|  |  |
| --- | --- |
| 3GPP TR 26.822 V1.3.0 (2024-11) | |
| Technical Report | |
| 3rd Generation Partnership Project;  Technical Specification Group Services and System Aspects;  Study on 5G Real-time Transport Protocol Configurations, Phase 2;  (Release 19) | |
|  | |
|  |  |
|  | |
| 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 address  3GPP support office address  650 Route des Lucioles - Sophia Antipolis  Valbonne - FRANCE  Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16  Internet  http://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.  © 2024, 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 members  3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  GSM® and the GSM logo are registered and owned by the GSM Association |

Contents

Foreword 10

1 Scope 12

2 References 12

3 Definitions of terms, symbols and abbreviations 14

3.1 Terms 14

3.2 Symbols 15

3.3 Abbreviations 15

4 Architectural Assumptions and Requirements 16

4.1 Architectural Assumptions 16

4.2 Architectural Requirements 17

5 Key Issues 18

5.1 Key Issue #1: Inaccuracy of the PDU Set Size (PSSize) information 18

5.1.1 Description 18

5.2 Key Issue #2: QoS handling requirements for lone PDU 18

5.2.1 Description 18

5.3 Key Issue #3: Enhancements for application-layer FEC support 19

5.3.1 Description 19

5.4 Key Issue #4: AL-FEC awareness for PDU Set handling 19

5.4.1 Description 19

5.5 Key Issue #5: RTP transport of XR metadata 20

5.5.1 Description 20

5.6 Key Issue #6: PDU Set marking for XR streams with RTP end-to-end encryption 20

5.6.1 Description 20

5.7 Key Issue #7: RTCP messages to better support XR services in 5G 21

5.7.1 Description 21

5.8 Key Issue #8: RTP retransmission in supporting XR services in 5G 21

5.8.1 Description 21

5.9 Key Issue #9: Feasibility of RTP multiplexing options for transport of XR media streams 21

5.9.1 Description 21

5.10 Key Issue #10: Use cases and intended deployment scenarios for enhancements of RTP header extension for PDU Set marking 22

