|  |
| --- |
| 3GPP TS 26.522 V0.2.0 (2023-11) |
| Technical Specification |
| 3rd Generation Partnership Project;Technical Specification Group Services and System Aspects;5G Real-time Media Transport Protocol Configurations(Release 18) |
|   |
|  |  |
|  |
| The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP.The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented.This Specification is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification.Specifications and Reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices. |

|  |
| --- |
|  |
| ***3GPP***Postal address3GPP support office address650 Route des Lucioles - Sophia AntipolisValbonne - FRANCETel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16Internethttps://www.3gpp.org |
| ***Copyright Notification***No part may be reproduced except as authorized by written permission.The copyright and the foregoing restriction extend to reproduction in all media.© 2023, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).All rights reserved.UMTS™ is a Trade Mark of ETSI registered for the benefit of its members3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational PartnersLTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational PartnersGSM® and the GSM logo are registered and owned by the GSM Association |

Contents

Foreword 4

Introduction 5

1 Scope 6

2 References 6

3 Definitions of terms, symbols and abbreviations 7

3.1 Terms 7

3.2 Symbols 7

3.3 Abbreviations 7

4 RTP Functionalities 7

4.1 Multiple Simultaneous RTP Streams in an RTP Session 7

4.2 Multiple RTP Sessions 7

4.3 RTP Retransmission 7

4.4 RTP Header Extensions 7

4.4.1 General 7

4.4.2 RTP Header Extension for PDU Set Marking 8

4.4.2.1 General 8

4.4.2.2 One-byte RTP Header Extension Format 8

4.4.2.3 Two-byte RTP Header Extension Format 8

4.4.2.4 Semantics 8

4.4.2.5 SDP Signaling 9

4.4.2.6 Guidelines for PDU Set Marking 10

4.4.2.6.1 End of Data Burst Field 10

4.4.2.6.2 PDU Set Importance Field 10

4.4.2.6.3 PDU Set Size Field 13

4.4.2.7 Guidelines for AS 15

4.5 RTP Forward Error Correction 15

4.6 SRTP 15

5 RTCP Feedback Reporting Procedures 15

5.1 General 15

Annex A (informative): Guidelines for PDU Set identification without PDU Set RTP HE 15

A.0 General 15

A.1 RTP/SRTP header 16

A.2 RTP payload 16

A.2.1 General 16

A.2.2 RTP payload for H.264/AVC codec 16

A.2.3 RTP payload for H.265/HEVC codec 17

Annex B (informative): Change history 19

# Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

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

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

In the present document, modal verbs have the following meanings:

**shall** indicates a mandatory requirement to do something

**shall not** indicates an interdiction (prohibition) to do something

The constructions "shall" and "shall not" are confined to the context of normative provisions, and do not appear in Technical Reports.

The constructions "must" and "must not" are not used as substitutes for "shall" and "shall not". Their use is avoided insofar as possible, and they are not used in a normative context except in a direct citation from an external, referenced, non-3GPP document, or so as to maintain continuity of style when extending or modifying the provisions of such a referenced document.

**should** indicates a recommendation to do something

**should not** indicates a recommendation not to do something

**may** indicates permission to do something

**need not** indicates permission not to do something

The construction "may not" is ambiguous and is not used in normative elements. The unambiguous constructions "might not" or "shall not" are used instead, depending upon the meaning intended.

**can** indicates that something is possible

**cannot** indicates that something is impossible

The constructions "can" and "cannot" are not substitutes for "may" and "need not".

**will** indicates that something is certain or expected to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**will not** indicates that something is certain or expected not to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**might** indicates a likelihood that something will happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

**might not** indicates a likelihood that something will not happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

In addition:

**is** (or any other verb in the indicative mood) indicates a statement of fact

**is not** (or any other negative verb in the indicative mood) indicates a statement of fact

The constructions "is" and "is not" do not indicate requirements.

# Introduction

TR 26.998 (5G Glass-type AR/MR) identified multiple aspects of normative work to support “5G/AR Real-time Communication” (clause 8.4). TR 26.998 identified normative work needed to support delivery of immersive media via RTP for IMS-based and WebRTC-based conversational services. To support XR split rendering as described in clause 8.6 of TR 26.998, RTP is also needed to transport immersive media and metadata information between the edge and device.

To improve support for the above XR services and enablers, it is necessary to configure RTP with specific settings and features that enable immersive experiences. Further improvements in performance and QoE over the 5G system can be achieved by specifying RTP configurations that are integrated and optimized for the 5G system, and leverage cross-layer optimizations used by other 3GPP specifications.

As these RTP configurations will be specified for use by multiple services, service enablers, and potentially, application developers, it is very important that they do not introduce unnecessary complexities that would discourage commercial deployment of the configurations. Therefore, technologies specified here should be commercially relevant and not introduce implementation and interoperability complexity without clearly demonstrating performance gains or new relevant functionalities.

# 1 Scope

The present document focuses on RTP [4] over UDP [x], optimizing the use of RTP for the uni-directional and bi-directional transport of real-time immersive media and associated metadata.

The following services and enablers are explicitly considered by this specification (in no specific order):

- IMS-based conversational XR services [x]

- WebRTC-based conversational XR services [x]

- WebRTC-based conversational services using traditional media [x]

- XR split-rendering, i.e., real-time transport of media between UE and network edge [x]

New transport protocols like QUIC [x] may be considered in a future update of this specification.

# 2 References

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

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

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

[2] ITU-T Rec H.264: "Advanced video coding specification".
<https://www.itu.int/rec/T-REC-H.264-202108-I/en>

[3] ITU-T Rec H.265: "High efficiency video coding specification ".
<https://www.itu.int/rec/T-REC-H.265-202108-I/en>

[4] IETF RFC 3550 (2003): "RTP: A Transport Protocol for Real-Time Applications", H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson.

[5] IETF RFC 6184 (2011): "RTP Payload Format for H.264 Video", Y.-K. Wang, R. Even, T. Kristensen, R. Jesup.

[6] IETF RFC 7798 (2016): "RTP Payload Format for High Efficiency Video Coding (HEVC)", Y.-K. Wang, Y. Sanchez, T. Schierl, S. Wenger, M. M. Hannuksela.

[7] Chromium WebRTC Implementation [https://source.chromium.org/chromium/chromium/src/+/main:third\_party/webrtc/media/base/media\_constants.cc;l=17?q=kVideoMtu&ss=chromium](https://source.chromium.org/chromium/chromium/src/%2B/main%3Athird_party/webrtc/media/base/media_constants.cc;l=17?q=kVideoMtu&ss=chromium)

[8] Linux Manual Page - Linux IPv4 protocol implementation <https://man7.org/linux/man-pages/man7/ip.7.html>

[9] Linux Manual Page - Linux IPv6 protocol implementation <https://man7.org/linux/man-pages/man7/ipv6.7.html>

[10] IETF RFC 5761 (2010): "Multiplexing RTP Data and Control Packets on a Single Port", C. Perkins, M. Westerlund

[11] IETF RFC 8285 (2017): "A General Mechanism for RTP Header Extensions", D. Singer, H. Desineni, R. Even

[12] RTP Header Extension for Absolute Sender Time
[https://webrtc.googlesource.com/src/+/refs/heads/main/docs/native-code/rtp-hdrext/abs-send-time](https://webrtc.googlesource.com/src/%2B/refs/heads/main/docs/native-code/rtp-hdrext/abs-send-time)
[retrieved on Nov 14, 2023]

[13] IETF RFC 5905 (2010): "Network Time Protocol Version 4: Protocol and Algorithms Specification”, D. Mills, J. Martin, J. Burbank, W. Kasch

[14] IEEE 1588-2019 – IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, June 2020

[15] IETF RFC 4574 (2006): "The Session Description Protocol (SDP) Label Attribute", O. Levin, G. Camarillo

[16] IETF RFC 3611 (2003): "RTP Control Protocol Extended Reports (RTCP XR)", T. Friedman, R. Caceres, A. Clark

[17] 3GPP TS 26.119: "Media Capabilities for Augmented Reality"

# 3 Definitions of terms, symbols and abbreviations

## 3.1 Terms

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

**Data Burst:** A data burst is a set of multiple PDUs generated and sent by the application such that there is an idle period between two data bursts. A Data Burst can be composed of one or multiple PDU Sets.

**PDU Set:** One or more PDUs carrying the payload of one unit of information generated at the application level (e.g. frame(s), video slice(s), metadata, etc.).

**XR Pose:** A position and orientation in space relative to an XR Space.

**XR Space:** A frame of reference in which an application chooses to track the real world. An XR Space provides a relation of the user’s physical environment with other tracked entities.

