mapperType**=0) packets shall use sequence numbers **seqNum** in the
25 **orderInfo** field. Only the **p-counter** field shall be used. The first sequence number in the
26 stream shall be 0x0000, and the **p-counter** shall monotonically increase by 1 with each
27 packet, wrapping to 0x0000 after reaching 0xFFFF.

28 For the line coding aware mode (**.mapperType**=1) and when the **orderInfo** contains a
29 sequence number, the overflow of the **p-counter** indicates a radio frame boundary. When
30 the **p-counter** field is 0, this indicates the start of the payload contained within the packet
31 is the start of the radio frame.

32 For example, let us assume there are 8 CPRI basic frames per RoE packet. The sequence
33 number related **orderInfo** parameter could be set as follows;

```
34       {
35           .seqNumPIncProp=1,
36           .seqNumPInc=8,
```

```

1      .seqNumPMax=38399,
2      .seqNumQIncProp=1,
3      .seqNumQInc=1, and
4      .seqNumQMax=4095.
5  }
```

6 Since the sequence number is incremented by 8 for each sent RoE data packet, the sequence
7 number p-counter would wrap every $150 \times 256 / 8$ sent packets i.e., after 150 hyper frames
8 worth of basic frames that equals exactly one 10ms radio frame. When the p-counter wraps,
9 the sequence number q-counter gets incremented by 1. The q-counter wraps on the 4096th
10 increment in order to cover 12-bit CPRI BFN. Similarly, if the **orderInfo** were counting
11 payload octets for a 32 octet basic frame the above example should be modified as
12 follow; **.seqNumPIncProp=2**, and **.seqNumPMax=12288799**. The number of octets per
13 RoE packet would be 8×32 i.e., 256 octets. The q-counter could now in maximum cover
14 11-bit CPRI BFN meaning **.seqNumQMax=2047**.

15 7.1.2.4 Length field

16 See sub-clause 4.4.3.

17 7.1.2.5 Payload field

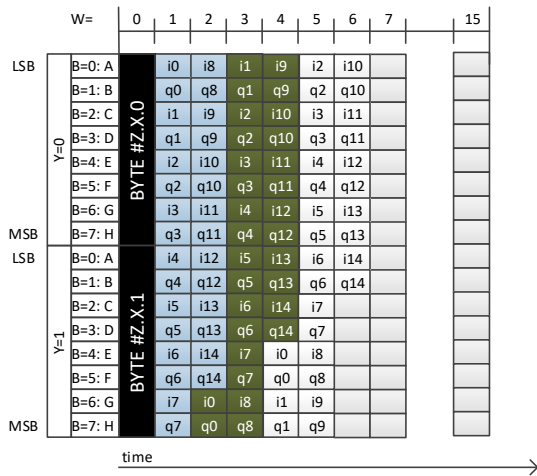
18 A structure-agnostic packet can have two payload types. When line coding aware mode is
19 off (simple tunneling mode **.mapperType=0**), the entire payload data shall be comprised
20 of the binary stream of bits without interpretation. When line coding aware mode is enabled
21 (**.mapperType=1**), the payload contains all the contents of the source stream, with the
22 exception of the line coding information.

23 Figure 7-1 below shows an example of how I/Q samples of CPRI basic frame appears in a
24 line coding aware structure-agnostic data packet, where;

- 25 a) Shows the originating CPRI basic frame.
- 26 b) Shows the common header (first 8 bytes in the packet).
- 27 c) Shows the remainder of the packet with I/Q samples.

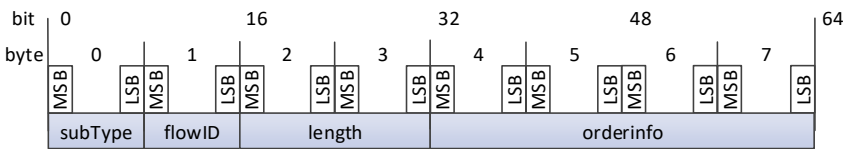
28
29

1 a)



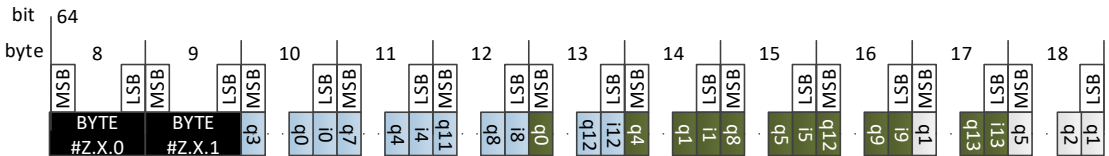
2

3 b)



4

5 c)



6

Figure 7-1 Example of how a CPRI basic frame appears in a structure-agnostic packet

8

9 7.2 Structure-aware RoE mapper

10 This sub-clause defines how the content of a CPRI packet is constructed in to and
11 deconstructed from basic frames, and how the CPRI control words are separated from the
12 data payload.

13 The content of a CPRI basic frame is divided into
14 **(de)mapper[(de)mapperID].numContainers** bit fields (i.e., containers). The container
15 parameters always fully describe a complete CPRI basic frame. The RoE packet payload
16 is then constructed based on the data extracted from the CPRI basic frames. A single
17 **(de)mapper[(de)mapperID].[containerID]** parameter set is described in Table 7-1.

7.2.1 Container definition

The **.flowID** identifies to which RoE mapper/ RoE de-mapper this container belongs. Typically the **.flowID** equates to an antenna carrier that is placed into separate RoE packets/flows. If the **.flowID** is set to NIL the **.flowID** shall be sent to the control process.

The **.lenSkip** describes the number of unused bits and the **.lenContainer** the number of actual payload bits per container (not including **.lenSkip**). Note that the **.lenSkip** bits are only valid when extracting/storing data from/to some other source/destination than RoE payload field such as the CPRI basic frame. When containers are stored into or read from the RoE payload field “skip bits” are not written or read.

The **.modulo** allows skipping containers and skipped containers are handled in the same way as **.lenSkip** bits. The modulo operation is applied to a sequence of input data that is counted from 0 to CPRI basic frame index. The container to select is matched comparing the **.index** to the output of the modulo operation. For example, to skip every second input container, set the modulo to 2 and set the index to zero. To turn off modulo logic, set the modulo to 0.

The above scheme allows constructing rather complex payload fields out of a CPRI basic frame as well as very simple ones. The container parameters are per direction; there may be different values for the RoE mapper and RoE de-mapper. A CPRI equivalent for a single container is basically an AxC Container. A series of container definitions can also be used to describe rough equivalents of an AxC Group and an AxC Container Group. The container definitions shall always describe one complete CPRI basic frame.

Table 7-1 - Parameters under RoE.Container branch

enParam	Parameters	Bits	Description
0	.(de)mapperID	8	The (de) mapperID this container belongs to. For containers that do not belong to a specific (de) mapperID but are processed separately, like CPRI control words, shall set the value to NIL flowID .
1	.flowID	8	The .flowID this mapper will insert into the generated RoE packet common header. On the mapper this parameter may be different from the .mapperID . On the de-mapper this parameter is equal to the .demapperID .
2	.lenSkip	8	The number of bits of reserved bits in front of the extracted container.
3	.lenContainer	10	The size of the container in bits.
4	.modulo	9	Values from 0 to 256. Value 0 turns off the modulo logic.

5	.index	8	Index to match after the modulo operation. Valid values from 0 to .modulo-1 .
---	---------------	---	--

1 **7.2.2 Structure-aware CPRI data packet**

2 **7.2.2.1 Sub type (subType) field**

3 The **subType** field for a structure-aware data packet shall be set to the value identified in
4 Table 4-1.

5 **7.2.2.2 Flow identifier (flowID) field**

6 See sub-clause 0 and sub-clause 6.2.7.

7 **7.2.2.3 Ordering information (orderInfo) field**

8 See sub-clause 4.4.4.

9 **7.2.2.4 Length field**

10 See sub-clause 4.4.3.

11 **7.2.2.5 Payload field**

12 I/Q samples are packed without gaps in the payload, un-interleaved, with **.sampleWidth**
13 bits MSB first and with I followed by Q. Figure 7-2 below shows an example of how I/Q
14 samples of CPRI basic frames are mapped into a single flow in a structure-aware data
15 packet, where;

16 a) Shows the originating CPRI basic frames (total series of four).

17 b) Shows the common header (first 8 bytes in the packet).

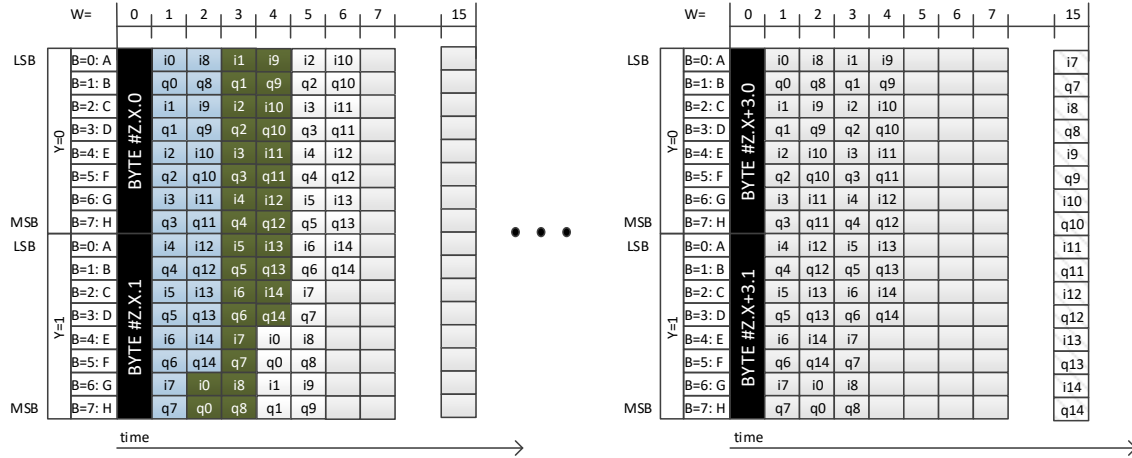
18 c) Shows the remainder of the packet with I/Q samples (assuming total 4 Basic
19 Frames of 15 bit samples).

20 The control word mapping into RoE control packets (bytes XZ.X.0, Z.X.1, etc.,) are not
21 sent within RoE structure-aware data packets See sub-clause 7.2.3.