5.10.1 Description 22

5.11 Key Issue #11: Enhancements of RTP header extension for PDU Set marking 22

5.11.1 Description 22

5.12 Key Issue #12: Enhancements of Data Burst Marking 22

5.12.1 Description 22

5.13 Key Issue #13: Applicability of the RTP header extension for PDU Set marking to different PDU Set types 23

5.13.1 Description 23

5.14 Key Issue #14: Traffic detection and QoS flow mapping for multiplexed media stream data flows 23

5.14.1 Description 23

5.15 Key Issue #15: Media and metadata delivery over multiple sessions 24

5.15.1 Description 24

6 Solutions 25

6.0 Mapping of Solutions to Key Issues 25

6.1 Solution #1: Different PDU Set types to support handling of immersive media 25

6.1.1 Key Issue mapping 25

6.1.2 Description 26

6.2 Solution #2: Gap analysis on the QoS requirements for lone PDUs 27

6.2.1 Key Issue mapping 27

6.2.2 Description 27

6.2.3 Conclusion 29

6.3 Solution #3: SRTP Usage for end-to-end encryption 29

6.3.1 Key Issue mapping 29

6.3.2 Description 29

6.4 Solution #4: Measurement Based Pre-compensation for PDU Set Size Correction 29

6.4.1 Key Issue mapping 29

6.4.2 Description 30

6.5 Solution #5: Introduction of AL-FEC schemes defined in IETF 36

6.5.1 Key Issue mapping 36

6.5.2 Description 36

6.5.3 Categorization 38

6.6 Solution #6: Time to next burst extension for the RTP HE for PDU Set marking 39

6.6.1 Key Issue mapping 39

6.6.2 Background 39

6.6.3 Description 40

6.7 Solution #7: PDU Set Size information correction by indicating the remaining PDU Set Size in RTP header extension 40

6.7.1 Key Issue mapping 40

6.7.2 Description 40

6.8 Solution #8: Definition of the PDU Set for Application-Layer FEC 42

6.8.1 Key Issue mapping 42

6.8.2 Description 42

6.9 Solution #9: RTP retransmission aware PDU Set handling 43

6.9.1 Key Issue mapping 43

6.9.2 Background 43

6.9.2.1 RTP retransmission payload format defined in IETF 43

6.9.2.2 WebRTC usage 45

6.9.2.3 Feasibility of RTP retransmission in XR applications 45

6.9.3 Description 46

6.10 Solution #10: AL-FEC awareness at RAN while considering upstream and downstream packet losses 47

6.10.1 Key Issue mapping 47

6.10.2 Description 47

6.10.2.1 Motivation 47

6.10.2.2 Solution 48

6.11 Solution #11: PSI indication to optimize RTP retransmission 50

6.11.1 Key Issue mapping 50

6.11.2 Description 51

6.12 Solution #12: MID packet filtering 52

6.12.1 Key Issue mapping 52

6.12.2 Description 52

6.13 Solution #13: Dynamic traffic characteristics of RTP sender implementations 54

6.13.1 Key Issue mapping 54

6.13.2 Description 54

6.13.2.1 General 54

6.13.2.2 WebRTC and paced sender implementation 54

6.13.2.3 GStreamer multimedia framework RTP implementation for P2P 57

6.13.2.4 Server-side senders 60

6.13.2.5 Aggregate Statistics 64

6.13.3 Discussion and conclusion 66

6.13.4 Conclusion 67

6.14 Solution #14: Candidate RTCP messages and RTP header extensions to support XR services in 5G 67

6.14.1 Key Issue mapping 67

6.14.2 Description 67

6.14.2.1 RTCP messages 67

6.14.2.2 RTP header extensions 70

6.15 Solution #15: PSI signaling for lone PDUs 70

6.15.2 Description 70

6.15.2.1 Background 70

6.15.2.2 Solution description 72

6.15.2.3 Analysis of the solution 72

6.16 Solution #16: RTP header extension for dynamic traffic characteristics 73

6.16.1 Key Issue mapping 73

6.16.2 Description 73

6.16.2.1 Intended usage in 5GS 74

6.16.2.3 One-byte RTP header extension format 74

6.16.2.4 Two-byte RTP Header Extension Format 75

6.16.2.5 Semantics 75

6.16.2.6 SDP Signaling 76

6.16.2.7 Guidelines fordynamic traffic characteristics signaling 76

6.16.2.8 Proposed Annex D.3 77

6.16.2.9 Discussion of the solution 77

6.17 Solution #17: Analysis of AL-FEC awareness in 3GPP 77

6.17.1 Key Issue mapping 77

6.17.2 Description 77

6.17.2.1 Common attributes of AL-FEC deployments 77

6.17.2.2 End-to-end transport perspective 79

6.17.2.3 Overhead of AL-FEC 80

6.18 Solution #18: Real-time communication congestion control algorithms and AL-FEC 83

6.18.1 Key Issue mapping 83

6.18.2 Description 83

6.18.2.1 General 83

6.18.2.2 Google Congestion Control (GCC) 83

6.18.2.3 PCC 85

6.18.2.4 NADA 85

6.18.2.5 SCReAMv2 85

6.18.2.6 Summary of congestion control algorithms 86

6.18.2.7 Packet loss rate calculation for AL-FEC 86

6.19 Solution #19: Congestion control enhancement to support AL-FEC awareness handling 87

6.19.1 Key Issue mapping 87

6.19.2 Description 87

6.19.2.1 Background of using AL-FEC for real-time communication in cellular networks 87

6.19.2.2 Potential Benefits of Application-layer FEC awareness for PDU Set handling 88

6.19.2.3 Implications of Application-layer FEC awareness for PDU Set handling on congestion control 89

6.19.2.4 The Proposed Solution 89

6.20 Solution #20: Guidelines for PDU Set Marking of Unmarked/Lone PDUs 90

6.20.1 Key Issue mapping 90

6.20.2 Description 90

6.20.2.1 General 90

6.20.2.2 Guideline for PDU Set marking of lone/unmarked PDUs 90

6.20.2.3 Discussion 92

6.21 Solution #21: Periodicity and TTNB with the lone PDU 92

6.21.1 Key Issue mapping 92

6.21.2 Description 92

6.21.3 Conclusion 93

6.22 Solution #22: Guidelines for PDU Set Marking in Multiplexing Scenarios 94

6.22.1 Key Issue mapping 94

6.22.2 Description 94

6.22.2.1 General 94

6.22.2.2 Unmarked packet handling 95

6.22.2.3 RTP HE for multiplexed content 95

6.22.3 Proposal 98

6.23 Solution #23: PDU Set Size and Over Provisioning in RTP HE for PDU Set Marking 98

6.23.1 Key Issue mapping 98

6.23.2 Description 98

6.23.2.1 General 98

6.23.2.2 Usage of PDU Set Size in NG-RAN 99

6.23.2.3 Discussion 100

6.23.3 Proposal 100

6.24 Solution #24: Traffic pattern prediction for real-time video communication 100

6.24.1 Key issue mapping 100

6.24.2 Description 100

6.24.2.1 The need for traffic pattern indication 100

6.24.2.2 Time to the next data burst (TTNB) 101

6.24.2.3 Burst size 102

6.25 Solution #25: Definition of the time to the next data burst 104

6.25.1 Key Issue mapping 104

6.25.2 Possible definitions and a comparison 104

6.26 Solution #26: RTP HE Enhancements for Data Boosting Indication 105

6.26.1 Key Issue mapping 105

6.26.2 Description 105

6.26.3 Conclusion 106

6.27 Solution #27: Conveying AL-FEC information to the network 106

6.27.1 Key Issue mapping 106

6.27.2 Description 106

6.28 Solution #28: Control-Plane Solution to the Key Issue on Enhancements of Data Burst Marking 107

6.28.1 Key Issue mapping 107

6.28.2 Description 107

6.28.3 Proposed solution 110

7 Conclusions 111

7.1 Key issues progress 111

7.2 Conclusions for Key Issue #1 112

7.3 Conclusions for Key Issue #2 113

7.4 Conclusions for Key Issue #3 113

7.5 Conclusions for Key Issue #4 113

7.6 Conclusions for Key Issue #5 114

7.7 Conclusions for Key Issue #6 114

7.8 Conclusions for Key Issue #7 114

7.9 Conclusions for Key Issue #8 114

7.10 Conclusions for Key Issue #9 114

7.11 Conclusions for Key Issue #10 114

7.12 Conclusions for Key Issue #11 114

7.13 Conclusions for Key Issue #12 115

7.14 Conclusions for Key Issue #13 115

7.15 Conclusions for Key Issue #14 115

7.16 Conclusions for Key Issue #15 115

# Foreword

This Technical Report 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.

# 1 Scope

The Technical Report focuses on optimizing the use of RTP for the transport of real-time XR media (including conversational media) and associated metadata. The use of the IMS Data Channel is supported by existing services such as MTSI but is outside the scope of this report. Aspects related to QUIC are outside the scope of this report.

# 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] 3GPP TS 26.522: "5G Real-time Media Transport Protocol Configurations".

[3] 3GPP TS 23.501: "System architecture for the 5G System (5GS)".

[4] IETF RFC 8872: "Guidelines for Using the Multiplexing Features of RTP to Support Multiple Media Streams".

[5] IETF RFC 5761: "Multiplexing RTP Data and Control Packets on a Single Port".

[6] 3GPP TR 23.700-70: "Study on architecture enhancement for Extended Reality and Media service (XRM); Phase 2".

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

[8] IETF RFC 3711: "The Secure Real-time Transport Protocol (SRTP)".

[9] IETF RFC 9335: "Completely Encrypting RTP Header Extensions and Contributing Sources".

[10] IETF RFC 6904 (2013): "Encryption of Header Extensions in the Secure Real-time Transport Protocol (SRTP)", J. Lennox.

[11] IETF RFC 8402 (2018): "Segment Routing Architecture".

[12] IETF RFC 791 (1981): "Internet Protocol".

[13] IETF RFC 5109: "RTP Payload Format for Generic Forward Error Correction (ULP FEC): Uneven Level Protection, different redundancies for different packets with different importance".

[14] IETF RFC 8627: "RTP Payload Format for Flexible Forward Error Correction (Flex FEC): flexible FEC".

[15] IETF RFC 6681: "Raptor Forward Error Correction (FEC) Schemes for FECFRAME: FEC scheme based on the Raptor".

[16] IETF RFC 6865: "Simple Reed-Solomon Forward Error Correction (FEC) Scheme for FECFRAME: FEC scheme based on Reed-Solomon".

[17] IETF RFC 5053: "Raptor Forward Error Correction Scheme for Object Delivery".

[18] IETF RFC 6330: "RaptorQ Forward Error Correction Scheme for Object Delivery".

[19] IETF RFC 6363: "Forward Error Correction (FEC) Framework"”.

[20] IETF RFC 8854: "WebRTC Forward Error Correction Requirements".

[21] 3GPP TR 38.340: "Study on User Equipment (UE) power saving in NR".

[22] IETF RFC 8298: "Self-Clocked Rate Adaptation for Multimedia".

[23] Enhancing Video Network Resiliency With LTR and RS Code | At Scale Conferences, available online: https://atscaleconference.com/enhancing-video-network-resiliency-with-ltr-and-rs-code/

[24] P. Aggarwal et al., [2304.03732] Enabling immersive experiences in challenging network conditions (arxiv.org)

[25] Nvidia GeForce Now, Video FEC for WebRTC presentation 17 Nov. 2022, available online: https://www.youtube.com/watch?v=igm7QkqxHqk&ab\_channel=KrankyGeek

[26] Holmer S., et al., Handling Packet Loss in WebRTC, 2013 IEEE International Conference on Image Processing, available online: <https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/41611.pdf>.

[27] A Google Congestion Control Algorithm for Real-Time Communication, draft-ietf-rmcat-gcc-02, 2016.

[28] WebRTC source code: https://source.chromium.org/chromium/chromium/src/+/main:third\_party/webrtc, retrieved May 1, 2024.

[29] IETF RFC 8698: "Network-Assisted Dynamic Adaptation: A Unified Congestion Control Scheme for Real-Time Media", 2020.

[30] Self-Clocked Rate Adaptation for Multimedia, draft-johansson-ccwg-rfc8298bis-screamv2-00, 2024.

[31] IETF RFC 4588: "RTP Retransmission Payload Format".

[32] 3GPP TS 26.114: "IP Multimedia Subsystem (IMS); Multimedia telephony; Media handling and interaction".

[33] IETF RFC 4585: " Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)".

[34] IETF RFC 8085: "UDP Usage Guidelines".

[35] IETF RFC 8834: "Media Transport and Use of RTP in WebRTC".

[36] IETF RFC 2250: "RTP Payload Format for MPEG1/MPEG2 Video".

[37] IETF RFC 9143: "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)".

[38] IETF RFC 7941: "RTP Header Extension for the RTP Control Protocol (RTCP) Source Description Items".

[39] GStreamer: open source multimedia framework:  
<https://gstreamer.freedesktop.org/>.

[40] Wowza Streaming Engine™:  
<https://www.wowza.com/>.

[41] Media MTX: Ready-to-use SRT / WebRTC / RTSP / RTMP / LL-HLS media server and media proxy that allows to read, publish, proxy, record and playback video and audio streams:  
<http://github.com/blueviron/mediamtx>.

[42] Adobe RTMP Specification (veriskope.com):  
<https://rtmp.veriskope.com/docs/spec/>.

[43] RTP Extensions for Transport-wide Congestion Control, draft-holmer-rmcat-transport-wide-cc-extensions-01  
<https://datatracker.ietf.org/doc/html/draft-holmer-rmcat-transport-wide-cc-extensions-01>

[44] IETF RFC 3550: "RTP: A Transport Protocol for Real-Time Applications".

[45] IETF RFC 5104: "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)".

[46] IETF RFC 6679: "Explicit Congestion Notification (ECN) for RTP over UDP".

[47] IETF RFC 8888: "RTP Control Protocol (RTCP) Feedback for Congestion Control".

[48] IETF RFC 8382: "Shared Bottleneck Detection for Coupled Congestion Control for RTP Media".

[49] IETF RFC 3611: "RTP Control Protocol Extended Reports (RTCP XR)".

[50] 3GPP TS 26.510: "Media delivery; interactions and APIs for provisioning and media session handling".

[51] Di Domenico, Andrea, et al. "A network analysis on cloud gaming: Stadia, GeForce Now and PS Now." Network 1.3 (2021): 247-260.

[52] 3GPP TR 26.928: "Extended Reality (XR) in 5G".

[53] 3GPP TS 26.506: "5G Real-time Media Communication Architecture (Stage 2)".

[54] 3GPP TS 23.228: "IP Multimedia Subsystem (IMS) - Stage 2".

[55] 3GPP TS 29.571 "5G System; common data types for Service based interfaces; stage 3".