## 3.2 Symbols

For the purposes of the present document, the following symbols apply:

Symbol format (EW)

<symbol> <Explanation>

## 3.3 Abbreviations

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

Abbreviation format (EW)

<ABBREVIATION> <Expansion>

# 4 RTP Functionalities

## 4.1 Multiple Simultaneous RTP Streams in an RTP Session

TBA

## 4.2 Multiple RTP Sessions

TBA

## 4.3 RTP Retransmission

TBA

## 4.4 RTP Header Extensions

### 4.4.1 General

TBA

### 4.4.2 RTP Header Extension for PDU Set Marking

#### 4.4.2.1 General

The RTP Header Extension for PDU Set marking can be used by an AS (e.g., MRF) or a sender UE that sends media to a receiver UE over RTP.

The RTP Header Extension for PDU Set marking shall support both RTP Header Extension formats (i.e., the one-byte and the two-byte formats) according to RFC 8285.

If the RTP Header Extension for PDU Set marking is the only RTP header extension used, the endpoints shall use the 1-byte header format for maximum savings. If other 2-byte RTP header extension elements are used, then the 2-byte header may be used.

NOTE: The headers are not shown with padding as this depends on other prospective extension elements in use, as per RFC 8285 alignment specifications.

#### 4.4.2.2 One-byte RTP Header Extension Format

The one-byte RTP Header Extension for the marking of PDU Sets and End of Bursts is defined as follows:

 0 1 2 3

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

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | 0xBE | 0xDE | length |

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | ID | len |E| R |D| PSI | PSSN | PSN |

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | PSSize | NPDS

 +.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+

 |

 +.+.+.+.+.+.+.+.+

#### 4.4.2.3 Two-byte RTP Header Extension Format

The two-byte RTP Header Extension for the marking of PDU Sets and End of Bursts is defined as follows:

 0 1 2 3

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

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | 0x100 |appbits| length |

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | ID | len |E| R |D| PSI | PSSN

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | PSN | PSSize |

 +-+-+-+-+-+-+-+-+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+

 | NPDS |

 +.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+

#### 4.4.2.4 Semantics

The semantics of the fields of the RTP Header Extension for the marking of PDU Set and End of Bursts are defined as follows:

- **End PDU of the PDU Set [E] (1 bit):** This field is a flag that shall be set to 1 for the last PDU of the PDU Set and set to 0 for all other PDUs of the PDU Set.

- **End of Data Burst [D] (1 bit):** This field is 1 bit in length and indicates the end of a Data Burst. The bit encodes the End of Data Burst indication as per the guidelines provided in Clause 4.4.2.6.1.

- Reserved (2 bits): This field is reserved for future usage (e.g., dynamic burst indication). It shall be set to 0 by the RTP sender and shall be ignored.

- **PDU Set Importance [PSI] (4 bits):** The PDU Set Importance field indicates the importance of this PDU Set compared to other PDU Sets within the same QoS flow. Lower values shall indicate a higher importance PDU Set with the highest importance PDU Set indicated by 0 and the lowest importance PDU Set indicated by 15.

NOTE 1: A complete set of guidelines for setting the PSI field for various audio/video codecs are provided in Clause 4.4.2.6.2

Editor’s Note: AS/UE are unaware of QoS flows and so the above text needs to be revised to remove the term. The PSI value needs to be set considering one or more RTP streams (depending on multiplexing and other aspects that are under discussion in reference to the guidelines). Once the guidelines are finalized, the text above will be revised possibly with the introduction of a new term in place for QoS flow that consists of one or more RTP streams.

- **PDU Set Sequence Number [PSSN] (10 bits):** The field encodes the sequence number of the PDU Set to which the current PDU belongs acting as a 10-bit numerical identifier for the PDU Set.

NOTE 2: This value wraps around at 1023, however, using the RTP packet sequence number and PSSN pair a receiver may uniquely distinguish between any PDU Sets.

- **PDU Sequence Number within a PDU Set [PSN] (6 bits):** The sequence number of the current PDU within the PDU Set. The PSN shall be set to 0 for the first PDU in the PDU Set and incremented monotonically for every PDU in the PDU set in order of transmission from the sender.

NOTE 3: A receiver may use the RTP packet sequence number together with the PSN to distinguish between PDUs within a PDU Set that contains more than 64 PDUs.

- **PDU Set Size [PSSize] (24 bits):** The PDU Set Size indicates the total size of all PDUs of the PDU Set to which this PDU belongs. This field is optional and subject to an SDP signaling offer/answer negotiation, where the Application Server may indicate whether it will be able to provide the size of the PDU Set for that RTP stream. If not enabled, the field should not be present. If enabled, but the Application Server is not able to determine the PDU Set Size for a particular PDU Set, it should set the value to 0 in all PDUs of that PDU Set. The PSSize shall indicate the size of a PDU Set including RTP/UDP/IP header encapsulation overhead of its corresponding PDUs. The PSSize is expressed in bytes. It is recommended to add the PDU Set Size field when the Number of PDUs in the PDU Set field is present.

- **Number of PDUs in the PDU Set [****NPDS] (16 bits):** The number of PDUs within the PDU Set indicates the total number of PDUs belonging to the same PDU Set. This field is optional and subject to an SDP signaling offer/answer negotiation, where the Application Server may indicate whether it will be able to provide the number of PDUs within the PDU Set for that RTP stream. It is recommended to add the Number of PDUs in the PDU Set field when the PDU Set Size field is present.

NOTE 4: This field may be optionally present given the signaling of the “pdu-set-size” extension attribute in the SDP offer/answer negotiation as per Clause 4.4.2.5.

NOTE 5: Guidelines to set the PDU Set Size in bytes by an Application Server are provided in Clause 4.4.2.6.3.

NOTE 6: When the receiver is aware about the used IP version at the sender, IP version changes in the path (e.g. due to a NAT64) can be handled by the receiver. When the receiver detects an IP version change, the receiver should correct the PDU Set Size value before further processing. The receiver can compute the correct PDU Set size by adding or subtracting the difference between IPv6 and IPv4 header size multiplied by the number of PDUs in the PDU Set.

#### 4.4.2.5 SDP Signaling

An AS or sender UE capable of sending PDU set marking HE shall use the SDP attribute extmap for PDU set marking HE in the media description of the RTP stream(s) carrying the PDU set HE. A receiver that does not support PDU set marking HE can ignore the RTP header when included. The signaling of the PDU Set and End-of-Burst marking RTP header extension shall follow the SDP signaling design and the syntax and semantics of the "extmap" attribute as outlined in RFC8285.The URN for the PDU Set marking shall be set to "**urn:3gpp:pdu-set-marking:rel-18**".

The header extension identifier shall be registered with IANA as maintained in [Real-Time Transport Protocol (RTP) Parameters (iana.org)](https://www.iana.org/assignments/rtp-parameters/rtp-parameters.xhtml#rtp-parameters-10).

The ABNF syntax for the extmap attribute for the signaling of PDU Set Information and End of Burst marking is defined as follows:

*extmap-attr="a=extmap:" 1\*5DIGIT ["/" direction] SP "urn:3gpp:pdu-set-marking:rel-18" SP extensionattributes*

*extensionattributes = \*3(format / "pdu-set-size")*

*format = "short" / "long"*

The extension attributes have the following semantics:

- format: indicates if the RTP header extension for PDU Set and End-of-Burst marking uses the 1-byte (short) or the 2-byte (long) format.

- pdu-set-size: if present, this attribute indicates that the application server will provide the PDU Set size in bytes in the RTP header extension with every RTP packet. This results in an additional 3 bytes of length for the RTP header extensi

#### 4.4.2.6 Guidelines for PDU Set Marking

##### 4.4.2.6.1 End of Data Burst Field

NOTE: These detailed guidelines are FFS.

##### 4.4.2.6.2 PDU Set Importance Field

NOTE: The following aspects need to be further defined:

- Default value for importance when the sender cannot define importance

- Codec level aspect:

- video: importance when PDU set is i) slice, ii) frame iii) parameter sets iv) tile set v) other?

- audio: when and if to use PDU set marking HE in an audio frame.

- text/metadata: when and if to use PDU set marking HE in text/metadata

- image: a frame is a PDU set and the importance for all frames are i) same ii) set based on application aspects.

- Importance across bitstreams

- Multiplexed streams: importance marking when a 5-tuple corresponds to more than one bitstream

- Importance marking considerations for non-multiplexed bitstreams

4.4.2.6.2.1 General

PDU Set Importance (PSI) field can be used by the RAN to discard PDU sets. PDU sets with higher PSI value are more likely to be discarded.