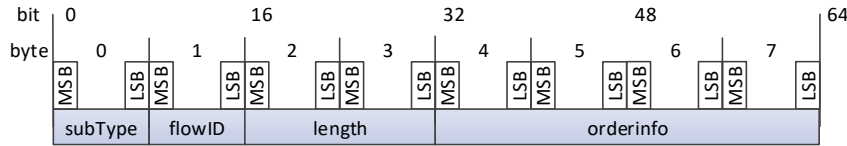
22

23

1 a)



3 b)



5 c)

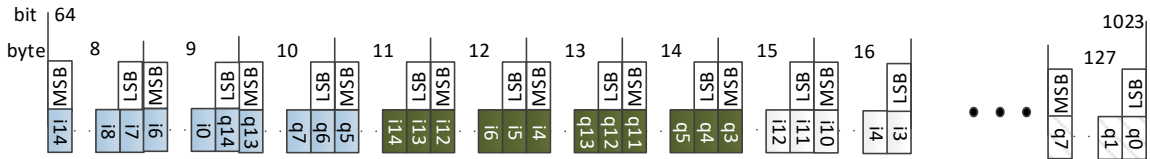


Figure 7-2 - Example mapping CPRI basic frame using structure-aware packet

7.2.3 Control data and the control process

CPRI defines sub-parts of basic frames as AxC containers. If containers contain control data, those are handled by a “control process” whose responsibility is to collect the control data (based on the container rules) before constructing a separate RoE control packet or other Ethernet packet (e.g. in a case of CPRI Fast C&M channel). The control process is responsible for meeting possible timing constraints on delivering control data within the required time frame. The behavior of the “control process” shall be defined by each RoE mapper that makes use of the control data.

When the **.flowID** in a container definition (see sub-clause 7.2.1) is set to NIL **flowID**, the control process shall set the **.flowID** in the RoE control packet header to the appropriate value according to the matching control process container construction (see sub-clause 7.2.8).

7.2.4 CPRI C-plane handling

The structure-aware RoE mapper has multiple “control process” RoE mappers to process CPRI Hyper-frame control words. Table 7-2 lists the control process mappers, their respective naming/variable prefixes and the sub-channels to which they (typically) refer. The RoE mapper definitions and variables have the common prefix **.ctrl**.

Table 7-2 - Control process RoE mappers for CPRI control words

Mapper description	CPRI control word sub-channels	Mapper prefix
Synchronization and L1 protocol fields	0 & 2	mapper[mapperID].ctrl.sync_l1
Slow C&M channel	1	mapper[mapperID].ctrl.slow
Fast C&M channel	p->63	mapper[mapperID].ctrl.fast
Combined VSD and Ctrl_AxC channels	4->7 & 16->p-1	mapper[mapperID].ctrl.axc_vsd

The control words the “control process” RoE mappers operate upon are structured in the same way as CPRI structures it. Areas not extracted or transported to/from the CPRI stream are assumed to be all zero (0). Figure 7-3 illustrates the CPRI Hyper-frame control words collection and construction for the “control process”. From the processing point of view control words are always processed in the order they arrive i.e., the first element is the control word 0 ($N_s=X_s=0$), the second element is the control word 1 ($N_s=1, X_s=0$), etc. The control process mappers are applied to the control words as they arrive and then stored into the memory as a dense array for possible second stage processing (such as placing data into RoE data packet payload field).

There are 36 control words in one hyper-frame reserved for future interface protocol extensions. Reserved words are completely filled with reserved bits (reserved bits are marked with “r”). This means that a transmitter shall send 0’s for bits marked with “r”, and the receiver shall not interpret bits marked with “r”. (transmit: r = 0, receiver: r = don’t care).

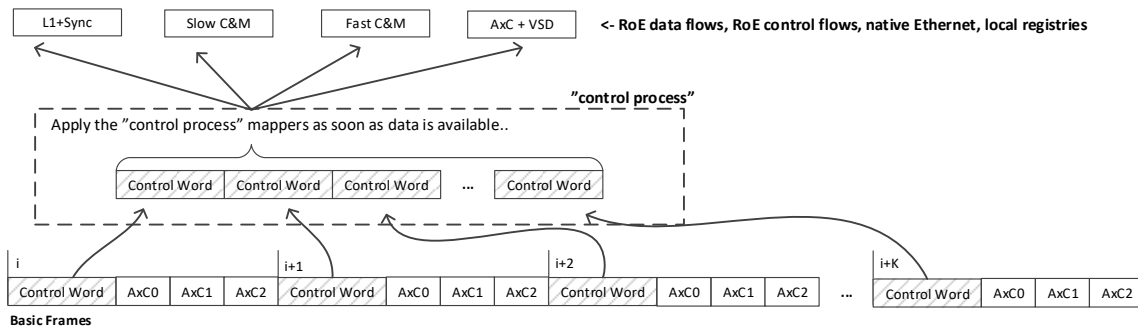


Figure 7-3 - CPRI basic frame and "control process" interaction

7.2.5 Synchronization and L1 protocol fields

CPRI Synchronization and L1 protocol fields are not transported over the RoE. They are only provided for the local use by the “control process”. The following information shall be supported (using CPRI control word notation):

- a) HFN (Hyper-frame number) at location Z.64.0 i.e., control word 64.
- b) BFN (CPRI 10 ms frame number) at locations Z.128.0 and Z.192.0 i.e., control words 128 and 192.
- c) Protocol version at location Z.2.0 i.e., control word 2.
- d) HDLC bit rate at location Z.66.0 i.e., control word 66.
- e) L1 signalling at location Z.130.0 i.e., control word 130.
- f) Ethernet pointer at location Z.194.0 i.e., control word 194.

The control process extracts L1 protocol fields c to f and populates them into the CPRI parameter list shown in Table 6-3. Fields a & b are implicit in the usage of **timingInfo**.

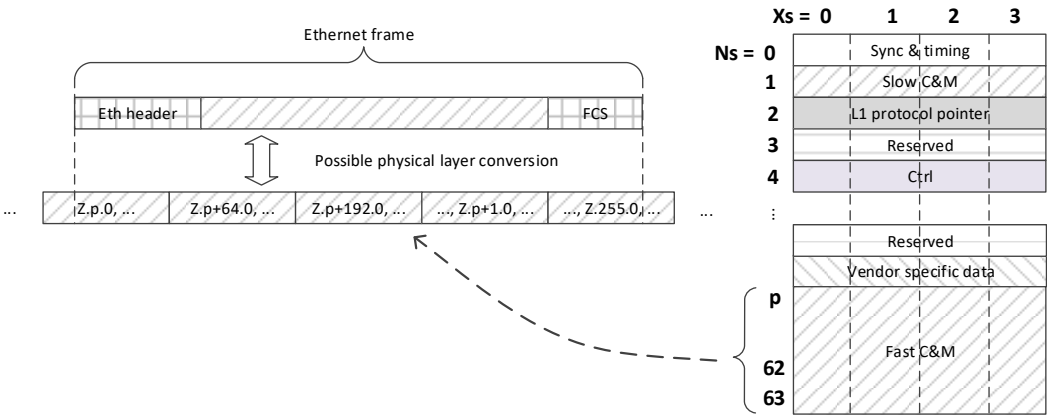
7.2.6 Handling of Fast C&M

The “control process” extracts the control words for the Fast C&M channel and creates an appropriate Ethernet packet out of the extracted Ethernet packet data (e.g., discarding SSD, ESD, IDLE bit sequences and line coding). The Fast C&M channel is sent and received as native Ethernet traffic. The used Physical Coding Sublayer (PCS) shall be according to the underlying link and the RoE mapper is responsible of doing possible required conversions.

Since Fast C&M is in a fixed position, no parameters are required. The Fast C&M channel simply extracts and uses CPRI control word Z.194.0 to determine the supported Ethernet rate.

Figure 7-4 illustrates an overly simplified construction of Fast C&M Ethernet packet based on the configuration given above. If the Fast C&M embedded Ethernet packet does not align into CPRI basic frame boundaries or does not fit into a single basic frame, it is the “control process” responsibility to buffer the required amount of data to make a successful conversion between CPRI control words and the native Ethernet packets.

1



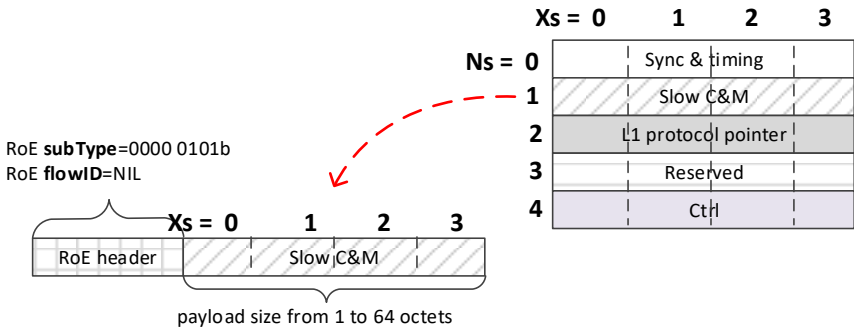
2

3 **Figure 7-4 – A simple example of CPRI fast C&M transport over**
4 **native Ethernet**

5 **7.2.7 Handling of Slow C&M**

6 The Slow C&M channel is mapped into its own **subType**. Since Slow C&M is in a fixed
7 position, very few parameters are required. The container & payload size is determined
8 from Z.66 and the CPRI line rate.

9 The Slow C&M channel uses the NIL value for **flowID** in the RoE control packet header.



10

11 **Figure 7-5 - Example RoE packet for transporting CPRI Slow C&M flow**

12 **7.2.8 Handling of other Ax_Ctrl and vendor specific words**

13 Figure 7-3 summarizes the parameters for the “control process” definitions and their sizes

1

Table 7-3 - CPRI control word RoE mapper container parameters

enParam	Variable	Bits	Description
0	.ctrlWordID	3	Each control word has a unique identifier.
1	.flowID	3	The .flowID used in the control process container construction for generated RoE packets. The .flowID is equal to the used container index that describes the control process mapper parameter and has a valid range of 0 to 7.
2	.cwSel	3	A three-bit mask for selecting sub-channel words (Xs) to extract. Bit 0 corresponds to Xs word with index 0 and Bit 3 to Xs =3. The selected sub-channel words equal to “logical or” of corresponding sub-channel masks.
3	.cwStart	6	The first sub-channel allocated for parser. Valid range is from 0 to 63.
4	.cwSize	6	Number of consecutive sub-channels allocated for the control data. Valid range is from 0 to 63.
5	.filterMode	3	Controls the generation of the RoE packets from the extracted CPRI control words.
6	.hfnModulo	8	Modulo operation applied to current CPRI hyper-frame number. Valid range from 1 to 150. Whether the modulo is applied depends on the .filterMode setting.
7	.hfnIndex	8	The index to match after the modulo operation. Valid range from 0 to 149.
8	.cwOffset	12	The offset into the extracted control word buffer for masking and value comparison operation. Valid range from 0 to 4095. Whether the offset-based mask and value comparison is applied depends on the .filterMode setting.
9	.cwValue	32	The value to compare against after the mask operation.