[56] 3GPP TS 38.415 "NG-RAN; PDU Session User Plane Protocol".

[57] Real-Time Transport Protocol (RTP) Parameters (iana.org).

[58] 3GPP TS 26.113: " Real-Time Media Communication; Protocols and APIs".

[59] 3GPP TR 26.922: "Video telephony robustness improvements extensions; Performance evaluation".

[60] 3GPP TR 26.926: "Traffic Models and Quality Evaluation Methods for Media and XR Services in 5G Systems".

[61] 3GPP TS 38.413: "NG-RAN; NG Application Protocol (NGAP)".

# 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.

**Lone PDU:** APDU that is not marked by the sender as part of a PDU Set.

**Multimedia Session:** An association among a group of participants engaged in the communication via one or more RTP sessions, as defined in section 2.2.4 of IETF RFC 7656 [18].

**PDU Set marking:** Marking the PDUs carrying a payload with the PDU Set Information.

**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 Tethered Device:** Device connected indirectly to 5G Network.

## 3.2 Symbols

Void.

## 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].

AL-FEC Application-Layer Forward Error Correction

AVC Advanced Video Coding

CDRX Connected mode discontinuous reception

GCC Google Congestion Control

H.266/VVC ITU H.266/MPEG Versatile Video Coding

HE (RTP) Header Extension

HEVC High Efficiency Video Coding

IMS IP Multimedia Subsystem

IRAP Intra Random Access Picture

MTSI Multimedia Telephony Service for IMS

NADA Network-Assisted Dynamic Adaptation

NAL Network Abstraction Layer

NG-RAN Next Generation Radio Access Network

NPDS Number of PDUs in a PDU Set

NTP Network Time Protocol

OS Operating System

PCC Performance-oriented Congestion Control

PSI PDU Set Importance

PSN PDU Sequence Number within a PDU Set (PSN)

PSSize PDU Set Size

PSSN PDU Set Sequence Number

PTP Precision Time Protocol

RLC Radio Link Control

rPSSize remaining PDU Set Size

RTC Real Time Communication

RTCP XR RTCP eXtended Report

RTCP RTP Control Protocol

SCReAM Self-Clocked Rate Adaptation for Multimedia

SRTP Secure RTP

UDP User Datagram Protocol

UPF User Plane Function

XR eXtended Reality

# 4 Architectural Assumptions and Requirements

## 4.1 Architectural Assumptions

In this report, a Real Time Communication (RTC) architecture for the 5G System [3] is assumed as defined in [53] and illustrated in Figure 4.1-1.

Figure 4.1-1 illustrates the high-level architecture.

Figure 4.1-2 illustrates additional elements of the general RTC Architecture [53].

The media transport RTC-4m, signaling RTC-4s shown in 4.1.1-2 are in scope of this report.

In addition, RTC-5 and RTC-1 are in scope in this report.

Usage in or adoption in other architectures is in no way precluded. In particular, it is expected that the real time communication can be deployed in the IP Multimedia Subsystem (IMS) [54].



Figure 4.1-1: Real-time media communication (RTC) in 5G System [53]



Figure 4.1-2: RTC General Architecture in the 5G System [53]

## 4.2 Architectural Requirements

Uplink (UL), i.e. UE via RAN and UPF to RTC-AS, is in scope.

Downlink (DL), i.e. AS to UE via UPF and RAN, is also in scope.

The architecture as defined in TS 23.501 [3] is assumed for the 5G System.

The architecture from TS 26.506 [53] serves as example for deploying real-time communication in the 5G System.

The report targets improving the end-to-end performance and other performance aspects for different services and use cases.

Enhancements targeting TS 26.522 [2] are in scope, but improvements may also target other specifications that implement reference points in scope of this report.

Changes in behaviour of RAN or UPF based components are considered in consultation with respective working groups in 3GPP.

# 5 Key Issues

## 5.1 Key Issue #1: Inaccuracy of the PDU Set Size (PSSize) information

### 5.1.1 Description

This key issue follows up on the work on signaling the PDU Set Size in the RTP Header Extension for PDU Set marking in [2]. A solution was proposed in Release 18 in 5G RTP work item and included into TS 26.522 [2] to enable accurate setting of the PDU Set Size field. However, several issues related to the end-end transmission were raised that may change the size of the PDU Set Size. These issues were not adequately addressed in Release 18 and to enable the practical use of the solution adopted in Release 18, there are needs for clarifications and/or refinements.

This key issue focuses on the PDU Set Size signaling and several end-end transmission aspects. The following objectives have been identified:

- Study and document the causes of PDU Set Size deviations in end-to-end transmission, such as NAT46/64 traversal [2], TURN, STUN use of IP Header Options, Internet Protocol fragmentation, and/or Segment Routing [11].

- Study and quantify the deviation percentages in practical RTC scenarios resulting from the causes.

- Study what type of accuracy is needed from a RAN scheduling perspective, as the PDU Set Size is mainly used to optimize the scheduling at the RAN, take their feedback into account.

NOTE 1: This point may require coordination with RAN2.

- Study and develop solutions to improve 5G RTP workflows using PDU Set marking taking such feedback into account.

- Discuss the overhead and added complexity that can be tolerated to support this feature.

NOTE 2: Coordination with RAN2 and SA2 might be needed.

## 5.2 Key Issue #2: QoS handling requirements for lone PDU

### 5.2.1 Description

In the FS\_5G\_RTP\_Ph2, one objective was to study ‘issues around "lone" PDU, as identified by SA2’.

In an LS from SA2, SA4 was asked the following questions:

*SA2 in Rel-18 has agreed that the PSA UPF marks, in the downlink, each N6-unmarked PDU (lone PDU) with PDU Set information into a PDU Set over N3/N9. As a consequence, RAN will apply the PDU Set QoS parameters, e.g. apply the PDU Set Delay Budget (which is assumed to be larger than the PDB, if applicable) for the lone PDU.*

*Questions: Will applying PDU Set QoS parameters to these lone PDUs pose any issue from application perspective? If yes, what is the issue?*

*SA2 will not change the agreement to map N6-unmarked PDUs to PDU Sets over N3/N9 in Rel-18. However, since this topic may be in the scope of the FS\_XRM\_Ph2 study, SA2 would like to get feedback from SA4 on the questions above.*

For a single PDU which doesn't belong to any PDU Set, the 5GS handles such lone PDU as a single PDU Set following the PDU Set QoS parameters. Furthermore, a lone PDU does not carry the RTP header extension for PDU Set marking defined in TS 26.522 [2] and thus cannot convey any PDU Set Information to the 5GS. It’s proposed to study:

- Whether there is any issue when applying PDU Set QoS parameters to the lone PDUs from the application layer perspective.

- Study and detail the scenarios when such lone PDU may arise, e.g., RTP/RTCP multiplexing, unmarked packets, incomplete sender implementation, existing guidelines for PDU Set marking.

- Develop solutions to handle the issue of missing PDU Set Information in case of lone PDUs.

- Study the impact on applications of marking versus not marking of lone PDUs into PDU Sets.

NOTE: Both the marking in DL direction and at the UE for uplink direction are considered.

## 5.3 Key Issue #3: Enhancements for application-layer FEC support

### 5.3.1 Description

Commercial adoptions may use application layer FEC (AL-FEC) as documented in clause 5.7.4 of TR 26.926 [60]. In RTC AL-FEC may optionally be used, but the usage is currently not documented. The objective of this key issue is to:

- study and summarize the AL-FEC schemes that may be used as available in IETF standards and also the status of identified commercial deployments. A summary and categorization based on different aspects of the implementation such as complexity, arbitrary loss resilience, keeping the source stream unaltered will be studied. In addition, other potential gaps may be identified.

- recommend adoption of one or more FEC schemes in 3GPP specifications for specific use cases such as split rendering, in case a clear benefit and a path forward is identified by the group for these use cases.

NOTE 1: The outcome of this key issue is expected to be shared in communication with SA2 to inform them about potential usage of AL-FEC in the RTC solutions developed by SA4 (and referenced by SA2).

NOTE 2: The outcome of this key issue is expected to be the basis for developing solutions for FEC awareness for PDU Set handling in Key Issue #4.