The PDU sets that contain audio data should be set with highest importance compared with other media PDU sets.

NOTE 1: PDU sets that carry immersive audio data are not set with highest importance compared with other media PDU sets. The importance value of immersive audio PDU sets is FFS.

The PDU sets that contains the reference frames present in the video bitstream should be set with higher importance compared with PDU sets that contain non-reference frames.

NOTE 2: It is assumed that the video bitstream uses referencing structures that have no coding delay caused by out-of-order output, as typically done for low-delay applications.

The following clauses provides the guidelines on setting the PSI field in a PDU set RTP header extension for the 3GPP video codecs. For specific PSI value ranges, refer to clause 4.4.2.6.2.5.

4.4.2.6.2.2 H.264 Codec

In an H.264 bitstream, NAL units with the nal\_unit\_type field assigned the value 5 (refer to Table 7.1 in AVC specification [2]) are Instantaneous Decoding Refresh (IDR) pictures. When the Type field value in the NAL Unit header of an RTP packet is 5, then the corresponding PDUs in that PDU set should be set with higher importance.

The parameter set NAL units such as Sequence Parameter Set (SPS) and Picture Parameter Set (PPS) are important for decoding the bitstream. Therefore, PDU sets with a Type field value equal to 7, 8, 13 or 15 (refer to Table 7.1 in AVC specification [2]) in the NAL Unit header of the RTP packet should be set with higher importance.

+---------------+

|0|1|2|3|4|5|6|7|

+-+-+-+-+-+-+-+-+

|F|NRI| Type |

+---------------+

Figure 4.4.2.6-1: NAL unit type octet in an RTP packet payload

The NAL unit type octet contains the NRI (nal\_ref\_idc) field highlighted in Figure 4.4.2.6-1. The NRI field indicate the relative transport priority. A value of b00 indicates that the content of the NAL unit is not used to reconstruct reference pictures for inter picture prediction. Such NAL units can be discarded without risking the integrity of the reference pictures. Values greater than b00 indicate that the decoding of the NAL unit is required to maintain the integrity of the reference pictures. The highest transport priority is b11, followed by b10, and then by b01; finally, b00 is the lowest. PDU sets with an NRI value b00 should be set with lower importance relative to the PDU sets with other NRI values. PDU sets with an NRI value b11 should be set with higher importance relative to the PDU sets with other NRI field values.

The Type and NRI fields can be used to set the PDU Set importance. The PDU set importance value assignment based on the Type and NRI field values is for further study.

4.4.2.6.2.3 HEVC Codec

Different from H.264 (AVC), H.265 (HEVC) NAL unit header (shown in Figure 4.4.2.6-2) is two bytes, contains a 6-bit Type field, a 5-bit LayerID field, a 3-bit TID field, and no NRI field. The Type and TID field in the NAL unit header indicate the relative transport priority. They can be used to set the PDU Set importance.

NAL unit types 0–31 indicate Video Coding Layer (VCL) NAL unit types; 32–40 indicate non-VCL NAL unit types. NAL unit types 41–47 are reserved, and types 48–63 are unspecified.

+---------------+---------------+

|0|1|2|3|4|5|6|7|0|1|2|3|4|5|6|7|

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

|F| Type | LayerId | TID |

+-------------+-----------------+

Figure 4.4.2.6-2: The Structure of the HEVC NAL Unit Header

All VCL NAL units of the same access unit must have the same NAL unit type, which defines the type of the access unit and its coded picture. There are three basic classes of pictures in H.265 (HEVC): intra random access point (IRAP) pictures, leading pictures, and trailing pictures.

In an HEVC bitstream, NAL units with the nal\_unit\_type field assigned a value in the range 16 to 23 (inclusive) (refer to Table 7.1 in HEVC specification [3]) are Intra Random Access Pictures (IRAP) pictures. This includes IDR, CRA, and BLA picture types as well as types 22 and 23, which currently are reserved for future use. When the Type field value in the NAL Unit header of RTP packet is in the range 16 to 23 (inclusive), then the corresponding PDUs in that PDU set should be set with higher importance.

The parameter set NAL units such as Sequence Parameter Set (SPS), Picture Parameter Set (PPS), Video Parameter Set (VPS) are important for decoding the bitstream. Therefore, PDU sets with payload Type field value in the NAL Unit header of RTP packet in the range 32 to 34 (inclusive) should be set with higher importance.

RFC 7798 [6] specifies Aggregation Packets (APs) to enable the reduction of packetization overhead for small NAL units, such as most of the non-VCL NAL units, which are often only a few octets in size. An AP aggregates NAL units within one access unit. Each NAL unit to be carried in an AP is encapsulated in an aggregation unit. An AP consists of a payload header (denoted as PayloadHdr) followed by two or more aggregation units. In an AP, the Type field in the PayloadHdr MUST be equal to 48. APs are typically used to aggregate parameters sets (VPS, SPS, PPS) into a single packet. When APs are used, the sender should consider the NAL unit types of the aggregation units while assigning the PSI value. For example, if the aggregation unit contains parameter sets, PDU sets containing those should be set with higher importance.

It could be that there are PDUs with different NAL unit types in a PDU set. For example, if the first PDU in PDU set is a prefix SEI message or Access Unit Delimiter (AUD), it would be misleading if the sender looked only at the first PDU of the PDU set to determine the PSI value. The sender should ignore the NAL units with non-VCL NAL unit types 35 and 39 and instead consider NAL unit types of the subsequent VCL NAL units while determining the PSI value for such PDU sets.

A leading picture is a picture that follows a particular IRAP picture in decoding order and precedes it in output order. There are two types of leading pictures in H.265 (HEVC): Random access decodable leading (RADL) pictures and Random access skipped leading (RASL) pictures. A RADL picture is a leading picture that is guaranteed to be decodable when random access is performed at the associated IRAP picture. Therefore, RADL pictures are only allowed to reference the associated IRAP picture and other RADL pictures of the same IRAP picture. A RASL picture is a leading picture that may not be decodable when random access is performed from the associated IRAP picture. Only other RASL pictures are allowed to be dependent on a RASL picture. Hence, in HEVC bitstreams, RASL pictures can be discarded during random access. HEVC provides mechanisms to enable specifying the conformance of a bitstream wherein the originally present RASL pictures have been discarded. Consequently, system components can discard RASL pictures, when needed, without worrying about causing the bitstream to become non-compliant.

PDU sets with Type field value equal to 6 or 7 (refer to Table 7.1 in HEVC specification [3]) in the NAL Unit header of RTP packet are RADL pictures. PDU sets with Type field value equal to 8 or 9 (refer to Table 7.1 in HEVC specification [3]) in the NAL Unit header of RTP packet are RASL pictures. PDU sets that contain RADL pictures should be set with lower importance relative to the IRAP pictures and higher importance relative to the RASL pictures in the bitstream.

 In video coding, temporal scalability is the option to decode only some of the frames in a video stream instead of the whole stream. This enables a media server to reduce the bitrate sent towards viewers that don’t have enough bitrate or CPU to handle the whole stream. In HEVC, pictures with lowest temporal identifier value (TID) are used as reference pictures in the bitstream and are important for decoding the dependent frames. PDU sets with TID value 1 (lowest possible value) should be set with higher importance relative to PDU sets that have a higher TID value. The PSI value for such pictures should be lower for IRAP pictures and slightly higher for non-IRAP pictures compared to the pictures with higher TID values. Pictures with highest TID value cannot be used as reference pictures and can be discarded at the network level when the throughput is not good, or network conditions are unstable. PDU sets with higher TID values should be set with lower importance compared with the PDU sets with lower TID values.

In H.265 (HEVC), each leading picture and trailing picture type has two type values. The even picture type numbers indicate sub-layer non-reference pictures and odd picture type numbers indicate sub-layer reference pictures. An encoder can use the sub-layer non-reference picture types for pictures that are not used for reference for prediction of any picture in the same temporal sub-layer. Note that a sub-layer non-reference picture may still be used as a reference picture for prediction of a picture in a higher temporal sub-layer.

PDU sets that contain sub-layer reference picture types should be assigned a lower PDU set importance value compared to the PDU sets with the corresponding sub-layer non-reference picture types.

4.4.2.6.2.4 PDU set importance based on affected PDU sets

When the transport layer is forced to perform immediate dropping/discarding of a PDU set but has a freedom of selection among the PDU sets, the PDU set with smaller degrees of artifact would be the better choice in most cases. Dropping of a PDU set may corrupt the decoded output of itself and the other PDU sets though they may already be transmitted perfectly to the receiving end or yet in a queue waiting to be transmitted. The degrees of artifact can be explicitly transferred as the number of affected frames which precedes/follows the PDU set, or can be implicitly transferred as the importance value where the lower value means the higher PDU sets are affected while higher values proportionally mean less number of PDU sets are affected for example. By considering such a quantization of various affected PDU sets can be translated into importance field, using 4 bits to represent 16 possible size ranges is recommended.