10	.cwMask	32	The bit mask (logical AND) for the offset-based comparison operation.
11-255			Reserved

All other Control process RoE mappers for other C-plane information use the following common container construction:

```

{
    .ctrlWordID
    .flowID
    .cwSel
    .cwStart
    .cwSize
    .filterMode
    .hfnModulo
    .hfnIndex
    .cwOffset
    .cwValue
    .cwMask
}

```

The **.cwSel** is a four bit mask for selecting sub-channel words (Xs) to extract. The selected sub-channel words equal to “logical or” of corresponding sub-channel masks. See Table 7-4 for the mask values.

Table 7-4 - Sub-channel word bit masks

Binary mask	Sub-channel word (Xs index)
0000b	sub-channel processing disabled – container not in use.
0001b	Xs = 0
0010b	Xs = 1
0100b	Xs = 2
1000b	Xs = 3

The **.cwStart** defines the start of sub-channel (Ns) and has the valid range from 0 to 63. The **.cwSize** defines the number of extracted sub-channels minus one and the valid range is from 0 to 63. The mask defined by **.cwSel** applies to the “area” defined by the **.cwStart** and **.cwSize**. The specific RoE mapper definitions may have more specific restriction to the ranges.

If there are multiple containers the areas they define shall not overlap each other.

The **.flowID** maps to RoE header **flowID** field.

The **.filterMode** specifically concerns the control process of the RoE mapper when it generates an RoE packet. See Table 7-5 for further details.

Note that the control packet is generated and sent as soon as the required amount of control word data has been collected as described by the container definition. The packetizer shall not wait until the entire hyper-frame has been received.

Table 7-5 - Hyper-frame filtering options

.filterMode	Description
0	Filtering is disabled. An RoE control packet is generated on every Hyper-frame
1	Non-zero content i.e., the extracted content has non-zero values.
2	Periodic generation according to modulo logic.
3	Pattern match.
4-7	reserved

Note that when the **.filterMode** is set to 0 (filtering disabled), the packet is generated and sent as soon as the required amount of control word data has been collected. The packetizer shall not wait until the entire hyper-frame has been received.

If the **.filterMode** is set to 1, then the **.cwOffset** and the **.cwMask** concern the non-zero content check. The **.cwOffset** has a valid range from 0 to 4095. The **.cwMask** is a 32-bit value. The non-zero content is applied to the hyper-frame after applying other parser options such as **.cwSel**, **.cwStart** and **.cwSize**, and there is enough buffered data for the match operation. The offset is relative to the extracted (dense array or buffer of) control words. The match is true when the 32-bit value extracted from the memory is non-zero after applying a ‘logical AND’ to it using **.cwMask**.

When the **.filterMode** is set to 2, the **.hfnModulo** operates on the entire extracted (Hyper-frame size) CPRI control words area and combined with **.hfnIndex** allows selecting specific Hyper-frames for further processing. The **.hfnModulo** has the valid range from 1 to 150. The **.hfnIndex** has the valid range from 0 to 149. The modulo logic is synchronized with the current hyper-frame Number (HFN). For example, **.hfnModulo** would select Hyper-frame control words for processing when the remainder of the modulo operation is equal to **.hfnIndex**.

If the **.filterMode** is set to 3, then the **.cwOffset**, **.cwValue** and **.cwMask** concern the pattern match. The **.cwOffset** has valid range from 0 to 4095. Both **.cwValue** and **.cwMask** are maximum 32 bit values. The pattern match is applied to the Hyper-frame after applying other parser options such as **.cwSel**, **.cwStart** and **.cwSize**, and there is enough buffered data for the match operation. The offset is relative to the extracted (dense array or buffer of) control words. The match is true when the 32-bit value extracted from the memory equals to the **.cwValue** after applying a “logical AND” to it using the **.cwMask**. Note that in cases where reading of the 32-bit value at the **.cwOffset** would partially be out of bound

of the buffered data, then the **.cwMask** and the **.cwValue** need to be set according the handle the situation. For instance, if the buffered data size is 16 octets and the **.cwOffset** is set to 15, then the three least significant octets of the **.cwMask** and the **.cwValue** are set to 0.

Whenever parameter configurations refer to a value “p” that refers to the pointer in CPRI control word Z.194.0 indicating the start of Fast C&M channel sub-channels.

7.3 Native RoE time domain packet mapper

Native RoE time domain packets contain time domain I/Q for a given flow.

7.3.1 Native RoE time domain packet

This sub-clause describes the native RoE time domain packet format. The packet payload carries a single flow or a group flow of radio sample data between two RoE end-points. The common RoE frame header content is described in sub-clause 4.4.

7.3.1.1 Sub type (subType) field

The **subType** field for an native RoE time domain data packet shall be set to the value identified in Table 4-1.

7.3.1.2 Flow identifier (flowID) field

See sub-clause 0 and sub-clause 6.2.7.

7.3.1.3 Ordering information (orderInfo) field

See sub-clause 4.4.4.

7.3.1.4 Length field

See sub-clause 4.4.3.

7.3.1.5 Payload field

See sub-clause 4.4.5 for the generic definition. The packet payload carries a single flow or a group of flows between RoE nodes. Time domain I/Q samples are packed without gaps in the payload, un-interleaved, with **.sampleWidth** bits MSB first and with I followed by Q.

Figure 7-6 below shows an example of how an I/Q data with **.sampleWidth**=15 is mapped into a single flow in a Native RoE time domain packet, where:

- a) Shows the common header (first 8 bytes in the packet)
- b) Shows the remainder of the packet (only two complete 15-bit I/Q sample pairs are shown)

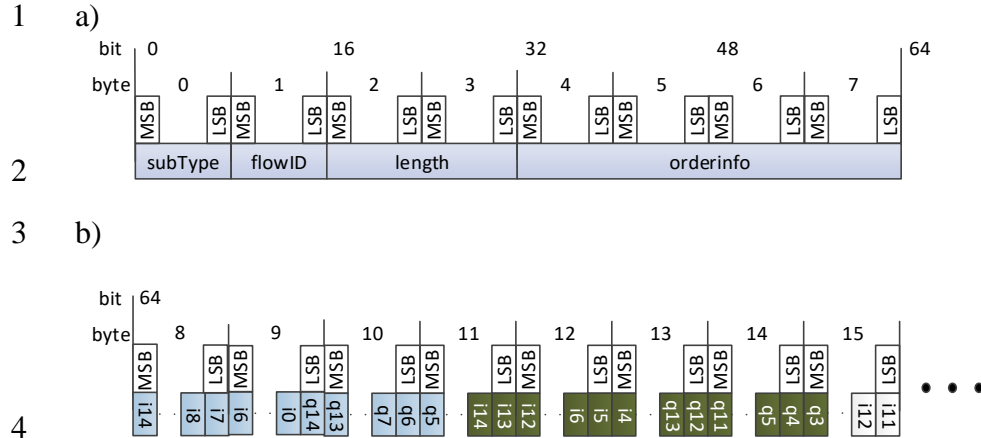


Figure 7-6 – Native Time Domain RoE packet example with I/Q sample payload

7.3.2 Control data

In RoE time domain native mode, control data is sent by TLVs or by out-of-band communication, which is out of scope.

7.4 Native RoE frequency domain packet mapper

Native RoE frequency domain packets contain frequency domain I/Q for a given flow.

7.4.1 (i)FFT definition

Frequency domain multiple access (FDMA) communication systems use Fourier series based functions to translate between the time and frequency domain. In the case of Evolved Universal Terrestrial Radio Access (E-UTRA), this is defined in 3GPP Specification TS36.211.

The native RoE frequency domain packet mapper allows for this function to be performed in the RoE node. Each function performs conversion from time to frequency (FFT) and frequency to time (iFFT) symmetrically.

Table 7-4 - Parameters under mapper[mapperID].[fftID] branch

enParam	Parameter	Bits	Default	Description
0	.fftID	8	0	Each (i)FFT in each node has a unique identifier.
1	.scFFT	16	1200	The number of sub-carriers the (i)FFT operates on, $N_{RB}^{UL} N_{SC}^{RB}$ where N_{RB}^{UL} = number of resource blocks (RB) in a symbol and N_{SC}^{RB} = number of FDMA subcarriers in a RB.

2	.fftSize	16	2048	The size (in bins) of the (i)FFT.
3	.cpLen0	16	160	Identifies the cyclic prefix length $N_{CP,l}$ when $l = 0$. Where symbols are transmitted in increasing order of l , starting with $l=0$.
4	.cpLenOther	16	144	Identifies the cyclic prefix length $N_{CP,l}$ for symbols when $l > 0$. Where symbols are transmitted in increasing order of l , starting with $l=0$.
5-255				Reserved

1 7.4.2 Native RoE frequency domain packet

2 This sub-clause describes the native RoE frequency domain packet format. The packet
3 payload carries a single flow or a group flow of radio sample data between two RoE end-
4 points. The common RoE frame header content is described in sub-clause 4.4.

5 7.4.2.1 Sub type (subType) field

6 The **subType** field for an native RoE frequency domain data packet shall be set to the value
7 identified in Table 4-1.

8 7.4.2.2 Flow identifier (flowID) field

9 See sub-clause 0 and sub-clause 6.2.7.

10 7.4.2.3 Ordering information (orderInfo) field

11 See sub-clause 4.4.4.

12 7.4.2.4 Length field

13 See sub-clause 4.4.3. RoE frequency domain packets contain (i)FFT information. The iFFT
14 takes **.lenPack** x **.scFFT** I/Q samples and produces **.lenPack** x **.fftSize** I/Q samples. The
15 FFT takes **.lenPack** x **.fftSize** I/Q samples and produces **.lenPack** x **.scFFT** I/Q samples.

16 7.4.2.5 Payload field

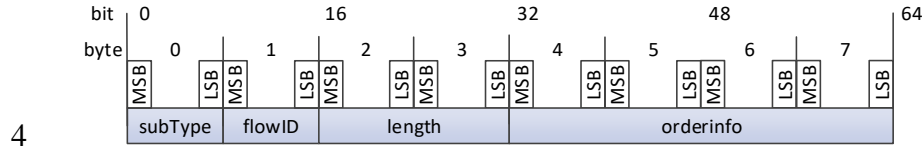
17 See sub-clause 4.4.5 for the generic definition. The packet payload carries a single flow or
18 a group of flows between RoE nodes. Frequency domain I/Q samples are packed without
19 gaps in the payload, un-interleaved, with **.sampleWidth** bits MSB first and with I followed
20 by Q.