## 5.4 Key Issue #4: AL-FEC awareness for PDU Set handling

### 5.4.1 Description

The application layer FEC mechanisms are widely used to improve packet transmission robustness in the presence of packet losses without going through packet retransmissions that can introduce delays that violate real-time constraints. In the draft TR 23.700-70 [6] of FS\_XRM\_Ph2, Key Issue #1 is proposed to study the enhancement of PDU Set handling including Application-Layer Forward Error Correction (AL-FEC) encoded PDU Sets, and specifically:

*whether, what and how PDU Set based handling (e.g. new standardized 5QI, enhancements to Alternative QoS profiles, FEC, etc.) and PDU Set information (including Control Plane and/or User plane information) provided by the AF/AS are enhanced.*

To provide some background information, the basic idea is to expose AL-FEC related information to the NG-RAN via the control plane or user plane. The AL-FEC related information could be redundancy information or markings to differentiate among source and repair PDUs of a PDU Set etc. Based on the AL-FEC awareness, the NG-RAN may optimize the PDU Set delivery over the air interface accordingly (e.g., by discarding redundant PDUs of AL-FEC encoded PDU Sets). In the context of this cross-layer design, it is important to understand how to expose the AL-FEC information to the communication network (UPF, RAN) to enable intelligent resource allocation. Furthermore, there are intricate interactions to consider between the application and the network. In particular, dropping extra PDUs of a PDU Set encoded with AL-FEC in a network, if any, may send a false signal to the application on the packet loss rate and the congestion level in the network. This may lead to undesired adaptation from the application such as increased redundancy ratio and reduced sending rate that can negatively impact end-to-end performance and user experience. It is thus important to understand the interactions between the application and the network in the case of AL-FEC and intentional packet dropping by the network and the impact on the media performance.

Therefore, it’s proposed to study:

- Benefits of AL-FEC awareness for PDU Set handling given application and network interactions in the context of 3GPP, if any;

- Whether and how to assist the 5GS to get aware of the AL-FEC;

- For AL-FEC awareness for PDU Set handling, how to avoid/minimize the impact to the application layer, if any;

- Benefits to the user experience and application end-to-end performance when enabling AL-FEC-awareness.

## 5.5 Key Issue #5: RTP transport of XR metadata

### 5.5.1 Description

This key issue was not progressed.

## 5.6 Key Issue #6: PDU Set marking for XR streams with RTP end-to-end encryption

### 5.6.1 Description

The usage of end-to-end encryption is broadly deployed in current networks to provide security. Similarly, secured deployments are expected for 5G RTP applications.

In this report, end-to-end encryption is referred to encryption that is commonly used in the industry that aims at the situation where only the two end users can access the confidential information but parties in between cannot.

Confidentiality is defined in this case as all user-related information being kept secret. This means that user-related information from endpoint A to endpoint B is kept secret from other entities. A 5G RTP end-to-end encrypted data flow contains RTP PDUs whose SDUs are encrypted, and headers may be partly encrypted.

Certain metadata not related to the information exchanged between the two parties need not be encrypted in this case. This follows industry best practices. For this issue the focus is expected to be on the aspects within the scope of the report relating to XR media delivery.

This key issue proposes to study the enhancement of PDU Set Identification in encrypted RTP streams, in particular when using the RTP HE for PDU Set marking.

The key issue studied the following aspects:

- Explore and document the different scenarios for providing end-to-end RTP encryption as targeted for 5G RTP

- If and how PDU Set information Identification may happen in an end-to-end encryption scenario for 5G RTP.

- If needed, develop methods for signaling PDU Set Information for end-to-end encrypted RTP streams applicable to different methods of end-to-end encryption.

NOTE 1: Solutions that rely on breaking end-to-end encryption are out of the scope of this key issue.

NOTE 2: The work on this key issue may need coordination with SA WG2 and WG3.

NOTE 3: The end-to-end encryption based on QUIC is out of scope of this report.

## 5.7 Key Issue #7: Existing RTCP messages and RTP header extensions to better support XR services in 5G

### 5.7.1 Description

Most existing RTCP messages were designed to provide feedback on RTP-based delivery of the conventional conversational media. It is not clear whether they are still suitable for delivering the new XR media, such as the media in split-rendering XR, which generally require lower latency and higher throughput and involve the transmission of metadata. It is important to examine the life cycle of typical and emerging XR applications and identify potential gaps in the RTCP messages.

Also, RTP header extensions may provide feedback on RTP-based delivery of both conventional media and XR media. It is beneficial to list the RTP header extensions that may be important to the application layer adaptation to the network condition and the computing resources.

The objectives of this key issue are to:

- List candidate RTCP messages that are essential to providing feedback on RTP-based delivery of the conventional converstional media and the new XR media

- Identify potential gaps in the existing RTCP messages

- List candidate RTP header extensions that are essential to providing feedback on RTP-based delivery of the conventional conversational media and the new XR media

- Identify potential gaps in the existing RTP header extensions

NOTE: The scope for the candidate RTCP messages and RTP header extensions is expected to cover IETF, WebRTC implementation [28] and 3GPP specifications.

## 5.8 Key Issue #8: RTP retransmission in supporting XR services in 5G

### 5.8.1 Description

RTP retransmission defined by IETF in RFC 4588 [31] is one of the media resilience techniques adopted in TS 26.114 [32] used to compensate packet losses for real-time media. Since retransmissions result in additional delay, the feasibility of RTP retransmission in XR applications subject to tight delay bounds needs to be investigated.

While application-layer retransmissions may be necessary for ensuring reliable data delivery in some cases, their usage may violate assumptions at the network layer such as 5GS QoS handling. Therefore, the usage of RTP retransmission when using 5GS QoS handling needs to be studied. In addition, the 5GS QoS handling is currently unable to distinguish retransmissions from original transmissions and thus cannot handle retransmission PDUs in a differentiated way. It needs to be understood how 5GS QoS handling, including PDU set based QoS Handling is affected when RTP retransmission is used and if there are potential benefits of retransmission awareness to 5GS QoS Handling.

It is therefore proposed to study:

- the feasibility of RTP retransmission for XR media services

- whether and how RTP retransmission can be combined with 5GS QoS handling,

- if and how awareness of RTP retransmission can benefit PDU Set based QoS handling in the network.

- whether and how PDU Set related information can be used to improve RTP retransmission.

NOTE: Work on this key issue may need coordination with SA2.

## 5.9 Key Issue #9: Feasibility of RTP multiplexing options for transport of XR media streams

### 5.9.1 Description

RTP originally relied on UDP multiplexing for carriage of different media streams (using different UDP ports for each stream and RTCP).

However, in practice assigning additional UDP ports has been problematic, and RTP based multiplexing is used as an alternative.

For RTP multiplexing of streams in a single RTP session, the SSRC is generally used as described in RFC 8872 [4].

In addition, combining RTCP and RTP on the same port is referred to as RTP/RTCP multiplexing.

In this case, the RTP and RTCP traffic can be multiplexed and demultiplexed using the shared second Byte of the UDP payload (i.e., the RTCP packet type and the RTP M bit & RTP payload type) as described in RFC 5761 [5].

In WebRTC the same port may be used for RTP, RTCP and different streams.

In addition, other forms of multiplexing may be used to support carriage of different streams over RTP that have been popular in the media industry (e.g. MPEG-2 based multiplexing).

For example, in MPEG-2 TS over RTP [36] the small TS packets of 188 bytes can be interleaved in an RTP packet. So, in this case an RTP packet contains multiple small transport stream packets that could be audio, video or even metadata related to a television program. This would lead to RTP packets including multiple media types.

It is proposed to:

- Study and document existing options for RTP multiplexing.

- Identify the potential gaps on support of different use cases.

- Identify and document other popular ways of supporting multiplexed content in RTP transmission if any.

- Study and identify how multiplexed RTP can benefit from PDU Set marking header extensions.

- Study the relevance of identifying multiplexed streams in 5GS and explore potential benefits of additional support in 5GS.

NOTE: This issue may require coordination with SA2. Additional support in 5GS refers to the multiplexed traffic detection and QoS Flow mapping in FS\_XRM\_Ph2.

## 5.10 Key Issue #10: Use cases and intended deployment scenarios for enhancements of RTP header extension for PDU Set marking

### 5.10.1 Description

This key issue was not progressed.

## 5.11 Key Issue #11: Enhancements of RTP header extension for PDU Set marking

### 5.11.1 Description

This key issue was not progressed.

## 5.12 Key Issue #12: Enhancements of Data Burst Marking

### 5.12.1 Description

A data burst indicates a set of multiple PDUs generated and sent in a short period of time as defined in clause 3.1 of TS 23.501 [3]. Data burst is a common transmission characteristic in communication networks.

The traffic characteristics regarding the data burst transmission could be beneficial for the 5GS network, e.g., power saving and efficient radio resource management. In Release 18, the End of Data Burst indication has been introduced to enable the UE power saving in the 5GS, i.e., the NG-RAN node can configure to move a UE into CDRX for power saving after transmitting the end PDU of the data burst.