The information on the size of propagation error which caused by the dropping of each PDU set may be provided by the application layer. The information may present the size of error propagation implicitly with a proportional mapping of error propagation size to an index such as the importance of the PDU set in the media stream.

The importance value of a PDU Set in PDU set information RTP HE is set as follows:

- The error propagation size is mapped to importance field value. The higher the error propagation size of a PDU set, that PDU set is more important, and it shall be assigned with the lower PDU set importance value. PDU sets with low error propagation are of less importance and the PDU set importance value for such PDU sets shall be higher compared to PDU sets with higher error propagation size.

4.4.2.6.2.5 PSI mapping based on PDU set dependencies

Senders should consider that multiplexed RTP streams are treated as a single QoS flow and set the PSI field accordingly, i.e., the PSI field for one bitstream will affect the PDU sets in other multiplexed streams as well. In some cases, dependencies may exist across bitstreams even when they are not multiplexed, particularly for XR services.

In case of such dependencies, it may not be enough to have PSI values based on codecs and media types alone. PSI values shall be set in this case based on the following, which are listed in an increasing order of importance, i.e., decreasing order of PSI values.

- The PDU set is considered not necessary for the processing of any other PDU set. Such PDU sets should be assigned the highest PSI values 14-15. When multiplexing, if a PDU set is assigned PSI value of 15, similar PDU sets of all streams should be given value 15 to prevent unfair treatment. If interdependency is known, e.g., in stereo streams (left eye is more important than right eye), then the more important stream can be given the PSI value 14.

- In H.264, these include the PDU sets with an NRI value equal to b00 in the NAL unit header.

- In HEVC, the NAL unit header does not contain a field like NRI that indicates the relative transport priority. Hence, it is up to the application to identify such PDU sets.

- The PDU set is necessary for the processing of some PDU sets of the stream to which it belongs. Such PDU sets should be assigned a PSI value in the range 9-13 (inclusive). The lower end of the range should be used for IDR/IRAP pictures since they are more important for the bitstream.

- In H.264, these include:

- IDR pictures with nal\_unit\_type equal to 5

- Non-IDR pictures with nal\_unit\_type in the range 1 to 4 (inclusive)

- In HEVC, these include:

- IRAP pictures with nal\_unit\_type field assigned a value in the range 16 to 23 (inclusive)

- RADL or RASL pictures with nal\_unit\_type in the range 6 to 9 (inclusive)

- The PDU set is necessary for the processing of all the other PDU sets of the stream to which it belongs. Such PDU sets should be assigned a PSI value in the range 6-8 (inclusive).

- In H.264, these include:

- SPS, PPS, i.e., NAL units with the nal\_unit\_type field equal to 7, 8, 13 or 15

- In HEVC, these include:

- SPS, PPS, VPS, i.e., NAL units with the nal\_unit\_type field in the range 32 to 34 (inclusive)

- The PDU set is necessary for the processing of some PDU sets of the stream to which it belongs and also necessary for the processing of some PDU sets of some other streams to which it does not belong. Such PDU sets should be assigned a PSI value in the range 4-5 (inclusive).

NOTE 1: Values in this and lower range shall be used for assigning PSI values to PDU sets in multiplexed streams or if dependencies exist across non-multiplexed bitstreams. Use cases for those cases are FFS. For single streams, the ranges provided by the previous bullet points shall be used.

NOTE 2: Considerations for multiplexed audio streams are FFS.

- The PDU set is necessary for the processing of all PDU sets of the stream to which it belongs and also of some other streams to which it does not belong. Such PDU sets should be assigned a PSI value in the range 2-3 (inclusive).

- The PDU set is necessary for the processing of all PDU sets of all streams. Such PDU sets should be assigned the lowest PSI value 1.

[Editor’s Note1] Alignment between all the clauses in Guidelines section is required.

##### 4.4.2.6.3 PDU Set Size Field

The PDU Set Size field may be present in the RTP HE for PDU Set marking if appropriately enabled for an RTP sender as per Clause 4.4.2.5. In case the PDU Set Size is enabled the application shall express the PDU Set Size in bytes as per the PSSize semantics defined in Clause 4.4.2.4.

The PDU Set Size value of a PDU Set should be determined by the RTP sender based on the RTP payload corresponding to the PDU Set, transmission path MTU Size, or alternatively, maximum RTP SDU size, and network IP transport configuration.

The RTP sender should follow the corresponding steps in determining the PDU Set Size.

1. The RTP sender should receive from a media encoder (e.g., a H.264 encoder, a H.265 encoder) payload data corresponding to a PDU Set. It is recommended that all Non-VCL NAL units (e.g. SPS NAL unit) are handled together with the associated VCL NAL units within the same PDU Set. The size of the received payload data (*R*) should be determined in bytes.

2. The RTP sender should perform next RTP fragmentation and packetization of the payload data (*R*). The maximum size of an RTP packet SDU (*S*) should be determined given a transmission path MTU size, or alternatively, a preconfigured maximum RTP SDU payload size less than the path MTU size. The RTP sender should determine the number of RTP packets (*P*) post-fragmentation given *S* and a packetization configuration of the RTP payloader. The RTP payloader should implement the payload formatting according to the corresponding payload type of the PDU Set (e.g., RFC 6184 [5] for H.264, RFC 7798 [6] for H.265) and the packetization configuration to yield the *P* RTP packets’ SDUs. *P* corresponds to the number of PDUs of the PDU Set.

NOTE 1: Some WebRTC implementations [7] in commercial user agents configure a maximum RTP SDU size of 1200 bytes compliant also with the recommendations of RFC 8200 and further corresponding to an MTU Size of 1280 bytes. Other valid configurations exploiting larger MTU Size based on path MTU discovery protocols, RFC 1191, or RFC 8201, may apply up to the RTP stack implementation capabilities.

NOTE 2: It is generally assumed that the configuration of the RTP payloader ensures RTP packets resulting from packetization do not violate the MTU Size. In addition, the RTP payloader may be configured by applications to favor low-latency delivery. For example, in some cases of RTP H.264 payload types, the RTP payloader may be configured to operate in packetization-mode 1 (i.e., "non-interleaved mode" as per RFC 6184 Clause 6.3) to allow for RTP packets to contain NAL units in decoding order and to map an RTP packet to a single NAL unit packet (as per RFC 6184, Clause 5.6), a STAP-A packet (as per RFC 6184, Clause 5.7.1) or a FU-A packet (as per RFC 6184, Clause 5.8). In other cases, applications may select other RTP payloader configuration up to implementation and application requirements.

3. The RTP sender should determine for each one of the *P* RTP packets the size of the RTP header overhead including any header extensions overhead (*Rh\_p*) as configured based on the SDP offer-answer negotiation.

NOTE 3: It may be possible for different PDUs in a PDU Set to contain distinct RTP header extensions besides the common RTP HE for PDU Set marking such that *Rh\_p* may differ among different PDUs of a PDU Set.

4. The RTP sender should further determine per RTP packet the size of the UDP/IP headers overhead associated with an OS UDP socket sending out the RTP packets. This may be done by the RTP sender using UDP socket options available programmatically over OS network stack API calls or based on SDP-configured IP endpoints and corresponding transmission IP addresses. The RTP sender should determine the type of the underlying IP version used for transport, i.e., IPv4 or IPv6, and determine accordingly the IP header overhead (*Ih\_p*) for each encapsulated RTP packet. If IPv4 options are configured for the UDP socket, or alternatively, if IPv6 header extensions are sent over the UDP socket, the RTP sender should consider the additional incurred size these have to the IP header overhead (*Ih\_p*) of each RTP packet. The RTP sender should consider a fixed size UDP header overhead (*Uh*) of 8 bytes for each RTP packet.

NOTE 4: In case no IPv4 header options are used, the RTP sender should consider *Ih\_p* corresponding to 20 bytes per RTP packet for IPv4. Whereas, in case no IPv6 extension headers are used, the RTP sender should consider *Ih\_p* corresponding to 40 bytes per RTP packet for IPv6.

NOTE 5: For example, in case of Linux-based open-source OSs, any additional IPv4 options up to 40 bytes may be set and accessed programmatically based on socket API calls and predefined socket options [8] (e.g., setsockopt/getsockopt, IP\_OPTIONS). Similarly, any additional IPv6 header extensions sent out as control messages to a remote may be set and accessed programmatically based on socket API calls [9] (e.g., sendmsg, IPV6\_DSTOPTS etc.). The RTP sender may make use of these APIs or other ones up to the RTP sender implementation to determine additional optional overheads to the IP header overhead, *Ih\_p*.