21 Figure 7-7 below shows an example of how an I/Q data with **.sampleWidth**=15 is mapped
22 into a single flow in a Native RoE packet, where:

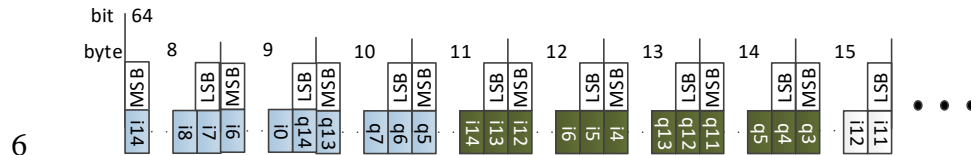
- 23 a. Shows the common header (first 8 bytes in the packet)

- 1 b. Shows the remainder of the packet (only two complete 15-bit I/Q sample pairs are
2 shown)

3 a)



5 b)



7 **Figure 7-7 – Native RoE frequency domain packet example with**
8 **I/Q sample payload**

9 7.4.3 Control data

10 In native mode, control data is sent by TLVs or by out-of-band communication (outside
11 RoE) scope of work.

12 7.4.4 PRACH definition

13 The Physical random access channel (PRACH) is used to obtain up-link synchronization
14 between UE and eNB. For Evolved Universal Terrestrial Radio Access (E-UTRA), it is
15 described in 3GPP Specification TS36.211. The PRACH function is only relevant in the
16 uplink direction.

17 **Table 7-6 - Parameters mapper[mapperID].[fftID].[PRACH] branch**

enParam	Parameter	Bits	Default	Description
0	.prachID	8	0	Each PRACH in each node has a unique identifier.
1	.prachConfigIndex	8	0	The PRACH configuration index, which sets the preamble format, and system frame number & sub-frame number position(s) of the PRACH in time in radio frames. Set by SIB2 prach-ConfigIndex in E-UTRA.
2	.prachOffset	16	0	Sets the PRACH position in frequency / subcarrier position, $n_{PRBOffset}^{RA}$ and is expressed as a physical resource block number.

				Set by SIB2, prach-FreqOffset in E-UTRA.
3	.prachSampleWidth	8	16	Indicates the total number of bits in each I/Q sample in bits. By default, 16-bit and 16-bit Q width is assumed.
4-255				Reserved

1 **7.4.5 Native RoE PRACH packet**

2 This sub-clause describes the native RoE PRACH packet format. The packet payload
3 carries a single flow or a group flow of radio sample data between two RoE end-points.
4 The common RoE frame header content is described in sub-clause 4.4.

5 **7.4.5.1 Sub type (subType) field**

6 The **subType** field for a PRACH data packet shall be set to the value identified in Table
7 4-1.

8 **7.4.5.2 Flow identifier (flowID) field**

9 See sub-clause 0 and sub-clause 6.2.7.

10 **7.4.5.3 Ordering information (orderInfo) field.**

11 See sub-clause 4.4.4.

12 **7.4.5.4 Length field**

13 See sub-clause 4.4.3.

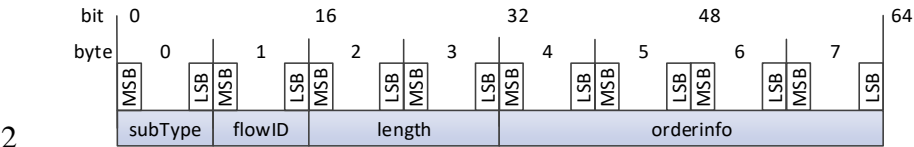
14 **7.4.5.5 Payload field**

15 See sub-clause 4.4.5 for the generic definition. The packet payload carries a single flow or
16 a group of flows between RoE nodes. I/Q samples (in time or frequency) are packed
17 without gaps in the payload, un-interleaved, with **.sampleWidth** bits MSB first and with I
18 followed by Q.

19 Figure 7-8 below shows an example of how an I/Q data with **.sampleWidth=15** is mapped
20 into a single flow in a Native RoE packet, where:

- 21 a) Shows the common header (first 8 bytes in the packet)
- 22 b) Shows the remainder of the packet (only two complete 15-bit I/Q sample pairs are
- 23 shown)

1 a)



3 b)

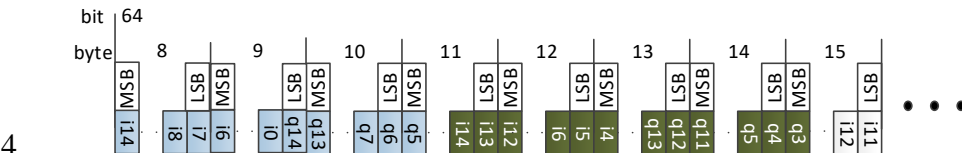


Figure 7-8 – Native RoE PRACH packet example

8 RoE control packet header format

This sub-clause documents the first octets in a frame that is common to RoE control packets. Figure 8-1 illustrates the frame format and its fields. The RoE control packet frame format follows the generic RoE frame format defined in sub-clause 4.3 unless stated otherwise in following sub-clauses.

Note that message acknowledgement or negative-acknowledgement process is out of RoE scope.

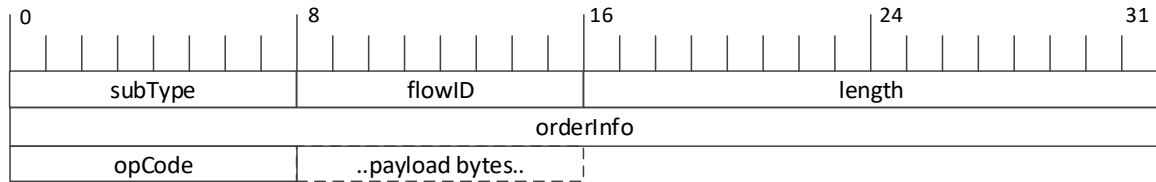


Figure 8-1 - RoE control packet common frame format

8.1.1 Sub type (subType) field

The **subType** field for an RoE Control Packet shall be set to the value identified in Table 4-1.

8.1.2 opCode field (opCode)

The **opCode** is a field of 8 bits and defines additional control sub types. This document reserves Control Packet **opCode** values listed in Table 8-1.

Table 8-1 – RoE Control Packet opCode values

OpCode	Function	Description
0000 0000b	RoE OAM TLV	Operation, administration and management packet with type, length and value.
0000 0001b	Ctrl_AxC words	Ctrl_AxC data.
0000 0010b	VSD	Vendor specific control packet.
0000 0011b	Timing control packet	A packet containing both timeStamp and seqNum .
0000 0100b – 1111 1111b	Reserved	Reserved control packet opCodes.

8.1.3 OAM TLV control packet

8.1.3.1 Opcode (opCode) field

The **opCode** field for an RoE Control Packet shall be set to the value identified in Table 8-1.

1 8.1.3.2 Flow identifier (flowID) field

2 The **flowID** field shall be set to the NIL value in the RoE control packet header. (see
3 Table 4-2).

4 8.1.3.3 Ordering information (orderInfo) field

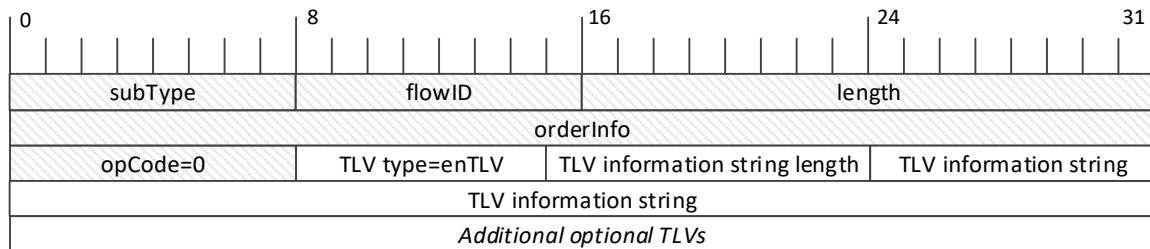
5 Control packets shall always use a sequence number for the **orderInfo** field. See sub-clause
6 4.4.4. Note that sequence numbers may behave differently between RoE control packets
7 and their associated RoE data packet flows. For example the **.seqNumPMax**
8 and **.seqNumPVal** can be different for RoE control packets and data packets.

9 8.1.3.4 Length field

10 See sub-clause 4.4.3.

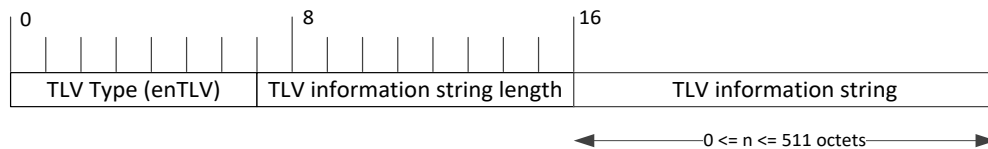
11 8.1.3.5 Payload

12 The packet's payload field contains one or multiple TLVs.



13
14 **Figure 8-2 - OAM TLV packet format**

15 A single TLV has 3 components, tag, length and value (the TLV information string). The
16 basic format and general field definition is defined in IEEE Std 802.1ABTM, IEEE Standard
17 for Local and metropolitan area networks – Station and Media Access Control Connectivity
18 Discovery. Figure 8-3 shows the basic TLV format.



19
20 **Figure 8-3 - Basic TLV format**

21 RoE object types are enumerated by the **.enTLV** field from Table 6-1. For RoE,
22 the **.enTLV** field is used to populate the TLV type in the basic TLV format. The TLV
23 information string length field is 9 bits long and is populated with the length of the TLV
24 information string (n) in octets, where n is 0 <= n <= 511. RoE parameters within this

1 document are enumerated by the field **.enParam** and shown in the tables contained within
2 sub-clause 6.2.

3 The TLV information string contains the following bytes (with parameters LSB aligned)
4 in sequence;

- 5 — **.enParam** (1 byte), the enumerated parameter type.
- 6 — the ID of the object (1 byte).
- 7 — the **.containerID** of the container (1 byte) for container type objects.
- 8 — the value (1 – 4 bytes). The parameter value bits are mapped into the least
9 significant bits of an integer number of bytes in network byte order.

10 Multiple TLVs may be concatenated into a single OAM TLV payload.

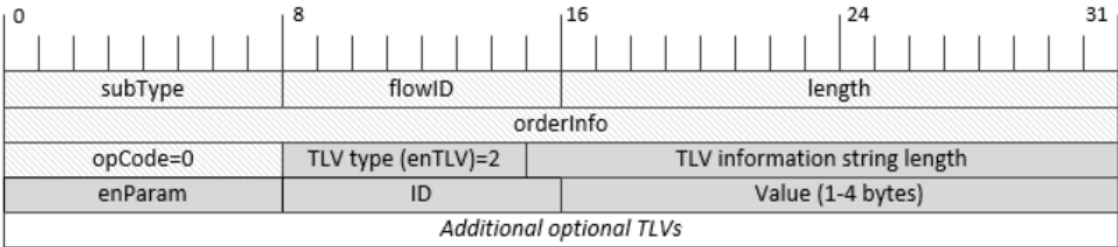


Figure 8-4 - TLV concatenation

13 **8.1.4 Ctrl_AxC control packet**

14 **8.1.4.1 Opcode (opCode) field**

15 The **opCode** field for an RoE Control Packet shall be set to the value identified in Table
16 8-1.

17 **8.1.4.2 Flow identifier (flowID) field**

18 The **flowID** field shall be set to the NIL value in the RoE control packet header. (see Table
19 4-2).

20 **8.1.4.3 Ordering information (orderInfo) field**

21 Control packets shall always use a sequence number for the **orderInfo** field. See sub-clause
22 4.4.4. Note that sequence numbers may behave differently between RoE control packets
23 and their associated RoE data packet flows. For example the **.seqNumPMax**
24 and **.seqNumPVal** can be different for RoE control packets and data packets.