Similarly, as stated in the draft TR 23.700-70 [6] from SA2 Rel-19 FS\_XRM\_Ph2, it also studied the 5GS network enhancements to support the burst related traffic characteristics.

Therefore, the following enhancements to data bust marking are studied in this key issue.

- Identify additional traffic characteristics beneficial to the 5GS network, for example, time to next burst, burst size and other potentially relevant characteristics

- Identify and document the way RTP senders can generate data bursts e.g.

- WebRTC paced sending implementation, including the different configurations of WebRTC paced sending.

- Other common RTP implementations or libraries that are commonly used

- Develop potentially additional signaling in the 5G RTP Header Extension to include additional traffic characteristics.

- Develop guidelines and recommendations for the setting of traffic characteristics related parameters in the RTP Header Extension (if needed).

- Potential backward compatibility issues need to be considered.

## 5.13 Key Issue #13: Applicability of the RTP header extension for PDU Set marking to different PDU Set types

### 5.13.1 Description

In the Rel-18 work, it was mainly assumed in TS 26.522 [2] that the PDU Set framework is applied to PDU Sets comprising either video frames or slices. However, the PDU Set definition in TS 23.501 [3] does not limit a PDU Set to be a video frame or slice.

***PDU Set:*** *One or more PDUs carrying the payload of one unit of information generated at the application level (e.g., frame(s) or video slice(s) etc. for eXtended Reality (XR) Services). All the PDUs of a PDU set are transmitted within the same QoS Flow.*

The objective of this key issue is to study the applicability of the PDU Set concept for the cases where the PDU Set is not a video frame or slice.

This key issue aims at addressing the following points:

- Study and document applicability criteria of PDU Set marking to different media types and formats.

- Whether and how to apply PDU Set marking to non-video data: metadata, audio, text, image.

- Whether and how to apply PDU Set marking to picture partitioning constructs other than slices such as tiles.

## 5.14 Key Issue #14: Traffic detection and QoS flow mapping for multiplexed media stream data flows

### 5.14.1 Description

RTP allows different delivery options for multiple media streams. The media streams can be transmitted as multiple RTP streams in a single RTP session, in multiple RTP sessions, or in some cases, multiplexed media can be carried in a single RTP stream. Hence, in some cases, a media stream may be split into multiple QoS flows or multiple media streams may be multiplexed into a single QoS flow. It is therefore important to study how the UPF and RAN nodes can identify the PDU sets belonging to a specific media stream in a PDU session in the case of multiplexed media streams.

In RTP, different streams typically use different multiplexing methods for the delivery of the media streams. Given that a QoS flow is composed of PDUs from multiple media streams, the traffic over one QoS flow will be a mix of traffic from different media streams. PDU Sets arriving at the UPF and RAN nodes are from different streams, and the RAN nodes needs to identify the respective media streams to which they belong.

In addition, some mappings of streams to QoS flows, may result in media streams that are split across one or more QoS flows. When media stream data is split across multiple QoS flows, then some PDU Sets of the stream may go over one QoS flow and some may go over other QoS flow. Therefore, the UPF and RAN nodes needs to handle PDUs arriving at the UPF and NG RAN with missing PDU Sets in a specific QoS flow. For example, the RAN nodes need to deal with gaps in the PDU Set Sequence Number (PSSN) for a stream in a QoS flow.

It is proposed to:

- study and document the issues arising due to multiplexing multiple media streams into a single QoS flow or splitting a media stream across multiple QoS flows.

- determine benefits for identifying the PDU Sets belonging to a media stream split over multiple QoS flows.

- provide solutions on how to identify different PDU Sets from the individual streams at UPF and RAN nodes. and how to handle missing PDU Sets in a QoS flow when stream splitting is performed.

## 5.15 Key Issue #15: Media and metadata delivery over multiple sessions

### 5.15.1 Description

In XR communication, certain media and metadata types, e.g., avatar and associated animation data, can be transmitted over a WebRTC or IMS data channel.

At the same time, it may still be possible to have a UE-to-UE voice call, e.g., an MTSI call, as the required latency for voice may be lower.

SDP procedures take care of grouping appropriate media flows for synchronization and other functionalities within the same RTP session. However, the synchronization between and RTP-based and a non RTP-based media/metadata delivery requires further investigation. It needs to be studied how this can be achieved effectively.

In addition, the case when media streams and metadata are delivered over different RTP sessions needs study.

Another use case where associated media may be sent over different RTP sessions is teleconferencing applications. The voice in this case maybe over a direct UE-to-UE communication (MTSI call), while other media (e.g., presentations, video) are delivered via a network media function. A high-level illustration is shown in Figure 5.15.1-1 below. Here the voice is delivered UE-to-UE, and the associated RTP session is shown as Session 3. The video from UE A to UE B is delivered via a network media function over two RTP sessions, Session 1 and Session 2. Depending on the use case and application requirements, the network media function may apply operations such as upscaling, merging video streams, or animation in case of avatar data.

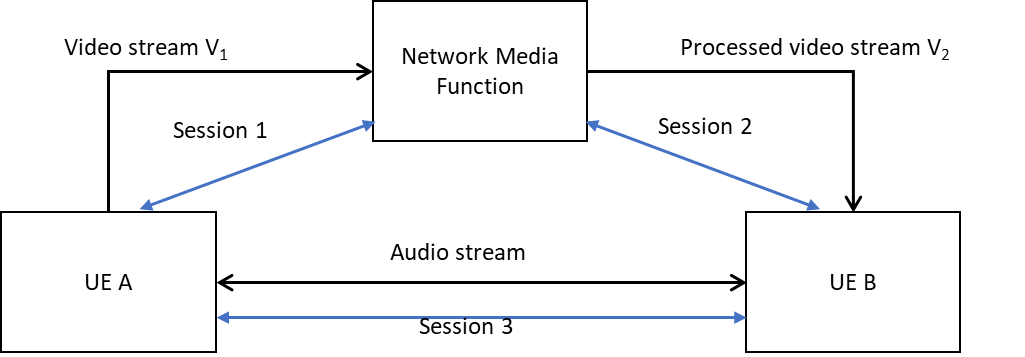


Figure 5.15.1-1: An example scenario with multiple media sessions.

In this key issue, it is proposed to study:

- Whether it is feasible (in terms of typical RTC use cases) to have media/metadata components that are sent over different paths, e.g., a UE-to-UE voice channel and a UE-MF-UE or AS/MF-to-UE channel for avatar data (sans audio).

- Identify synchronization issues, if any.

- Identify session establishment issues, if any.

- How to achieve cross-session referencing for XR media and metadata that are sent over different RTP sessions and data channels that don’t have common endpoints.

- E.g. SDP signaling description examples.

# 6 Solutions

## 6.0 Mapping of Solutions to Key Issues

Table 6.0-1: Mapping of Solutions to Key Issues

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Solutions |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | KI#1 | KI#2 | KI#3 | KI#4 | KI#5 | KI#6 | KI#7 | KI#8 | KI#9 | KI#10 | KI#11 | KI#12 | KI#13 | KI#14 | KI#15 |
| #1 |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |
| #2 |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |
| #3 |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |
| #4 | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| #5 |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |
| #6 |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |
| #7 | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| #8 |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |
| #9 |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |
| #10 |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |
| #11 |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |
| #12 |  |  |  |  |  |  |  |  | X |  |  |  |  | X |  |
| #13 |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |
| #14 |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |
| #15 |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |
| #16 |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |
| #17 |  |  | X | X |  |  |  |  |  |  |  |  |  |  |  |
| #18 |  |  | X | X |  |  |  |  |  |  |  |  |  |  |  |
| #19 |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |
| #20 |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |
| #21 |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |
| #22 |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |
| #23 | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| #24 |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |
| #25 |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |
| #26 |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |
| #27 |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |
| #28 |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |

## 6.1 Solution #1: Different PDU Set types to support handling of immersive media

### 6.1.1 Key Issue mapping

This solution addresses key issue #13 on the Applicability of the RTP header extension for PDU Set marking to different PDU Set types.

### 6.1.2 Description

In addition to frames and slices, video PDU Sets can be defined as other sub-picture constructs such as H.265/HEVC tiles.

Such PDU Sets are particularly relevant for immersive media use cases, where it could be beneficial to assign different importance to different regions of a picture, for example depending on content saliency and user’s viewing direction, as in the example of 360-degree video streaming. However, as the usage of tiles is not limited to 360-degree video streaming, the solution may also apply to other kinds of immersive media, such as volumetric video.