5. The RTP sender should determine the PDU Set Size as the sum in bytes of all RTP/UDP/IP headers overhead of each one of the *P* packets and the received RTP payload corresponding to the PDUs of the PDU Set, e.g., *PSSize =R +* $\sum\_{p=1}^{P} $(*Ih\_p + Uh\_p + Rh\_p*). The value should be indicated in the PSSize field of the RTP HE for PDU Set marking for all PDUs of the PDU Set before the corresponding RTP PDUs are sent over the UDP socket.

In case any of the above steps fails to determine for a PDU Set any of the *Ih\_p*, *Uh\_p*, *Rh\_p*, *P*, or *R*,the RTP sender should set the PSSize to 0 for the PDU Set.

NOTE 6: The PDU Set Size guidelines above are generally applicable to video and audio media payload types.

#### 4.4.2.7 Guidelines for AS

This clause describes guidelines for an AS that is on the media path between two or more UEs, e.g., an MRF, MCU etc. Such an AS may receive media over RTP with PDU set marking HE added by the sender UE.

NOTE: These detailed guidelines are FFS.

### 4.4.3 RTP Header Extension for Pose

An RTP sender that uses RTP to deliver pre-rendered video streams to a UE should include an RTP header extension for pose to indicate the XR pose used for rendering the media (rendered pose). The RTP header extension for pose may also be used with audio streams.

The RTP header extension for pose may also be used by a UE to indicate the XR pose to another UE or to a server.

An RTP client that supports the RTP header extension for pose shall negotiate the use the of the extension using SDP. The signaling of the RTP header extension for pose shall follow the SDP signaling design, the syntax, and semantics of the "extmap" attribute as outlined in RFC8285. The header extension shall be registered with IANA.

For IANA registration, the "reference" field in the registry is 3GPP TS 26.522.

The ABNF syntax for this header extension extendsg the "extmap" attribute as follows:

*extensionname* = "urn:3gpp:xr-pose"

*extensionattributes* = ["media:" 1\*(SP token)]

The extension attribute "media" is followed by a list of tokens for "mid" (as defined in RFC 5888) for media streams that can reuse the pose included in the RTP header extension. Further details on reuse are provided later in the section.

An RTP client that supports the RTP header extension for pose and receives an SDP offer with "a=extmap" attribute with the URN: "urn:3gpp:xr-pose" shall remove the attribute from the answer for any media that will not use the extension, and retain it for any media that will use it.

If the RTP header extension for pose is used by a server, the server should use the RTP header extension for pose to associate the selected pose with the rendered frame.

If negotiated successfully, an RTP sender should add the RTP header extension for pose to the RTP stream. The frequency of RTP header extension for pose shall be at least once in a frame. It may be sent more often but not necessarily in every RTP packet.

The 2-byte (RFC 8285) RTP header extension format shall be used for signalling the RTP header extension as follows:

0 1 2 3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0x100 |appbits| length |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| ID | L=36+2n | rx …

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | ry …

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | rz …

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | rw …

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | x …

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | y …

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | z …

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | XR timestamp …

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| XR timestamp continued …

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| XR timestamp continued | action\_id #1 |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| action\_id #2 | ... |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

The fields rx, ry, rz, rw, x, y, z are defined in single-precision floating-point format (binary32 as per ISO/IEC 60559:2020).

**rx (32 bits):** x coordinate of the orientation quaternion of the pose.

**ry (32 bits):** y coordinate of the orientation quaternion of the pose.

**rz (32 bits):** z coordinate of the orientation quaternion of the pose.

**rw (32 bits):** w coordinate of the orientation quaternion of the pose.

 in meters

 in meters

 in meters

[Editor’s Note: Definition of the header extension for 3DoF pose is FFS.]

**[**

**XR timestamp (64 bits)**: Timestamp for the pose. If the header extension is used for rendered pose, this timestamp indicates the predicted XR runtime display time. Otherwise, this timestamp indicates the associated XR runtime display time for the predicted XR pose. XR timestamp uses the XR system clock and is represented in nanoseconds. The timestamp is passed to the XR runtime together with the rendered swapchain images (e.g. as part of the xrEndFrame call in OpenXR).

]

NOTE 1: It is left to the discretion of the application how to use the XR timestamp.

[Editor’s Note: Rendered pose is sent from the SR server to the SR client. If the pose is not rendered pose, it is sent from a UE to a server or to another UE.]

**action\_id (32 bits)**: A list of actions corresponding to the pose x, y, z, rx, ry, rz, rw coordinates. An action\_id uniqely identifies an action and it may be an action identifier as defined in the action format of TS 26.119 Clause 6.2.3. The number of action identifiers in one RTP header extension for pose shall be no more than 10. Hence, the size of the header extension is 36+2\*n, where n is the number of action identifiers in the header extension.

If the RTP header extension for pose is sent by a server, it should contain an action\_id field as defined above, with the list of action identifiers identifying the processed actions for the rendering of the frame.

If the RTP header extension for pose is sent by a UE, it should contain an action\_id field as defined above, with the list of action identifiers identifying the action for which the pose coordinates apply.

NOTE 2: A peer to a UE XR client should be aware of the UE actions configuration in an action space. Signalling aspects for the UE actions configuration are defined in other specifications such as TS 26.119 and TS 26.565.

NOTE 3: An XR server should be aware of the XR space used by the XR client for the pose fields defined above. Signalling aspects for this XR space are defined in other specifications such as TS 26.119 and TS 26.565.

When both video and audio are delivered to an RTP receiver, or when either audio or video is delivered using multiple real-time streams (e.g., left eye + right eye), multiple RTP streams may be associated with the same header extension data, e.g., the same pose may have been used for generating multiple streams. This may lead to sending the same header extension data multiple times in different streams.

A sender may reuse the pose RTP header extension of one stream for multiple RTP streams. For example, only the video stream carries the pose RTP header extension, but the pose is applicable also for the audio bitstream. In this case, the sender shall include the extension attribute media followed by a semi-colon separated list of media ID (MID) values in the "a=extmap" attribute. The MID values indicate all media streams for which the pose RTP header extension is applicable to. If the extension attribute media is present, then the media description of all bitstreams that reuse the header extension shall include the attribute "mid" as defined in RFC 5888.

NOTE 4: In case there is a mismatch between the frame rates of the streams, the receiver may use the few most recent samples from the source RTP stream to obtain a synchronized sample in the dependent stream via interpolation. Alternatively, the receiver may choose to not perform any interpolation and simply use the last available sample from the source RTP stream for the dependent stream. It is left to the discretion of the receiver application to select an appropriate synchronization method.

### 4.4.4 RTP Header Extension for in-band end-to-end delay measurement

#### 4.4.4.1 General

An RTP Header Extension that allows an RTP packet to carry timestamp(s) may help obtain measured delays that are representative of the end-to-end instantaneous delays experienced by the media in the user plane.

NOTE 1: End-to-end connections may imply in some cases a multi-hop link including non-3GPP network paths, such as a data network link and a tethering link.

Figure 4.4.4.1-1 shows how the RTP Header Extensions are used to measure the delays, where T1, T2, T3 and T4 are the Originate Timestamp, the Receive Timestamp, the Transmit Timestamp, and the Destination Timestamp, respectively. The one-way delay from the Requester to the Responder is calculated as T2 - T1, the one-way delay in the opposite direction is calculated as T4 – T3, the RTT is calculated as (T4 – T1) – (T3 – T2), and the processing delay on the Responder is calculated as T3 – T2.

NOTE 2: Time synchronization between the Requester and the Responder for example via PTP [14] is a pre-requisite for computation of one-way delays in any direction.

NOTE 3: The Requester may use T1 to group T1, T2, T3, T4 measurements and index them to compute all the above delay measurements and any corresponding statistics. Specific means to achieve this are left to RTP implementers.



Figure 4.4.4.1-1: The RTP header extensions for in-band end-to-end delay measurement.

The RTP Header Extensions defined below follow RFC 8285 [11].

####  4.4.4.2 One-byte RTP Header Extension Format

The RTP header extension element for the RTP packet that carries ony one timestamp T1 is shown below. This is the same as the **"**RTP Header Extension for Absolute Sender Time" [12].

 0 1 2 3

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

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | 0xBE | 0xDE | length |

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | ID | L=2 | T1 (24 bits) |

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

The RTP header extension element for the RTP packet that carries three timestamps T1, T2 and T3 is shown below.