25 **8.1.4.4 Length field**

26 See sub-clause 4.4.3.

8.1.4.5 Payload

The payload contains CPRI stream Ctrl_AxC control words according to the **mapper[mapperID].ctrl.axc_vsd** container definition as described in sub-clause 7.2.8. The size of the extracted control word is determined by the CPRI option in use (i.e., line bit rate, see CPRI specification V7.0 Table 3). The **.cwStart** shall be within range from 4 to 7. The sum of **.cwStart** and **.cwSize** shall be less than 8.

The Ctrl AxC packet's common RoE header **flowID** gets populated according to **mapper[mapperID].ctrl.axc_vsd** container **.flowID** definition. The **orderInfo** field shall contain a sequence number.

The control packet's payload is populated in a network byte order as extracted by the "control process" (see sub-clause 7.2.3 and illustrated in CPRI specification V7.0 Figure 16).

8.1.5 VSD packet control packet

8.1.5.1 Opcode (opCode) field

The **opCode** field for an RoE Control Packet shall be set to value identified in Table 8-1.

8.1.5.2 Flow identifier (flowID) field

The **flowID** field shall be set to the NIL value in the RoE control packet header. (see Table 4-2).

8.1.5.3 Ordering information (orderInfo) field

Control packets shall always use a sequence number for the **orderInfo** field. See sub-clause 4.4.4. Note that sequence numbers may behave differently between RoE control packets and their associated RoE data packet flows. For example the **.seqNumPMax** and **.seqNumPVal** can be different for RoE control packets and data packets.

8.1.5.4 Length field

See sub-clause 4.4.3.

8.1.5.5 Payload

The payload contains CPRI stream VSD control words according to the **mapper[mapperID].ctrl.axc_vsd** container definition as described in sub-clause 7.2.8. The size of the extracted control word is determined by the CPRI option in use (i.e., line bit rate, see CPRI specification V7.0 Table 3). The **.cwStart** shall be within range from 16 to p (the pointer p as defined in CPRI specification V7.0 sub-clause 4.2.7.4). The sum of **.cwStart** and **.cwSize** shall be less than p.

The Ctrl AxC packet's common RoE header **flowID** gets populated according to **mapper[mapperID].ctrl.axc_vsd** container **.flowID** definition. The **orderInfo** field shall contain a sequence number.

The control packet's payload is populated in a network byte order as extracted by the "control process" (see sub-clause 7.2.3 and illustrated in CPRI specification V7.0 Figure 16).

8.1.6 Timing control packet

8.1.6.1 Opcode (opCode) field

The **opCode** field for an RoE Control Packet shall be set to the value identified in Table 8-1.

8.1.6.2 Flow identifier (flowID) field

The **flowID** field shall be set to the NIL value in the RoE control packet header. (see Table 4-2).

8.1.6.3 Ordering information (orderInfo) field

Control packets shall always use a sequence number for the **orderInfo** field. See sub-clause 4.4.4. Note that sequence numbers may behave differently between RoE control packets and their associated RoE data packet flows. For example the **.seqNumPMax** and **.seqNumPVal** can be different for RoE control packets and data packets.

8.1.6.4 Length field

See sub-clause 4.4.3.

8.1.6.5 Payload

The timing packet is a control packet which contains both the **seqNum** and **timeStamp** fields. The **.targetTimestamp** contains the **orderInfo** as a **timeStamp** and **.targetSeqnum** contains the associated **orderInfo** in **seqNum** format as defined in sub-clause 4.4.4.. The **orderInfo** field shall contain a sequence number. The **.targetFlowID** identifies which RoE flow the timing packet concerns. The **.flowID** in the RoE common header is set to the NIL **flowID**.

The Timing control packet contains a correlation reference of **seqNum** to **.targetSeqnum** and **.timeStamp** to **.targetTimestamp** and it is to be received ahead of time compared to a corresponding RoE data packet in the associated flow. Thus, the Timing control packet has a **.flowID** equal to associated **.targetFlowID** and a timestamp sequence number equal to the associated **.targetSeqnum**. The maximum time window the **.targetTimestamp** can point to in the future is limited by the **.acceptTimeWindow**.

Upon receiving a timing control packet and its corresponding RoE data packet, a receiver can compare its own understanding of the presentation time for a given sequence number in a given flow to the corresponding presentation time the received RoE data packet. Based on this information, a receiver may adjust its internal clocks and rates accordingly.

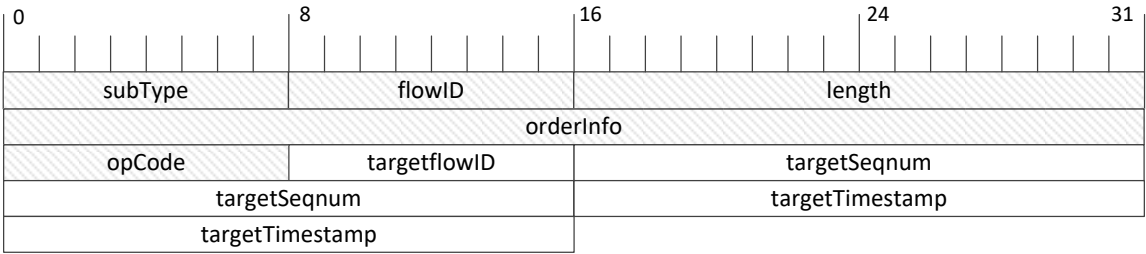


Figure 8-5 - Timing packet format

Annex A (normative) Protocol implementation conformance statement (PICS) proforma

Annex A.1 Introduction

The supplier of a device implementation that is claimed to conform to this standard is required to complete a protocol implementation conformance statement (PICS) proforma.

A completed PICS proforma is the PICS for the implementation in question. The PICS is a statement of which capabilities and options of this standard have been implemented. The PICS proformas for each of the packages defined in Clause 4 are included in annexes to that clause. The PICS can be used for a variety of purposes by various parties, including the following:

- a) As a checklist by the protocol implementer, to reduce the risk of failure to conform to the standard through oversight
- b) As a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma, by the supplier and acquirer, or potential acquirer, of the implementation
- c) As a basis for initially checking the possibility of interworking with another implementation by the user, or potential user, of the implementation (note that, while interworking can never be guaranteed, failure to interwork can often be predicted from incompatible PICS)
- d) As the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation, by a protocol tester

Annex A.1.1 Abbreviations and special symbols

The following symbols are used in the PICS proforma:

M	mandatory field/function
!	negation
O	optional field/function
O.<n>	optional field/function, but at least one of the group of options labeled by the same numeral <n> is required
O/<n>	optional field/function, but one and only one of the group of options labeled by the same numeral <n> is required
X	prohibited field/function
<item>:	simple-predicate condition, dependent on the support marked for <item>
<item1>*<item2>:	AND-predicate condition, the requirement needs to be met if both optional items are implemented
<item1>+<item2>:	OR-predicate condition, the requirement needs to be met if at least one of optional item is implemented

Annex A.1.2 Instructions for completing the PICS proforma

The first part of the PICS proforma, Implementation Identification and Protocol Summary, is to be completed as indicated with the information necessary to identify fully both the supplier and the implementation.

The main part of the PICS proforma is a fixed-format questionnaire divided into sub-clauses, each containing a group of items. Answers to the questionnaire items are to be provided in the right-most column, either by simply marking an answer to indicate a restricted choice (usually Yes, No, or Not Applicable) or by entering a value or a set or range of values. (Note that there are some items where two or more choices from a set of possible answers can apply; all relevant choices are to be marked.)

Each item is identified by an item reference in the first column; the second column contains the question to be answered; the third column contains the reference or references to the material that specifies the item in the main body of the standard; the fourth column contains values and/or comments pertaining to the question to be answered. The remaining columns record the status of the items—whether the support is mandatory, optional, or conditional—and provide the space for the answers.

The supplier may also provide, or be required to provide, further information, categorized as either Additional Information or Exception Information. When present, each kind of further information is to be provided in a further sub-clause of items labeled A<i> or X<i>, respectively, for cross-referencing purposes, where <i> is any unambiguous identification for the item (e.g., simply a numeral); there are no other restrictions on its format or presentation.

A completed PICS proforma, including any Additional Information and Exception Information, is the protocol implementation conformance statement for the implementation in question.

Note that where an implementation is capable of being configured in more than one way, according to the items listed under Major Capabilities/Options, a single PICS may be able to describe all such configurations. However, the supplier has the choice of providing more than one PICS, each covering some subset of the implementation's configuration capabilities, if that would make presentation of the information easier and clearer.

Annex A.1.3 Additional information

Items of Additional Information allow a supplier to provide further information intended to assist the interpretation of the PICS. It is not intended or expected that a large quantity be supplied, and the PICS can be considered complete without any such information. Examples might be an outline of the ways in which a (single) implementation can be set up to operate in a variety of environments and configurations or a brief rationale, based perhaps upon specific application needs, for the exclusion of features that, although optional, are nonetheless commonly present in implementations.

References to items of Additional Information may be entered next to any answer in the questionnaire and may be included in items of Exception Information.

Annex A.1.4 Exception information

It may occasionally happen that a supplier wishes to answer an item with mandatory or prohibited status (after any conditions have been applied) in a way that conflicts with the indicated requirement. No preprinted answer is found in the Support column for this; instead, the supplier is required to write into the Support column an X<i> reference to an item of Exception Information and to provide the appropriate rationale in the Exception item itself.

An implementation for which an Exception item is required in this way does not conform to this standard. Note that a possible reason for the situation described above is that a defect in the standard has been reported, a correction for which is expected to change the requirement not met by the implementation.

Annex A.1.5 Conditional items

The PICS proforma may contain conditional items. These are items for which both the applicability of the item itself and its status if it does apply—mandatory, optional, or prohibited—are dependent upon whether certain other items are supported.

Individual conditional items are indicated by a conditional symbol of the form “<item>:<s>” in the Status column, where “<item>” is an item reference that appears in the first column of the table for some other item, and “<s>” is a status symbol, M (Mandatory), O (Optional), or X (Not Applicable).

If the item referred to by the conditional symbol is marked as supported, then

- a) The conditional item is applicable,
- b) Its status is given by “<s>”, and
- c) The support column is to be completed in the usual way.

Each item whose reference is used in a conditional symbol is indicated by an asterisk in the Item column.