This allows a dynamic spatial adaptation of the PSI based on user’s viewport and/or other content-related metadata. Depending on the selected criteria (e.g., the user’s actual viewport), the sender can dynamically modify the PSI for each tile at different time points. Alternatively, the sender may consider a group of tiles as a PDU Set to reduce the number of PDU Sets contained within a picture.

The sender may determine the PSI for the PDU Sets belonging to a region based on the likelihood of that region corresponding to the user’s viewport. For example, if a tile is likely to fall outside the user's viewport, it is assigned a higher PSI for that time point. Hence, it is more likely to be dropped in case of congestion. Later, if it becomes more likely to fall inside the user's viewport, it is assigned a lower PSI.

The figure below shows an example where the left picture is divided into 12x6 tiles and the right picture into 8x4. The blue color shows the current viewport region, and the grey colour shows the background region. The PSI values assigned by the sender for a given time point are shown inside each tile. When the picture is split into a greater number of tiles (left), the sender may set the viewport center with higher importance and gradually reduce the importance as the distance increases from the viewport center. PSI values for the tiles on the boundaries may be decided depending on the portion of the tile that falls into the viewport. In the example shown on the left, the sender assigns PSI=5, if more than 50% of a tile falls inside the viewport, and PSI=6, if a tile overlaps with the viewport but less than 50% of it falls inside the viewport. Also, the sender may distinguish between the background tiles that are outside but close to the viewport (set to PSI=10 in the example) and the other background tiles that are far from the viewport (set to PSI=15 in the example). With a smaller number of tiles, 8x4 as in the example on the right, the sender can perform the PSI assignment in a coarser manner.

A close-up of a number

Description automatically generated

Figure 6.1.2-1. Example tiling and PSI assignment.

In the current PDU Set handling framework, the network is unaware of what application data unit is marked as a PDU Set by the sender. When sub-picture constructs such as tiles are marked as PDU Sets, it may be beneficial for the network to be aware of the PDU Set type/structure used by the application to adequately configure the scheduling or the PSI-based discard, if it is used. For example, if the network knows that PDU Sets are tiles, the RAN may decide to apply a more relaxed discard mechanism to PDU Sets, knowing that it is not as critical as discarding an entire frame. The PDU Set type can be indicated by the application in the RTP header extension for PDU Set marking or via control plane signaling, for example as an extension to the Protocol Description provided by the AF.

The solution is also suitable for the next generation codec (H.266/VVC) that makes use of a more flexible partitioning mechanism called sub-pictures.

## 6.2 Solution #2: Gap analysis on the QoS requirements for lone PDUs

### 6.2.1 Key Issue mapping

This solution intends to give gap analysis on the KI#2: QoS handling requirements for lone PDUs.

### 6.2.2 Description

According to TS 23.501 [3], in case a single PDU doesn't belong to a PDU Set based on the Protocol Description for PDU Set identification, the UPF still maps it to a PDU Set and determines the PDU Set Information accordingly. In this case, both the lone PDU and the PDUs belonging to a PDU Set are in the same service data flow and the lone PDU is delivered to the UE in the DL direction following the PDU Set QoS parameters.

There could be different scenarios where the application server may send the PDU Sets and lone PDUs in the same service data flow which can be detected by the 5GS. For video data, as described in Annex A.2.2.1 of TS 26.522 [2], it is generally recommended that the network function considers non-VCL NAL units as part of the PDU Set of the associated VCL NALUs, e.g. identified by the same timestamp. Once the RTP header extension for PDU Set has been negotiated between the RTP sender and receiver, the RTP sender marks each packet with RTP HE for PDU Set marking. However, there are other scenarios where lone PDUs and PDUs belonging to a PDU Set are multiplexed in a single service data flow as following.

- **Scenario #A:** RTP streams multiplexed in a single RTP session. In this scenario, multiple RTP streams are multiplexed in a single RTP session which is carried over a single service data flow. For example, the audio and video streams are multiplexed in a single RTP session, while the PDU Set handling is needed only for the video streams. Similarly, when FEC or RTP retransmission feature is enabled, the corresponding repair packets or retransmission packets may also be multiplexed with the original video stream. As of Rel-18, the 5GS cannot distinguish different RTP streams multiplexed in a single service data flow and has to treat the PDUs in other RTP streams as lone PDUs.

- **Scenario #B**: RTP data and control packets are multiplexed on a single port. In this scenario, the RTP and RTCP flows are carried over a single service data flow. When the PDU Set handling is needed for the RTP flow(s), the 5GS has to treat the RTCP traffic as lone PDUs since it cannot distinguish between the RTP and RTCP traffic.

NOTE: A combination of scenario #A and #B is possible.

As can be seen from the above, one key reason for the lone PDU handling is that the PDUs belonging to a PDU Set and the lone PDUs are carried over a single service data flow and as of Rel-18, therefore, the 5GS cannot differentiate the multiplexed data flows in a single service data flow.

Therefore, it is clear that

- **Coexistence of lone PDUs and PDUs belonging to a PDU Set in a single service data flow can be due to the lack of the capability to differentiate multiplexed media flows in 5GS.**

NOTE: This solution mainly focuses on the scenario where the lone PDUs are resulted from the missing capability of multiplexed traffic identification.

However, the scenario where lone PDUs may exist, is still possible due to the multiplexed RTP and RTCP or RTP audio and video traffic flows. As the streams are in a single QoS Flow as requested by the application layer, e.g., the QoS requirements for them have to be the same.

However, the QoS requirements for multiplexed media streams could be different. For example, the QoS requirements for audio and video streams could be different. In Release 19, limited support for mapping multiplexed traffic flows was added and this is studied in KI #14.

For PDU Set based QoS handling, the PDU Set QoS parameters are introduced in TS 23.501 [3] as following:

- PDU Set Delay Budget, which defines an upper bound for the delay that a PDU Set may experience for the transfer between the UE and the N6 termination point at the UPF.

- PDU Set Error Rate, which defines an upper bound for the rate of PDU Sets that have been processed by the sender of a link layer protocol (e.g., RLC in RAN of a 3GPP access) but that are not successfully delivered by the corresponding receiver to the upper layer (e.g., PDCP in RAN of a 3GPP access).

- PDU Set Integrated Handling Information, which indicates whether all PDUs of the PDU Set are needed for the usage of the PDU Set by the application layer in the receiver side.

If the NG-RAN receives PDU Set QoS Parameters, it enables the PDU Set based QoS handling and applies PDU Set QoS Parameters. When the PDU Set QoS parameters are available, they will supersede the PDU QoS parameters (i.e. PSDB/PSER supersedes the PDB/PER).

For the corresponding PDU QoS parameters, they are at a per packet granularity including the per-packet latency requirement (i.e. packet delay budget), the per-packet loss rate requirement (i.e. packet loss rate), etc. From the application perspective, the PDU Set QoS parameters and the PDU QoS parameters need to reflect the same network requirements while at different granularities.

- When an RTP video stream and an RTP audio stream are multiplexed in a single RTP session and the PDU Set based QoS handling is enabled for the RTP video stream, the PDU Set QoS parameters can indicate the delay and reliability requirements for the video PDU Set (e.g. a video frame/slice), which are also applied to the audio PDU Set (typically an audio frame carried in a single audio packet). In this case, applying the PDU Set QoS parameters to the lone audio PDUs is totally fine.

- When RTCP traffic and RTP video stream are multiplexed using a single UDP port, the same PDU Set QoS parameters could then be applied to the RTCP packets and the video PDU Sets (e.g. video frame/slice), assuming they are mapped into the same QoS flow. This is expected to be the case when RTCP traffic is used to measure the network characteristics (e.g. round-trip time).

Therefore, **QoS requirements for lone PDUs and marked PDU Sets could be the same and** **applying the PDU Set QoS parameters to a single PDU could be no problem. However, QoS requirements for lone PDUs and marked PDU Sets may be different and an issue. This depends whether the lone PDUs requires (or can sustain) the same QoS requirements as the PDU Sets.**

The solution to KI#4 in TR 23.700-70 [6] enables the network to differentiate multiplexed streams sent in the same media transport such that they can be mapped into distinct QoS flows. However, in some cases this may result in unintended behavior, e.g. RTCP packets mapped to a different QoS flow would no longer measure the RTP media QoS flow characteristics which may result in errors e.g. in measuring the media flow characteristics. On the other hand, it could be problematic to apply the PDU Set QoS to lone PDUs, as described above.