 0 1 2 3

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

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | 0xBE | 0xDE | length |

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | ID | L=8 | T1 |

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | T2 | T3 …

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 |

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

#### 4.4.4.3 Two-byte RTP Header Extension Format

The RTP header extension element for the RTP packet that carries one timestamp T1 is shown below.

 0 1 2 3

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

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | 0x100 |appbits| length |

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | ID | L=3 | T1 …

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 |

 +-+-+-+-+-+-+-+-+

The RTP header extension element for the RTP packet that carries three timestamps T1, T2 and T3 is shown below.

 0 1 2 3

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

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | 0x100 |appbits| length |

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | ID | L=9 | T1 …

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | T2 |

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | T3 |

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

#### 4.4.4.4 Syntax

**T1:** consists of 24 bits, taken from the 6 LSB bits of the integer part and the 18 MSB bits of the fractional part of the NTP timestamp format defined in RFC 5905 [13].

**T2:** follows the syntax of T1.

**T3:** follows the syntax of T1.

NOTE: The timestamps are expressed in seconds according to the above syntax, with a 64 second wraparound and a 3.8 microsecond resolution.

#### 4.4.4.5 Semantics

**T1: Originate Timestamp.** It represents the time when the Requester transmits the RTP packet toward the Responder.

**T2: Receive Timestamp.** It represents the time when the Responder receives the RTP packet that carries the Originate Timestamp T1.

**T3: Transmit Timestamp.** It represents the time when the Responder transmits the RTP packet that carries the Originate Timestamp T1, the Receive Timestamp T2, and the Transmit Timestamp T3.

#### 4.4.4.6 SDP signaling

The signaling of the delay measurement RTP header extensions shall follow the SDP signaling design and the syntax and semantics of the "extmap" attribute as outlined in RFC 8285 [11]. The header extension identifiers shall be registered with IANA. The "reference" entry in the IANA registry shall be "3GPP TS 26.522 [v18.x.x.x]".

For the RTP header extension carrying only T1, the ABNF syntax for the "extmap" attribute is as follows:

*extensionname* = "http://www.webrtc.org/experiments/rtp-hdrext/abs-send-time"

*extensionattributes* = ["short"/"long"]

If the *extensionattributes* is absent, the RTP header extension follows the one-byte format, i.e., the "short" format. If *extensionattributes* is "short", the RTP header extension follows the one-byte format. If *extensionattributes* is "long", the RTP header extension follows the two-byte format.

NOTE 1: http://www.webrtc.org/experiments/rtp-hdrext/abs-send-time is the extension URI of the RTP header extension, and is currently implemented in WebRTC. This extension URI, instead of URN-based ones, allows for support from WebRTC without any change to the WebRTC implementation.

NOTE 2: This allows to reuse the "Absolute Sender Time" RTP header extension in WebRTC without changes to the SDP syntax implemented in WebRTC.

Below is an example (Example 1):

a=extmap:4 http://www.webrtc.org/experiments/rtp-hdrext/abs-send-time

For the RTP header extension carrying T1, T2 and T3, the ABNF syntax for the "extmap" attribute is as follows:

*extensionname* = "urn:3gpp:delay-measurement-response:rel-18"

*extensionattributes* = [format SP] binding-info

format = "short"/"long"

binding-info = dependent-extmap-ID [";"m-line-label] [";"processing-ID]

dependent-extmap-ID = "dependent-extmap-ID="1\*5DIGIT

m-line-label = "dependent-rtp-he-m-line-label="token

processing-ID = "processing-ID="token

; token as defined by RFC 4566

The extension attributes have the following semantics:

- dependent-extmap-ID: identifies an RTP header extension on which this RTP header extension depends in the sense that the timestamps T1 and T2 incliuded in this RTP header extension are the time the other RTP header extension is transmitted and the time the other RTP header extension is received, respectively.

- processing-ID: identifies a processing module on the Responder which takes data carried in RTP packets with the RTP header extension identified by dependent-extmap-ID, processes them and produces data that are then carried in RTP packets with this RTP header extension.

NOTE 3: The details of processing-ID are left to implementation at the application level.

- m-line-label: is the SDP "label" attribute defined in RFC 4574 [15], and it identifies a media stream from the Requester to the Responder and associates the RTP packet header extension in that media stream to this RTP header extension.

NOTE 4: There may be multiple media streams that carry RTP packets whose RTP header extensions may be used for the binding.

Below is an example (Example 2):

a=extmap:5 urn:3gpp:delay-measurement-response:rel-18 short dependent-extmap-ID=4;dependent-rtp-he-m-line-label=2;processing-ID=7

In the example,

* 5 is the RTP header extension ID
* 4 is the value of the attribute dependent-extmap-ID, which is the RTP header extension ID of the RTP header extension in Example 1. This establishes a binding between the two RTP header extensions.
* 7 is the processing-ID.
* 2 is the SDP "label" attribute that identifies the media stream corresponding to "a=label:2" in the SDP signaling, and the RTP packets from the media stream are used for the binding.

## 4.5 RTP Forward Error Correction

TBA

## 4.6 SRTP

TBA

# 5 RTCP Feedback Reporting Procedures

## 5.1 General

This clause defines the RTCP feedback reporting messages to transmit control information. There are a number of possible ways to carry a variety of control information using RTCP packets. This includes:

- profile-specific extensions to the sender (PT=200) and receiver report (PT=201),

- application-defined RTCP packet with payload type equal to 204 (PT=204),

- Generic RTP Feedback reports with payload type equal to 205 (RTPFB; PT=205), and

- payload-specific RTCP feedback messages with payload type equal to 206 (PSFB; PT= 206),

- extended reports (XR) with payload type equal to 207 (PT=207).

## 5.2 Transmission of timing information data for QoE measurements

### 5.2.1 General

In use cases for shared interactive immersive services, the user interaction information is sent from a UE to a server. The server handles the user’s request to the immersive media scene (e.g., changing the context such as translation, rotation, and scaling or adding a new object in the scene). In the case of the edge-assisted UE type, the UE offloads the scene rendering to the Split Rendering Server (SRS) and the server rasterizes the XR viewport and does pre-rendering to generate the XR media which is encoded and delivered to the UE.

In the context of interactive immersive services, one important parameter to estimate the user quality of experience is the *roundtrip interaction delay.* The *roundtrip interaction delay* is defined as the sum of the *age of content* and the *user interaction delay.*

The*age of content* is defined as the time duration between the moment the content is created and the time it is presented to the user. It is impacted by the downlink latency of the wireless network.

The *user interaction delay* is defined as the time duration between the moment at which a user action is initiated and the time such an action is taken into account by the content creation engine. It is impacted by the uplink latency of the wireless network.

The *estimated-at-time* (T1) and *start-to-render-at-time* (T3) provide the times when the pose was estimated and when the SRS started to render the rendered frame, respectively. The *split-renderer-output-time* (T5) provides the time when the output of the SRS for a rendered frame is available. This T5 information can be used to measure the server processing delay and the overall application delay excluding the server processing delay. The SRS processes the interaction according to the actions in the action message from the UE and updates the scene. The Scene Manager records the *sceneUpdateTime* (T6) timestamp when it starts to process the actions. The *sceneUpdateTime* is used to measure the user interaction delay, age of content and the roundtrip interaction delay. The details of *sceneUpdateTime*, measurement of *User-interaction-delay*, *Age-of-content* and *Roundtrip-interaction-delay* QoE interaction metrics.

The *user interaction delay*, *age of content*, and *round-trip interaction dela*y measurements are described as quality of experience metrics for XR content. These delay measurement metrics need to be calculated at the UE for providing better QoE to the user. Also, the server processing delay measurement helps the UE in the adaptation process with the split rendering server for achieving better QoE.

### 5.2.2 RTCP message-based transmission of timing information

The timing information data recorded at the SRS or at the RTP sender can be transmitted to the UE by enhancing the RTCP XR packets, which are specified in IETF RFC 3611 [16]. The RTCP XR report is identified by payload type (PT) equal to 207, which refers to an extended report block message. For transmission of timing information data using RTCP XR messages, the block type (BT) defined in RFC 3611 can be extended with a value TBD.

NOTE: The block type value for the QoE timing information RTCP XR message will be assigned by IANA and the specification will be updated with that block type value later.

####  5.2.2.1 Extended Report block for QoE timing information

This extended report block type permits detailed reporting of timing information recorded at the SRS. These reports can be used, for example, for calculating the QoE metrics such as *round-trip delay*, *server processing delay*, *user interaction delay*, *age of content* and the *round-trip interaction delay* at the UE.