Annex A.1.6 Identification

Annex A.1.6.1 Implementation identification

Supplier (Note 1)	
Contact point for queries about the PICS (Note 1)	

Implementation Name(s) and Version(s) (Notes 1 and 3)	
Other information necessary for full identification – e.g., name(s) and version(s) of machines and/or operating system names (Note 2)	
<p>NOTE 1 – Required for all implementations.</p> <p>NOTE 2 – May be completed as appropriate in meeting the requirements for the identification</p> <p>NOTE 3 – The terms Name and Version should be interpreted appropriately to correspond with a supplier’s terminology (e.g., Type, Series, Model)</p>	

1 Annex A.1.6.2 Protocol summary

Identification of protocol specification	IEEE Std 1914.3-2017, Radio over Ethernet
Identification of amendments and corrigenda to the PICS proforma that have been completed as part of the PICS	Amd: _____ Cor: _____ Amd: _____ Cor: _____
Have any exceptions been noted? (See Annex A.1.4. The answer, “Yes” means that the implementation does not conform to IEEE Std 1914.3.)	Yes [] No []

2 Annex A.2 PICS proforma for Radio over Ethernet

3 Annex A.2.1 Major capabilities/options

Item	Description	Sub-clause	Value/Comment	Status	Support
ET	RoE packets use EtherType XX-XX ⁴	3.4		M	[] Yes
MA1	Structure-agnostic RoE mapper tunneling mode implemented?	7.1	At least one of MA1 to MA5 shall be answered ‘Yes’.	O.1	[] Yes [] No

⁴ The value will be assigned at sponsor ballot time

MA2	Structure-agnostic RoE mapper line coding aware mode implemented?	7.1		O.1	[] Yes [] No
MA3	Structure-aware CPRI mode implemented?	7.2		O.1	[] Yes [] No
MA4	Native RoE packet with time-domain I/Q implemented?	7.3		O.1	[] Yes [] No
MA5	Native RoE packet with frequency-domain I/Q implemented?	7.4		O.1	[] Yes [] No
MA6	Structure-agnostic line coding aware mode supports un/redoing scrambling?	7.1		MA2:M	[] Yes [] N/A
NFL	Number of supported RoE flows?	4.4, 4.4.2	Although flowID space is 8 bits an implementation may support less than 255 concurrent RoE flows.	M	_____

1 Annex A.2.2 RoE packets and subTypes

2 Annex A.2.2.1 Data packets

Item	Description	Sub-clause	Value/Comment	Status	Support
DSA	Structure-agnostic data packet type implemented?	4.4.1	subtype 0000 0010b	MA1+MA2:M	[] Yes
DSW	Structure-aware data packet type implemented?	4.4.1	subType 0000 0011b	MA3:M	[] Yes
DCM	CPRI Slow C&M data packet type implemented?	4.4.1	subType 0000 0100b	MA3:M	[] Yes
DNT	Native packet type for time-domain samples implemented?	4.4.1	subType 0001 0000b	MA4:M	[] Yes
DNF	Native packet type for frequency-domain	4.4.1	subType 0001 0001b	MA5:M	[] Yes

	samples implemented?				
DNP	Native RoE PRACH packet type implemented?	4.4.1	subType 0001 0010b	MA5:O	[] Yes [] No
DFC	CPRI Fast C&M data packet type implemented?	7.2.6		MA3:M	[] Yes

1 Annex A.2.2.2 Control packets

Item	Description	Sub-clause	Value/Comment	Status	Support
CACK	RoE Control packet acknowledgement	8	Specify how control packets are acknowledged	M	_____
CP	RoE Control Packets subType implemented?	4.4.1, 8	subtype 0000 0000b is reserved for control packets	M	[] Yes
COAM	Control packet for RoE OAM TLV transport implemented?	8.1.3	opCode 0000 0000b	M	[] Yes
CAXC	Control packet for Ctrl_AxC control word transport implemented?	8.1.4	opCode 0000 0001b	MA3 * CP:M	[] Yes [] N/A
CVSD	Control packet for VSD control word transport implemented?	8.1.5	opCode 0000 0010b	MA3 * CP:M	[] Yes [] N/A
CTC	Is Timing Control packet supported?	8.1.6	opCode 0000 0011b	CP:O	[] Yes [] No [] N/A

2

1 Annex A.2.2.3 Other subTypes

Item	Description	Sub-clause	Value/Comment	Status	Support
EXP	RoE experimental packet subtype support.	4.4.1	Explain how packets with experimental subTypes are treated.	M	_____
RES	RoE reserved packet subtype support.	4.4.1	Explain how packets with reserved subTypes are treated.	M	_____

2 Annex A.2.3 RoE header

3 Annex A.2.3.1 flowID

Item	Description	Sub-clause	Value/Comment	Status	Support
NIL	Reserved NIL flowID supported and not interpreted as any flow?	4.4.2	flowID 1111 1111b	M	[] Yes

4 Annex A.2.3.2 orderInfo

Item	Description	Sub-clause	Value/Comment	Status	Support
SN	Sequence number mode implemented.	4.4.4.1	Either SN or TS shall be supported	O.1	[] Yes [] No
TS	Timestamp mode implemented.	4.4.4.2		O.1	[] Yes [] No
NS	Timestamp supports fractional nanosecond accuracy?	4.4.4.2		TS:O	[] Yes [] No [] N/A
LTS	Timestamps support detecting past timestamps?	6.2.5, 6.2.6		TS:M	[] Yes [] N/A

5 Annex A.2.4 Configuration parameters and options

6 Annex A.2.4.1 Ethernet link parameters

Item	Description	Sub-clause	Value/Comment	Status	Support
------	-------------	------------	---------------	--------	---------

EID	Maximum number of supported Ethernet links.	4.2, 6.2.5, 6.2.6, 6.2.7	Minimum 1, maximum 65536.	M	_____
-----	---	-----------------------------------	------------------------------	---	-------

1 Annex A.2.4.2 CPRI port parameters

Item	Description	Sub-clause	Value/Comment	Status	Support
CPP	CPRI port parameters are implemented and supported?	4.2, 6.2.5, 6.2.6, 6.2.7		MA3:M	[] Yes [] N/A
CID	Maximum number of supported CPRI ports.	4.2, 6.2.5, 6.2.6, 6.2.7	Minimum 1, maximum 65536.	MA3:M	_____
CPS	Supported CPRI port rates.	4.2, 6.2.5, 6.2.6, 6.2.7	List all supported CPRI port rates in Mb/s.	MA3:M	_____

2 Annex A.2.4.3 Mapper and RoE de-mapper parameters

Item	Description	Sub-clause	Value/Comment	Status	Support
MAP	(De)mapper parameters are implemented and supported?	6.2.5, 6.2.6, 6.2.7		M	[] Yes
CMPR	Compression is supported?	6.2.5, 6.2.6, 6.2.7	No support means only value 0x0 is possible.	MAP:O	[] Yes [] No
ENCR	Encryption is supported?	6.2.5, 6.2.6, 6.2.7	No support means only value 0x0 is possible.	MAP:O	[] Yes [] No
ATW1	Accept time window size.	6.2.5, 6.2.6, 6.2.7	The acceptTimeWindow minimum value.	LTS:M	[] N/A _____
ATW2	Accept time window size.	6.2.5, 6.2.6, 6.2.7	The acceptTimeWindow maximum value.	LTS:M	_____

LPAC	Maximum supported lenPack value.	6.2.5, 6.2.6, 6.2.7	Minimum 1, maximum 65536. The lenPack implicitly defines the maximum supported packet size.	MAP:M	_____
------	----------------------------------	---------------------	---	-------	-------

1 Annex A.2.5 Timing and Synchronization

Item	Description	Sub-clause	Value/Comment	Status	Support
ETS	External time source support available?	5.1		O	[] 1PPS [] 10MHz Other _____
JB	Jitter buffer maximum size.	5.2, 5.3	List (de)jitter buffer maximum sizes measured in units of microseconds.	M	_____

2 Annex A.2.6 Native RoE packet options

Item	Description	Sub-clause	Value/Comment	Status	Support
NRST	Supported sample widths in time domain.	7.3.1.5	List all supported sample widths in bits.	MA4:M	_____
NRSF	Supported sample widths in frequency domain.	7.4.2.5	List all supported sample widths in bits.	MA5:M	_____
NRFF	FFT support	7.4.1	Is FFT functionality supported?	MA5:O	[] Yes
NRNF	Number of FFT functions	7.4.1, 6.2.6, 7.2.7	Minimum 1, maximum NFL	NRFF:M	_____
NRPF	PRACH support	7.4.1	Is PRACH functionality supported?	MA5:O	[] Yes [] No

NRNP	Number of PRACH functions	7.4.4, 6.2.6	Minimum 1, maximum NRNF	NRPF:M	_____
------	---------------------------	--------------	-------------------------	--------	-------

1 **Annex A.2.6.1 RoE container definitions**

2 **Annex A.2.6.2 Container definition for structure-aware CPRI RoE mapper**

Item	Description	Sub-clause	Value/Comment	Status	Support
CNM1	Maximum number of containers.	7.2, 6.2.6, 6.2.7	Sub-clauses 6.8 and 6.9 reserve 5 bits for the maximum number, but the system may have its own internal maximum.	MA3:M	_____
CSKP	Maximum supported lenSkip value.	7.2.1		MA3:M	_____
CLEN	Maximum supported lenContainer value.	7.2.1		MA3:M	_____
CMOD	Modulo logic is supported.	7.2.1		MA3:M	[] Yes
CMLN	Maximum supported modulo value	7.2.1		CMOD:M	_____

3 **Annex A.2.6.3 Container definition for CPRI control plane**

Item	Description	Sub-clause	Value/Comment	Status	Support
CNM2	Number of containers is 8.	7.2.8		MA3:M	[] Yes
CFTM	All filter modes are supported.	7.2.8		MA3:M	[] Yes
CMOF	Maximum supported offset value.	7.2.8	Minimum 0, maximum 4095.	COFS:M	_____

Annex B Structure-aware examples (Informative)

Figure B-1 shows an example of a CPRI basic frame description, and how RoE container definitions are used to describe a “Packed Position” AxC Container mapping and how to separate the control words from sample data containing containers. The example below is for **.mapperID=8** where the AxC container length is *y* bits and **.containerID** 1,2 & 3 is associated with **(de)mapperID** 5,6 & 4 respectively.