NOTE: Other measurement methods may be used instead in this case and it is up to the application whether to request differentiated QoS handling for the RTP and RTCP traffic.

In addition, as discussed in draft TR 23.700-70 [6], how to support the traffic detection and QoS mapping for multiplexed data flows is ongoing in SA2 Rel-19 FS\_XRM\_Ph2 as shown below:

*This key issue proposes study traffic detection and QoS Flow mapping in 5GS for different media streams multiplexed within a single end-to-end transport connection.*

*- How to identify multiplexed traffic flows with different QoS requirements within a single transport connection.*

*- How to do QoS Flow mapping for traffic flows with different QoS requirements.*

*- Whether and what information needs to be provided from AF for traffic detection.*

*- Whether and how AF provides QoS requirements of different traffic flows to the 5GS.*

Via the potential R19 enhancements in 5GS, it is possible to differentiate the multiplexed RTP streams or RTP/RTCP flows, which may avoid the co-existence of lone PDUs and PDUs belonging to a PDU Set in a QoS flow.

As concluded in clause 8.4 in TR 23.700-70 [6], the application layer may ask the 5G system to differentiate the different RTP/RTCP streams in one RTP session with the extended packet filter set. The extended packet filter includes the legacy IP packet filter set as defined in clause 5.7.6 of TS 23.501 [3] and also the additional packet filter to detect the multiplexed traffic and map them into different QoS requirements as requested by the AF. This additional packet filter may contain the RTP-SSRC, etc.

In case that the RTP/RTCP streams are multiplexed in an RTP session and one RTP stream needs the PDU Set based QoS handling, the legacy packet filter set together with the corresponding SSRC(s) can be used to detect the target RTP stream(s) and map to the QoS Flow with PDU Set QoS requirements. Therefore, the lone PDU issue resulted from the multiplexing could be avoided considering the additional support in 5GS in SA2 FS\_XRM\_Ph2 if the application requests different QoS handling for different multiplexed media flows.

Hence, it’s proposed that the RTC AF further provides the RTP-SSRC(s) to the 5GS if the media streams with RTP HE for PDU Set marking enabled requires the PDU Set based QoS handling. Then the 5GS can differentiate the RTP streams with RTP HE for PDU Set marking and other traffic in order to avoid the lone PDUs that would arise due to multiplexing.

NOTE: Impact to the RTC architecture in TS 26.506 needs to be considered during the normative work phase for Key Issue #14.

### 6.2.3 Conclusion

Based on the gap analysis in the above, it is proposed to make the following conclusions.

**- QoS requirements for lone PDUs and marked PDU Sets could be the same and** **applying the PDU Set QoS parameters to a single PDU could be no problem.**

- **In case the QoS requirements for the lone PDUs and the marked PDU Sets are different, this could be an issue. Such use cases still need further study.**

NOTE: Further coordination with SA2 may be necessary regarding potential normative solution in this case.

**- Communicate with SA2 if needed to reply the question raised by SA2.**

## 6.3 Solution #3: SRTP Usage for end-to-end encryption

### 6.3.1 Key Issue mapping

Solution to key issue number #6 PDU Set marking for XR streams with RTP end-to-end encryption using SRTP.

### 6.3.2 Description

When end-to-end encryption is applied, methods of inspecting the video bitstream will not work for PDU Set detection and NAL syntax cannot be read, neither.

Besides, when the RTP is tunnelled over an end-to-end encrypted channel, the method of RTP header extension for PDU Set Marking in TS 26.522 [2] will not work, neither.

As the RTP header extension (HE) for PDU Set marking uses the general mechanism for RTP Header Extensions from RFC 8285 [7], it is not encrypted in secure RTP solution RFC 3711 [8]. Therefore, the SRTP could potentially be used together with RTP HE for PDU Set marking.

In Release 18 of TS 26.522 [2] SRTP is supported, so this solution requires limited changes to TS 26.522 [2] but some explicit text.

NOTE: Some cases when the RTP HE is also encrypted, e.g., RFC 6904 [10], RFC 9335 [9], are FFS.

## 6.4 Solution #4: Measurement Based Pre-compensation for PDU Set Size Correction

### 6.4.1 Key Issue mapping

This solution maps to Key issue #1.

### 6.4.2 Description

As discussed in 5.1.1, there are multiple reasons for causing the PSSize to be inaccurate. Although the impact of those reasons on the size of a single IP packet seems insignificant, a PDU Set may consist of many IP packets and the aggregate impact can still be significant. If the PSSize for which gNB schedules is less than the actual PSSize, when the last packets arrive it may take gNB one or more slots to schedule them, therefore delaying the delivery of the PDU Set, which is detrimental to low-latency applications such as XR applications. The opposite can also happen, which wastes resources.

Therefore, we have the following observation:

**Observation 1: it is important to make the PSSize accurate for low-latency applications.**

There are efforts to correct the impact of NAT46/64 on the PSSize [11]. However, these efforts are not able to tackle other causes. Moreover, the list of causes in the introduction clause is not complete – even if we list all possible causes today, novel network protocols that change the PSSize are likely to be deployed in the future. We have the following observation:

**Observation 2: A generic solution for correcting the PSSize is preferred.**

The UPF is the gateway to the 5G core. A UPF may handle a very large amount of traffic. In fact, some network operators have very few UPFs. Therefore, any solution that requires UPF to take action, such as addition, subtraction and multiplication, is undesirable. We have the following observation:

**Observation 3: To reduce the UPF complexity, it is preferred not to require UPF to correct the PSSize.**

When we don’t know the causes, how do we correct the error? There is a similar problem in physical-layer communication, where the channel seen by a receiver is the result of reflection and refraction of many unknown objects in the radio propagation environment. The solution there is to measure the channel by the sender sending a pilot signal known to the receiver and the receiver comparing the pilot signal and the received signal. We borrow the measurement idea, and the counterpart of the ‘pilot signal’ is the indicated PSSize value in the RTP header extension and the counterpart of the ‘received signal’ is the observed PSSize. If a PDU Set is delivered successfully, the UE will observe the same PSSize (by summing the sizes of all PDUs of a PDU Set) as the gNB does.

Once the UE figures out the difference between the indicated PSSize and the observed PSSize, it can signal to the sender on how to pre-compensate for the PSSize.

**Proposal 1: UE computes the difference between the actual PSSize and the indicated PSSize, and signals the difference to the RTP sender for PSSize pre-compensation.**

Specifically, the UE calculates a correction ratio - the actual PSSize to the indicated PSSize ratio – and sends the correction ratio to the RTP sender. The RTP sender pre-compensates the PSSize by multiplying the PSSize and this correction ratio.

To validate the effectiveness of the proposed method, we tested the method on video sequences. The results below are for the Racehorse video sequence, and the video encoder is H.264 with a target NAL Unit size of 1400 bytes. The average video frame size is 22.377 kB. The cumulative distribution of the slice size (with each slice encapsulated into an IP packet) is shown in Figure 6.4.2-1. Note that there are a significant number of sizes uniformly distributed between 0 and around 1300 bytes.

NAT46 occurs in the network.

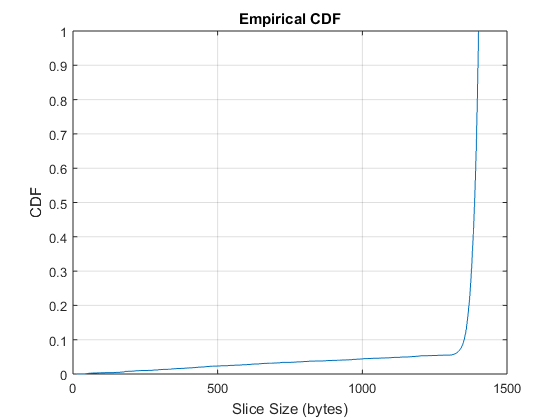


Figure 6.4.2-1: Slice size distribution

**Scenario 1 (MTU=576 bytes):**

The MTU size is set to 576 bytes (considered as a ‘safe’ MTU, because it is the IP packet size that all IPV4 nodes need to support [12]). This leads to fragmenting a packet size of 1400 bytes into three IP packets. As a result, the error in the PSSize comes from two sources: the presence of NAT46 and IP fragmentation. For the measurement-based correction method, the correction ratio is initialized to 1.

NOTE: The network configuration, e.g., NAT46 and MTU, typically changes much slower than the time scale of the feedback delay which is on the order of RTT. Thus, the feedback mechanism does not cause instability.

NOTE: As needed feedback - The UE can adapt the rate of feedback based on the observed error in the PSSize and a threshold on the tolerance of error.