The timing information required for measuring QoE metrics may be expressed in the same units as in the RTP timestamps of RTP data packets. This is so that, for each packet, one can establish the relation between the media data flowing and the corresponding QoE timing information recorded at the SRS for a specific media frame.

For optimum use of the RTCP bandwidth, the RTCP XR block payload may contain the whole or part of the timing information required to calculate the QoE metrics. time\_info field present in Figure 1 represents the timing information present in an RTCP XR block report. When a bit is set to ‘ONE’ in time\_info field the respective timing information shall be present in the payload. When a bit is set to ‘ZERO’ in time\_info field, the respective time information shall not be present in the payload. E.g., when the sender like to transmit only T1 and T3 information, the time\_info field is set to b0011 and only T1 and T3 information is present in the message payload.

The identifiers of all actions that were processed for the rendering of a frame at a specific time are reported in the "Pose" RTP header extension. This header extension is identified using the "a=extmap" attribute URN: "urn:3gpp:xr-pose". The synchronization between the various timing information present in the below XR report and the action identifiers present in the "Pose" RTP HE messages is performed using the RTP timestamp information present in the RTP header of the packet containing the "Pose" RTP HE and the RTP timestamp field present in the below XR report block.

The QoE timing information Report Block has the following format:

 0 1 2 3

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

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | **BT=TBD** | resv |time\_info| block length |

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | SSRC of source |

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | RTP time stamp |

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | estimated-at-time (T1) |

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | start-to-render-at-time (T3) |

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | server-output-time (T5) |

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | scene-update-time (T6) |

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Figure 5.2.2.1-1: RTCP XR block format for QoE timing information data

The semantics of the fields in QoE time information Extended Report (XR) block are as follows:

- block type (BT) [8 bits]: A QoE time information Report Block is identified by a constant value.

- block length [16 bits]: The length of this report block, including the header, in 32-bit words minus one.

- resv [4 bits]: This field is reserved for future definition. In the absence of such definition, the bits in this field MUST be set to zero by the sender and MUST be ignored by the receiver.

- time\_info [4 bits]: This field bits represent the time stamps that are present in the XR report block. When T1 is present in the XR report, 1st bit (least significant bit) is set to 1. When the LSB is set to 0, T1 information shall not be present. When T3, T5 and T6 are present in the RTCP XR block data, bits 2, 3 and 4 are set to 1 respectively. When T1, T3, T5 and T6 are present in an RTCP XR block data, the time\_info field value shall be b1111. The timing information when present shall follow the order T1, T3, T5 followed by T6. For example, when the time\_info field value is b0101, the XR block carries the T1 information followed by T5. T3 and T6 timing information will not be present in that XR block content.

- The transmission frequency of T1, T3, T5 and T6 time information in XR report block can be negotiated during the configuration phase.

- SSRC of source [32 bits]: The SSRC of the RTP data packet source being reported upon by this report block.

- RTP timestamp [32 bits]: This field represents the RTP time stamp of the media frame at which the corresponding QoE timing information date was recorded at the SRS. This correspondence may be used for synchronization between the media data and the QoE timing information measurements recorded at the SRS for a specific media frame.

- estimated-at-time (T1) [32 bits]: This field represents the time when the pose estimation was made. This time information is expressed in the same units and with the same random offset as the RTP timestamps in data packets.

- start-to-render-at-time (T3) [32 bits]: This field represents the time when the renderer in the split rendering server started to render the associated media frame. This time information is expressed in the same units and with the same random offset as the RTP timestamps in data packets.

- server-output-time (T5) [32 bits]: This field represents the recorded time at the output of the split rendering server. This time information is expressed in the same units and with the same random offset as the RTP timestamps in data packets.

- scene-update-time (T6) [32 bits]: This field represents the time when the Scene manager processes the interaction task according to the actions in the action message from the UE and updates the scene. This time information is expressed in the same units and with the same random offset as the RTP timestamps in data packets.

### 5.2.3 SDP signaling and attributes

RFC 3611 [16] defines the SDP attribute "rtcp-xr" that can be used to signal to participants in a media session that they should use the specified XR blocks. This attribute is extendable with new parameters to cover any new type of XR report blocks.

The extended RTCP XR blocks with QOE time information SDP attribute is defined as below in Augmented Backus-Naur Form (ABNF).

rtcp-xr-attrib = "a=" "rtcp-xr" ":" [xr-format \*(SP xr-format)] CRLF

 xr-format = pkt-loss-rle

 / pkt-dup-rle

 / pkt-rcpt-times

 / rcvr-rtt

 / stat-summary

 / voip-metrics

 / **qoe-timing-info**

 / format-ext

 pkt-loss-rle = "pkt-loss-rle" ["=" max-size]

 pkt-dup-rle = "pkt-dup-rle" ["=" max-size]

 pkt-rcpt-times = "pkt-rcpt-times" ["=" max-size]

 rcvr-rtt = "rcvr-rtt" "=" rcvr-rtt-mode [":" max-size]

 rcvr-rtt-mode = "all"

 / "sender"

 stat-summary = "stat-summary" ["=" stat-flag \*("," stat-flag)]

 stat-flag = "loss"

 / "dup"

 / "jitt"

 / "TTL"

 / "HL"

 voip-metrics = "voip-metrics" ["=" max-size]

 **qoe-timing-info= "qoe-timing-info"** ["=" max-size]

 max-size = 1\*DIGIT ; maximum block size in octets

 DIGIT = %x30-39

 format-ext = non-ws-string

 non-ws-string = 1\*(%x21-FF)

 CRLF = %d13.10

The "rtcp-xr" attribute contains zero, one, or more XR block related parameters. Each parameter signals functionality for an XR block, or a group of XR blocks. The attribute is extensible so that parameters can be defined for any future XR block. The parameters are extended to support delivery of QoE timing information data over RTCP packets with XR type.

The parameter names and their corresponding XR formats are as follows:

 Parameter name XR block (block type and name)

 -------------- ------------------------------------

 pkt-loss-rle 1 Loss RLE Report Block

 pkt-dup-rle 2 Duplicate RLE Report Block

 pkt-rcpt-times 3 Packet Receipt Times Report Block

 stat-summary 6 Statistics Summary Report Block

 voip-metrics 7 VoIP Metrics Report Block

 **qoe-timing-info TBD Timing information for QoE metrics calculation**

The "qoe-timing-info", parameter MAY specify an integer value. This value indicates the largest size the whole report block SHOULD have in octets.

Annex A (informative):
Guidelines for PDU Set identification

# A.0 General

This informative annex provides guidelines for network functions like the UPF, which needs to determine PDU Set information, as described in TS 23.501 [X], Clause 5.37.5. The network function is typically provisioned with at least the Service Data Flow Filter to identify the Service Data Flow, and optionally additional information about the presence of RTP header extensions according to IETF RFC 8285 [11], the used RTP Payload Type, the used RTP Payload Format and other information.

When the RTP sender multiplexes RTP data and control packets onto the same Service Data Flow using a single port, the RTP Sender should implement the Payload Type separation according to IETF RFC 5761, Clause 4 [10] and the network function should separate RTP data from RTCP data accordingly.

To avoid IP fragmentation, the RTP sender should select a sufficiently small RTP payload.

A.1 Leveraging RTP Header Extensions

When the PDU Set related RTP Header Extensions are available within the RTP headers, the network function only needs parse the RTP header and the RTP header extensions. The RTP Header Extension for PDU Set Marking are defined in Clause 4.4.2.

An intermediate network function determines based on the RTP header X bit being set to 1, whether the optional header extension fields are present in the RTP packet, after the SSRC and the (optional) CSRC fields in the RTP header. All information for the PDU Set identification is present within the RTP Header Extension and the network function does not need to know the RTP Payload format. The RTP Payload may be encrypted (i.e. SRTP).

When multiple RTP header extensions are present within the RTP header, the network function uses the RTP Header Extendion ID for finding the PDU Set related HE.

Editor’s Note: It is ffs, whether guideline on the usage of the Protocol Description is needed.

A.2 Obtaining PDU Set information from RTP Payload

A.2.0 General

When the PDU Set based RTP Header Extension is not available, some or all of PDU Set information can be derived from the RTP/SRTP header, header extension and/or payloads, e.g., by a network function like the UPF. The possible PDU Set information to be derived based on the RTP/SRTP header, header extension and the payloads are provided as following.

## A.2.1 RTP/SRTP header

When RFC 6184 [5] or RFC 7798 [6] are used as payload formats, a network function can obtain some of the PDU Set information from RTP headers by following these guidelines.