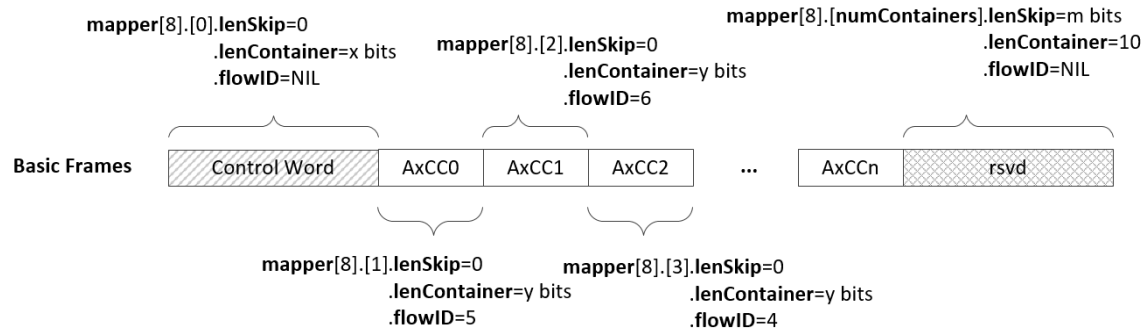


Figure B-1 – Packed position AxC container mapping in the I/Q data block

Figure B-2 shows an example of a CPRI basic frame, and how RoE container definitions are used to describe a “Flexible Position” AxC Container mapping. Note that an RoE container “data” length can be 0 while the reserved bits can be non-zero.

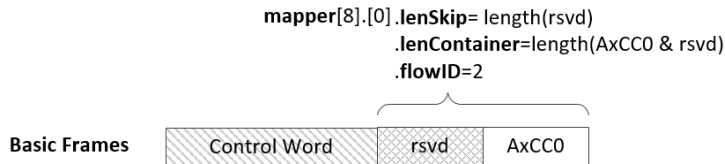


Figure B-2 – Flexible position AxC container mapping in the I/Q data block

Figure B-3 shows an example of using the RoE container definitions to describe a CPRI mapping method #3 (with an assumption there are no stuffing samples). Depending on the RoE container definitions each CPRI AxC Container can be treated as an individual RoE flow or grouped into one or more RoE flows.

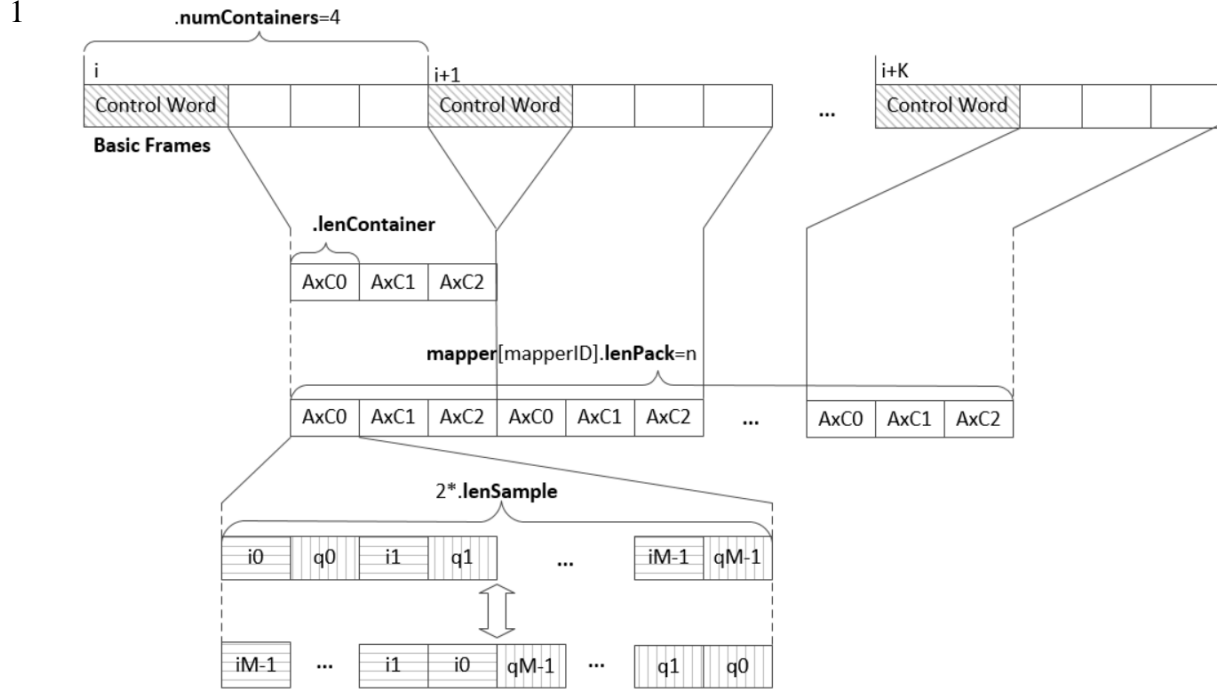


Figure B-3 - Example of CPRI mapping method #3

4 Figure B-4 shows an example of using a structure-aware RoE mapper with a modulo rule.
 5 The CPRI flow has AxCO and AxC1 at rate R and AxC2 and AxC3 at rate 0.5*R (i.e., half
 6 rate AxCs). Using the modulo rule for the “half rate” AxCs it is possible to construct CPRI
 7 basic frames where the “half rate” AxCs occupy every other CPRI basic frame. This allows
 8 transporting four AxC flows but only use three AxC Containers per CPRI basic frame.
 9 When transported over RoE there would be two RoE flows at rate R (for AxCO and AxC1)
 10 and another two RoE flows at rate 0.5*R (for AxC2 and AxC3). Total K*3 samples (or
 11 rather containers with sample data) are buffered/collected before
 12 assembling/disassembling an RoE packet.

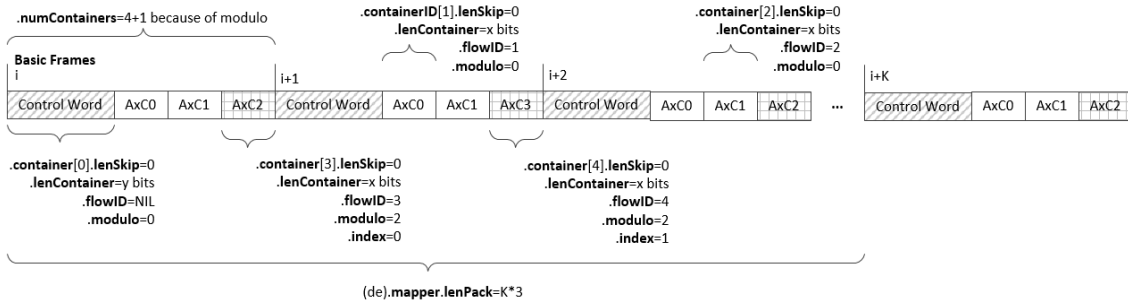


Figure B-4 - Example of using modulo rules and the structure-aware RoE mapper

17 This sub-clause defines a structure-aware CPRI to RoE mapper that looks into the CPRI
 18 frame and is able to further divide its content into different components. The RoE mapper

1 packetizes a number of CPRI Basic Frames worth of I/Q samples/AxC Containers for one
 2 AxC into an RoE data packet payload. The RoE mapper is equivalent to CPRI mapping
 3 method #3 without stuffing samples.

4 When enabled, the RoE mapper removes the line coding used by CPRI. The RoE mapper
 5 is aware of the start of the radio frame. In the context of this RoE mapper and CPRI
 6 specification the radio frame is the 10 ms frame number, which for E-UTRA-FDD would
 7 be aligned with BFN.

8 The RoE mapper buffers **.lenPack** worth of octets and then stores/extracts individual AxC
 9 containers to/from one or more RoE packets.

10 The below RoE configuration parameter example is for an RoE mapper with
 11 **mapperID=3**, **.ethID=4**, CPRI **.cpriID=5** with line rate option 3 (assuming 20MHz LTE
 12 and 2x2 MIMO) and CPRI mapping method 3 without any stuffing samples. The CPRI
 13 AxC Group size is 2 (i.e., $N_A=2$) and the AxC Container Block contains 16 samples (i.e.,
 14 $N_C=N_A*8$). The AxC Group AxC₀ has **.flowID=3** and the AxC₁ has **.flowID=2** i.e., there
 15 are two RoE data packet flows. The I/Q sample size is 15 bits per component. One RoE
 16 data packet contains 64 I/Q samples i.e., 8 BFs worth of samples.

17 The RoE configuration parameters are set as follows:

- 18 a) **mapper[2,3].destID=4** // both flows use the same Ethernet link
- 19 b) **mapper[2,3].srcID=5** // both flows use the same CPRI port
- 20 c) **mapper[2,3].lenPack=8** // 8 basic frames worth
- 21 d) **mapper[2,3].seqNumPMax=150*256/8**
- 22 e) **mapper[2,3].seqNumPInc=1**
- 23 f) **mapper[2,3].seqNumPIncProp=1**
- 24 g) **mapper[2,3].seqNumQMax=4095**
- 25 h) **mapper[2,3].seqNumQInc=1**
- 26 i) **mapper[2,3].seqNumQIncProp=1**
- 27 j) **mapper[2,3].numContainer=17** // 16 plus the control word
- 28 k) **mapper[2,3].[0].lenSkip=0**
- 29 l) **mapper[2,3].[0].lenContainer=32** // the control word
- 30 m) **mapper[2,3].[0].flowID=0xFF** // NIL flowID
- 31 n) **mapper[2,3].[0].modulo=0**

- 1 o) **mapper[2].[2,4,6,8,10,12,14,16].lenSkip=0**
- 2 p) **mapper[2].[2,4,6,8,10,12,14,16].lenContainer=30**
- 3 q) **mapper[2].[2,4,6,8,10,12,14,16].flowID=3** // Ax_{C₁}
- 4 r) **mapper[2].[2,4,6,8,10,12,14,16].modulo=0**
- 5 s) **mapper[3].[1,3,5,7,9,11,13,15].lenSkip=0**
- 6 t) **mapper[3].[1,3,5,7,9,11,13,15].lenContainer=30** // 15 I & 15 Q
- 7 u) **mapper[3].[1,3,5,7,9,11,13,15].flowID=2** // Ax_{C₀}
- 8 v) **mapper[3].[1,3,5,7,9,11,13,15].modulo=0**

1 **Annex C Presentation time usage**

2 This annex discusses how the presentation ToD counter is aligned to a chosen timescale's
 3 epoch, how the presentation timestamp is derived from the presentation ToD value at a
 4 mapper, and how the presentation timestamp is interpreted at a de-mapper to recover the
 5 correct presentation time.

6 **Aligning presentation time ToD to PTP timescale ToD**

7 An example of how a presentation time counter (PTIME_TOD) is aligned to a PTP
 8 timescale's ToD counter (PTP_TOD[95:0]) is given below.

9 The PTP timescale's ToD counter consists of:

```
10        sec = PTP_TOD[95:48]; // seconds
11        ns = PTP_TOD[47:16]; // nanoseconds, wrapping at 1 billion
12        frac_ns = PTP_TOD[15:11]; // only keep upper 5 bits of
13                                        // fractional nanoseconds
```

14 Aligning the PTIME_TOD counter is done as follows:

```
15        MAX_PT_NS = 16777216;
16        PTIME_TOD = (sec * 10^9 + ns) * 32 + frac_ns;
17        PTIME_TOD &= (MAX_PT_NS - 1); // keep only lower 29 bits
18                                        // of PTIME_TOD;
```

19 The result of this alignment is a PTIME_TOD counter that behaves like it started counting
 20 from zero at the same time the chosen timescale's ToD counter started counting from zero,
 21 at the chosen timescale's epoch.