Figure A.2.1-1: RTP header fields as defined in RFC 3550 [4]

When the RTP/SRTP is used to convey the video content and when the PDU Set represents a video frame, the video frame may be identified based on the RTP header fields as following:

- The "marker (M)" bit is used with the video payload formats in clause A.2 to indicate the frame boundary, by setting the M bit on the last PDU of a frame. With the "M" bit and the sequence number in RTP header, the Indication of End PDU of a PDU Set and PDU SN within a PDU Set/frame can be derived. The network function should monitor the preceeding packets to detect and compensate for potential packet reordering.

- The "timestamp" field indicates the sampling instant of the first octet in the RTP data packet and all RTP packets in the video frame is generally marked with the same timestamp. Therefore, with the "timestamp" field and the sequence number in RTP header, the Indication of End PDU of a PDU Set and PDU SN within a PDU Set/frame can be derived.

NOTE 1: When multiple RTP streams multiplexed over a single RTP session, the "M" bit, "timestamp" field, and sequence number information can be used together with the synchronization source (SSRC) in the RTP header to identify the boundary of video frame for each of the RTP streams that can be separated by their different SSRC values.

NOTE 2: For the timestamp-based solution, generally, the end PDU of the PDU Set can only be determined when a PDU with new RTP timestamp arrives, which may introduce additional latency.

- The PDU Set size can only be determined by a network function with reception of the last PDU belonging to the PDU Set, by summing up the individual PDU contributions to the PDU Set size. The PDU Set importance cannot be derived by the RTP header fields.

## A.2.2 RTP payload

### A.2.2.1 General

When the RTP Payload is not encrypted, intermediate network functions may obtain additional information from the RTP payload.

The PDU Set information identification based on the RTP payload format is presented in this clause, including information on the RTP payload formats for H.264/AVC [5] and H.265/HEVC [6] codecs. The information about the used RTP Payload format for a service data flow is provided in advance to 5GC (e.g., UPF).

It is generally recommended that the network function consideres Non-VCL NAL units (e.g. SPS NAL unit) as part of the PDU Set of the associated VCL NALUs, e.g. identified by the same timestamp.

### A.2.2.2 RTP payload for H.264/AVC codec

For a video content with H.264 RTP payload, the PDU Set Information can be realized by following approach.

According to RFC 6184 [5], the first octet in the RTP payload indicates the content of the RTP payload, e.g. coded slice of an IDR frame, coded slice of a P frame, and also the possible structures of the RTP payload, e.g. single NAL unit packet, aggregation packet and fragmentation unit (FU). Depending on the indication of the first octet of the RTP payload, a second octet (the FU header) should also be processed.

* For single NAL unit packet and aggregation packet, it can be easily detected that each RTP packet can be treated as a complete PDU Set when the first Type field of Figure A.2.2-1 is less than 28.
* In case of aggregation packets, the network function may need to process all embedded NAL units.
* When the first Type field in Figure A.2.2-1 is 28 or 29, one NAL unit is carried over multiple RTP packets. In this case, the first byte of RTP payload is also named the fragmentation unit (FU) indicator and the following byte is the FU header. The NAL unit type is contained in the Type field of rhe FU header (Figure A.2.2.-1). In the FU header, the "S" bit and "E" bit separately represents the start and end of the NAL unit. Therefore, based on the NAL unit type (also known as FU indicator for fragmentation unit) and the FU header, the start/end of the PDU Set can be identified.



Figure A.2.2-1: RTP header [4] and NALU header format for H.264 [2]

With the RTP payload (i.e. NALU header and optionally Fragmentation Unit (FU) header) and the sequence number in the RTP header, the indication of the End PDU of the PDU Set and the PDU SN within a PDU Set can be derived.

When using Fragmentation Units (Type equals 28 or 29), the size of the NALU can only be determined after reception of the last packet of this Fragmentation Unit. Thus, a network function can only determine the PDU Set Size with the reception of the last PDU of this fragmentation unit.

As described in clause 4.4.2.6.2.2, the Type and NRI value in the NAL unit header indicates the relative transport priority and can be used to set the PDU Set importance. Besides, different NRI values can also indicate different requirements, which can be used to provide different protections against transmission losses, e.g. reliabilities (tolerable frame/slice error rate), and priorities.

### A.2.2.3 RTP payload for H.265/HEVC codec

For a video content with H.265 RTP payload, the identification of the PDU Set can be realized by following approach.

According to RFC 7798 [6], within the RTP packet, the first two octets of the RTP payload indicate the content of RTP packet. Besides, it also indicates the possible structures of the RTP payload, e.g. single NAL unit packet, aggregation packet (APs), fragmentation unit (FUs), and Payload Content Information (PACI) carrying RTP packet.

- For single NAL unit packet and aggregation packet, it can be easily detected that each RTP packet can be treated as a single PDU Set when the NAL unit type is less than 49.

- When NAL unit type is 49, one NAL unit is carried over multiple RTP packets. In this case, the first two-byte of RTP payload is also named the payload header (denoted as NALU header) and the following byte is the FU header. In the FU header, the "S" bit and "E" bit separately represents the start and end of the NAL unit. The FuType field contains the actual NAL unit type. Therefore, based on the Type field of the first two octets (also known as FU indicator for fragmentation unit) and the FU header, the start/end of the PDU Set can be identified.

- When NAL unit type is 50, this is a PACI packet which may carry a single NAL unit packet or FU. In this case, the first two-byte of RTP payload is also named as the PACI header (denoted as NAL Unit header). In the following two bytes, the "A" bit is the copy of "F" bit and cType field is the copy of Type field in the PACI payload NAL unit. Then the following is the PHES field, whose length is determined by the PHSize. Finally, the following is the PACI payload NAL unit, during which the first byte is FU header when cType (within the PACI payload header) is 49. Therefore, based on the PACI header and PACI payload NAL unit, the start/end of the PDU Set can be identified.



Figure A.2.3-1: The Structure of the HEVC NAL Unit Header [6]



Figure A.2.3-2: The Structure of FU Header

With the RTP payload (i.e. NAL unit header and optionally FU header) and the sequence number in the RTP header, the indication of the End PDU of the PDU Set and the PDU SN within a PDU Set can be derived.

As described in clause 4.4.2.6.2.3, the Type field and the TID field in the NAL unit header indicates the relative transport priority and can be used to be mapped to the PDU Set importance. They can also indicate different requirements, which can be used to provide different protections against transmission losses, e.g. reliabilities (tolerable frame/slice error rate), and priorities.

When using Fragmentation Units (Type equals 49, or 50 where cType is 49), the size of the NALU can only be determined after reception of the last packet of this Fragmentation Unit. Thus, a network function can only determine the PDU Set Size with the reception of the last PDU of this fragmentation unit.

Annex B (informative):
Examples of SDP offers and answers

# B.1 SDP example for pose RTP header extension

An example SDP description using the pose RTP header extension is presented below. Using the extension attribute media, the pose RTP header extension with URI urn:3gpp:xr-pose provided in the video stream with MID m1 is also applicable to another video stream with MID m3.

v=0

o=alice 2890844526 2890844526 IN IP4 host.atlanta.example.com

s=SDP Session

c=IN IP4 host.atlanta.example.com

t=0 0

m=application 1001 UDP/DTLS/SCTP webrtc-datachannel

a=sendonly

m=video 23458 RTP/AVP 96

a=mid:m1

a=recvonly

a=rtpmap:96 H264/90000

**a=extmap:1 urn:3gpp:xr-pose media:m3**

m=audio 23468 RTP/AVP 97

a=mid:m2

a=recvonly

a=rtpmap:97 PCMU/8000

m=video 23478 RTP/AVP 97

a=mid:m3

a=recvonly

a=rtpmap:96 H264/90000

Annex C (informative):
Change history

|  |
| --- |
| Change history |
| Date | Meeting | TDoc | CR | Rev | Cat | Subject/Comment | New version |
| 2023-04 | SA4#123-e | S4-230719 |  |  |  | Initial version, with text from WID in SP-220613 and S4-230713 | 0.0.1 |
| 2023-05 | SA4#124 | S4-231044 |  |  |  | Implementing S4-230848, S4-230965, S4-231026, S4-231028 | 0.0.2 |
| 2023-05 | SA4#124 | S4-231101 |  |  |  | Agreed version | 0.1.0 |
| 2023-08 | SA4#125 | S4-231544 |  |  |  | Implementing S4-231440, S4-231524, S4-231533 | 0.1.1 |
| 2023-11 | SA4#126 | S4-231752 |  |  |  | Implementing S4aR230101, S4aR230106 | 0.1.2 |
| 2023-11 | SA4#126 | S4-231983 |  |  |  | Implementing S4-231756, S4-231758, S4-231925, S4-231927, S4-231928, S4-231929, S4-231930, S4-232028 | 0.2.0 |