22 **Conversion from presentation ToD to timestamp at the mapper**

23 At the mapper function, the desired presentation time at the far-end de-mapper is
 24 determined and then placed into the RoE header's timestamp field (see sub-clause 4.4.4.2).

25 In the following example, the mapper timestamps the arrival of radio data. This
 26 timestamped data is placed at the first payload location of the RoE packet into which the
 27 radio data is encapsulated. The RoE data was pre-determined to be "presented" by the de-
 28 mapper 8.5 ms after its arrival at the mapper. So, the mapper calculates the presentation
 29 time by adding 8.5 ms to its arrival ToD. The timestamp that is inserted into the RoE
 30 header is then created by truncating this sum. The least significant 24-bits of the
 31 nanoseconds field and the most significant 5-bits of the fractional nanoseconds field of the
 32 sum are used to create the 29-bit timestamp.

33 *Example 1a: at the mapper*

- 34 1. define MAX_PT_NS = 16,777,216 ns
- 35 2. radio data arrival time, as per PTIME_TOD[28:0], at mapper = 1,234,567.09375 ns
- 36 3. desired presentation time = arrival time + 8.5 ms
- 37 = modulo(9,734,567.09375, MAX_PT_NS) ns

- 1 $= 9,734,567.09375 \text{ ns}$
- 2 $= 0x9489A7 + 0x03/0x20 \text{ ns}$
- 3 4. $\text{timestamp}[28:0] = \text{presentation time}[28:0] = 0x1291\ 34E3$ in units of $1/32 \text{ ns}$

4 **Conversion from timestamp to presentation time at the de-mapper**

5 In this continuing example, the RoE packet arrives at the de-mapper 6.5 ms after it arrived
6 at the mapper.

7 *Example 1b: at the de-mapper, no rollover event*

- 8 1. define $\text{MAX_PT_NS} = 16,777,216 \text{ ns}$
- 9 2. radio data arrival time, as per $\text{PTIME_TOD}[28:0]$, at de-mapper
10 $= \text{modulo}(1,234,567.09375 + 6,500,000, \text{MAX_PT_NS}) \text{ ns}$
11 $= 7,734,567.09375 \text{ ns}$
12 $= 0x760527 \text{ ns} + 0x03/0x20 \text{ ns}$
13 $= 0x0EC0\ A4E3$ in units of $1/32 \text{ ns}$
- 14 3. Since $\text{presentation time} \geq \text{arrival time}$, this timestamped radio data arrived before or
15 at the presentation time so no rollover conditions need to be dealt with. The de-
16 mapper waits until its $\text{PTIME_TOD}[28:0]$ value reaches the presentation time,
17 $0x1291\ 34E3$, before it presents the radio data, starting with the timestamped data that
18 is located at the first payload byte of the packet, to the targeted recipient.

19 *Example 2: With 16,777,216 ns roll-over event*

- 20 1. radio data arrival time, as per $\text{PTIME_TOD}[28:0]$, at mapper = $9,000,000 \text{ ns}$
- 21 2. desired presentation time = arrival time + 8.5 ms
22 $= \text{modulo}(9,000,000 + 8,500,000, \text{MAX_PT_NS}) \text{ ns}$
23 $= 722,784.0 \text{ ns}$, a roll-over event occurred in PTIME_TOD
24 $= 0x0B0760 \text{ ns}$
- 25 3. **timeStamp** $[28:0] = \text{presentation time}[28:0] = 0x0160\ EC00$ in units of $1/32 \text{ ns}$
- 26 4. network delay is 4.5 ms
- 27 5. radio data arrival time, as per $\text{PTIME_TOD}[28:0]$, at de-mapper
28 $= \text{modulo}(9,000,000 + 4,500,000, \text{MAX_PT_NS})$
29 $= 13,500,000.0 \text{ ns}$
30 $= 0x00CD\ FE60 \text{ ns}$
31 $= 0x19BF\ CC00$ in units of $1/32 \text{ ns}$
- 32 6. Because $\text{presentation time} < \text{arrival time}$, a rollover event is detected. The de-mapper
33 must wait for $\text{PTIME_TOD}[28:0]$ counter to roll-over and then increment to equal the
34 presentation time, $0x0160\ EC00$, before the radio data is presented to the targeted
35 recipient.

36 **Error detection using .acceptTimeWindow**

37 The **.acceptTimeWindow** parameter, used with the presentation time and the arrival time
38 at the de-mapper, can be used to detect excess transit time problems. The following

1 pseudo-code shows how to use this parameter to determine if the timestamped radio data
 2 arrived at the de-mapper in the desired time window.

```

3     #define MAX_PT_NS    16777216
4     arrival_time_at_demapper &= 0x1FFFFFFF;
5     //mask off all but the lower 29 bits of the arrival time
6     if presentation_time < arrival_time_at_demapper
7     // compensate for rollover event
8         adj_presentation_time = presentation_time + MAX_PT_NS;
9     else // no rollover event
10        adj_presentation_time = presentation_time;
11    end if;
12    if ((adj_presentation_time >= arrival_time_at_demapper) AND
13        (adj_presentation_time - arrival_time_at_demapper
14        <= .acceptTimewindow)) then
15        on_time_arrival = TRUE;
16    else
17        on_time_arrival = FALSE;
18    end if;
```

19 Because of the rollover events at 16,777,216 ns, this comparison only works for packet
 20 transit times that are less than 16,777,216 ns. Delays beyond that value will not be properly
 21 detected.
 22

Annex D Sequence number example code

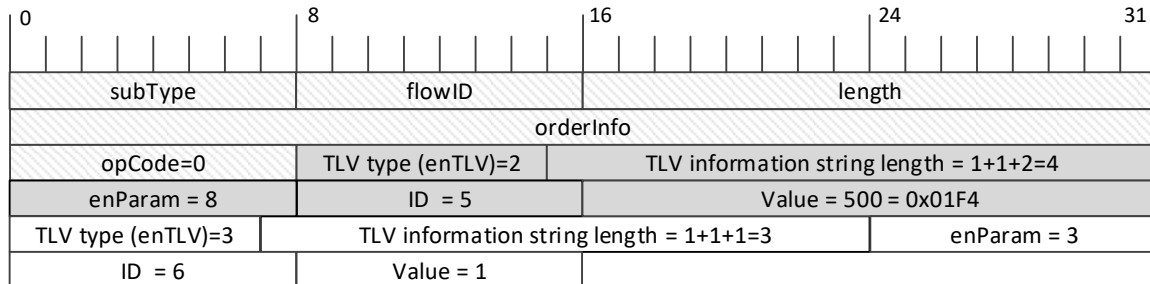
```

1  typedef struct {
2      uint32_t pval;
3      uint32_t qval;
4      uint32_t pmax;
5      uint32_t qmax;
6      uint32_t rsvd;
7      uint8_t q, p;
8  } seqnum_t;
9
10 // Initialize the sequence number
11 int initSeqNum( seqnum_t* sn,
12                uint32_t pmax, uint32_t qmax,
13                uint32_t pval, uint32_t qval,
14                uint32_t rsvd ) {
15
16     // sanity checks
17     if (pval >= pmax) return -1;
18     if (qmax == 0 && qval != 0) return -1;
19
20     // initialize sequence number..
21     sn->pmax = pmax;
22     sn->qmax = qmax;
23     sn->pval = pval;
24     sn->qval = qval;
25     sn->p = pmax > 0 ? 1 : 0;
26     sn->q = qmax > 0 ? 1 : 0;
27
28     // calculate q and p
29     while (pmax >= 1) sn->p++;
30     while (qmax >= 1) sn->q++;
31
32     // more sanity checks
33     if (sn->p == 0) return -1;
34     if (sn->p + sn->q > 32) return -1;
35
36     // remaining initialization
37     sn->rsvd = sn->q + sn->p >= 32 ? 0 : rsvd << sn->q + sn->p;
38     return 0;
39 }
40
41 // Construct 32-bit sequence number from counter fields
42 uint32_t getSeqNum( const seqnum_t* sn ) {
43     return sn->rsvd | sn->qval << sn->p | sn->pval;
44 }
45
46 // Increase sequence number counter fields(sender side operation)
47 uint32_t incSeqNum( seqnum_t* sn, uint32_t pinc, uint32_t qinc ) {
48     uint32_t oldpval = sn->pval;
49     sn->pval += pinc;
50     if (sn->pval > sn->pmax || sn->pval < oldpval) {
51         sn->pval = sn->pval - sn->pmax - 1;
52         sn->qval += qinc;
53     }
54     if (sn->qval > sn->qmax) {
55         sn->qval = sn->qval - sn->qmax - 1;
56     }
57     return getSeqNum( sn );
58 }
59

```

1 Annex E RoE OAM TLV Example

2 Figure E-1 shows an example of an RoE OAM TLV control packet with 2 concatenated
3 TLVs setting 2 parameters.



4
5 **Figure E-1 – RoE OAM TLV Example**

6 The first TLV in the payload sets RoE mapper number 5's packet length to 500, or
7 **mapper[5].lenPack=500**

- 8 a) **.enTLV** = 2. The RoE mapper (see Table 6-1)
- 9 b) **.enParam** = 8. The **.lenPack** parameter (see Table 6-5)
- 10 c) **.ID** = 5
- 11 d) **.value** = 500 = 0x01F4

12 The second TLV in the payload sets RoE mapper number 6's destination Ethernet port to
13 1, or **demapper[6].destID=1**

- 14 a) **.enTLV** = 3. The RoE de-mapper (see Table 6-1)
- 15 b) **.enParam** = 3. The **.destID** parameter (see Table 6-6)
- 16 c) **.ID** = 6
- 17 d) **.value** = 1 = 0x01

1 Bibliography (informative)

- 2 [B1] IEEE Std 802.1ABTM, IEEE Standard for Local and metropolitan area networks –
3 Station and Media Access Control Connectivity Discovery.
4 <http://www.ieee802.org/1/files/private/ab-drafts/d12/802-1AB-D12.pdf>
- 5 [B2] IEEE Std 802.1CMTM, IEEE Standard for Local and metropolitan area networks –
6 Time-Sensitive Networks for Fronthaul. <http://www.ieee802.org/1/pages/802.1cm.html>
- 7 [B3] ITU-T G.8275, Architecture and requirements for packet based time and phase
8 distribution. <https://www.itu.int/rec/T-REC-G.8275/en>
- 9 [B4] ITU-T G.8271/Y.1366, Time and phase synchronization aspects of packet networks.
10 <https://www.itu.int/rec/T-REC-G.8271/en>
- 11 [B5] IEEE Std 1588v2TM, IEEE Standard for a Precision Clock Synchronization Protocol.
12 <https://standards.ieee.org/findstds/standard/1588-2008.html>