The correction ratio (blue line) is shown in Figure 6.4.2-2.

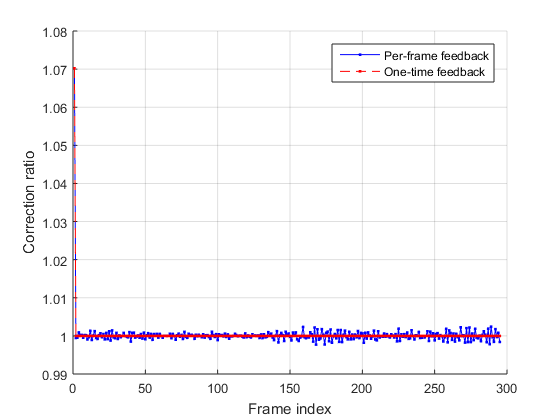


Figure 6.4.2-2: Correction for the per-frame feedback (blue line) and one-time feedback (red line).

The various PSSize’s are shown in Figure 6.4.2-3. The errors in the PSSize are shown in Figure 6.4.2-4. Without PSSize correction, the mean absolute error of the PSSize is 1616 bytes; with PSSize correction assuming NAT46 only, the mean absolute error is 1283 bytes; with the proposed measurement based PSSize correction, the mean absolute error is 27.5 bytes.

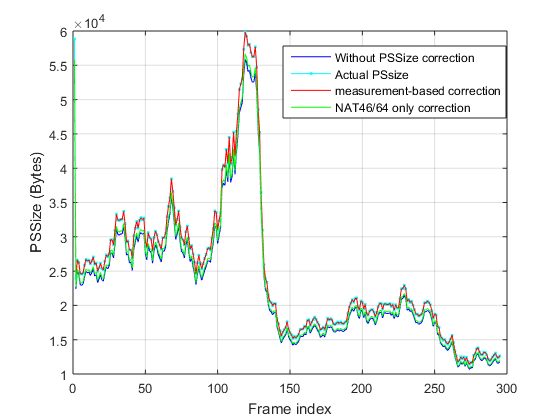


Figure 6.4.2-3: The actual PSSize (cyan line) vs the PSSize without PSSize correction (blue line), the PSSize with measurement-based correction (red) and the PSSize with the ‘NAT46/64 only correction’ (green).

If the UE sends the feedback once during the transmission of the video sequence, without further correction ratio received, the sender will use a correction ratio of 1 for future frames. The correction ratio is shown by the red curve in Figure 6.4.2-2. The mean absolute error is 23.9 bytes, which is lower than the error when per-frame (or per PDU Set) feedback is used. This is because the first frame is an I-frame with a large size and the estimated ratio based on it tends to be more accurate than the estimates obtained in other smaller sized frames which are P frames.

**Observation 4: For measurement-based PSSize correction, one-time feedback can be more accurate than more-frequent feedback.**

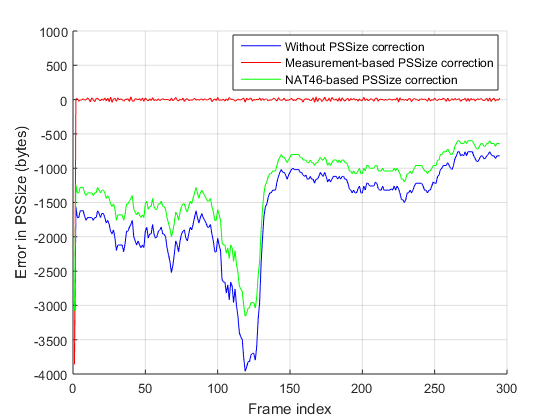


Figure 6.4.2-4: PSSize error for the cases of without correction (blue), with correction assuming NAT46 only (green), and with the proposed measurement-based correction with per-frame feedback(red) for Scenario 1. Scenario 2 (MTU=1300 bytes):

This is to show the effect of MTU size. The MTU size is set to 1300 bytes. A typical packet of 1400 bytes is fragmented into 2 packets. The errors in the PSSize are shown in Figure 6.4.2-5. The mean absolute error in the PSSize are 963.3 bytes, 630.0 bytes and 24.7 bytes for the case of no PSSize correction, NAT46 only correction and measurement-based correction, respectively.

With one-time feedback, the error is 20.7 bytes, again lower than the error of per-frame feedback 24.7 bytes.

We see that the respective errors decrease compared to smaller MTU size. However, the errors for case of no PSSize correction and the case of NAT46 only correction are still significant.

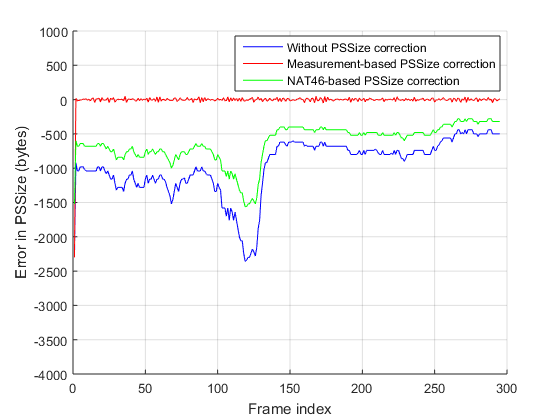


Figure 6.4.2-5: PSSize error for the cases of without correction (blue), with correction assuming NAT46 only (green), and with the proposed measurement-based correction with per-frame feedback (red) for Scenario 2.

From the results, we observe that:

**Observation 5: The proposed pre-compensation based PSSize correction method effectively reduces the PSSize error.**

For the ‘NAT46/64 only correction’ method to work, the sender needs to be aware of the presence of NAT46/64. The awareness may be obtained by feedback from the receive on the type of the IP address type seen by the receiver. The feedback may be one time during a session, and can be done by SDP signaling, Simple WebRTC Application Protocol (SWAP) in TS 26.113 [58], or RTCP.

TS 26.522 [2] provides guidelines for preventing IP fragmentation, either through path MTU discovery or by assuming a conservative MTU size at the sender in generating IP packets. Path MTU discovery needs support from the routers on the end-to-end path and incurs communication overhead, and a conservative MTU size may lead to unnecessarily small IP packet sizes which come with a higher packet header cost (i.e., the ratio of the size of packet headers to the size of the media). We consider the guidelines as a solution for IP fragmentation prevention.

We compare the three solutions. The first criteria is whether the solution is generic, i.e., whether the solution can tackle multiple and even unknown causes to the error in the PSSize. The criteria ‘need support from the network?’ means whether the network needs to be configured (e.g., configured to support path MTU discovery) to enable a solution.

Table 6.4.2-1: Pros and Cons of the three solutions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Solution | Generic | Accuracy | Need support from the network? | Communication overhead? | Need spec change? |
| NAT46/64 only correction | No | Low | No | Low | Yes |
| IP fragmentation prevention guidelines | No | Low | Yes if use path MTU discovery;  no if use conservative MTU size. | Moderate if use path MTU discovery;  none if use conservative MTU size. | No |
| Measurement-based correction | Yes | High | No | From low (one-time feedback) to high (per PDU Set feedback) | Yes |

## 6.5 Solution #5: Introduction of AL-FEC schemes defined in IETF

### 6.5.1 Key Issue mapping

This maps to Key Issue #3.

### 6.5.2 Description

IETF defined a few AL-FEC schemes including the codes, packet formatting and transmission methods, as detailed below. Some of them are Maximum Distance Separable (MDS) codes, meaning that they enable a receiver to recover the k source symbols from any set of k received encoded symbols.

- Non-MDS FEC schemes:

- FlexFEC: or Flexible Forward Error Correction, as defined in RFC 8627 [14]:

- FlexFEC relies on XOR operation in generating repair packets from source packets.

- FlexFEC currently is supported in the WebRTC implementation (RFC 8854 [20]).

- The encoding may be done in 1-dimensional or 2-dimensional fashion.

- A repair packet may protect a limited number of source packets.

- In the WebRTC implementation, the amount of redundancy depends on the packet loss rate, bitrate and RTT.

- The source packets have the same RTP packet format as regular packets without FEC, and the repair packets carry encoding information in the FEC Header (shown below) indicating which of the source packets are protected by this repair packet.

- Note that the FEC Header is part of the RTP payload and becomes invisible in the case of SRTP.

A list of text on a white background

Description automatically generated

Figure 6.5.2-1: RTP packet format for the repair packet for FlexFEC.

- ULPFEC: or Uneven Level Protection Forward Error Correction, as defined in RFC 5109 [13]:

- ULPFEC is similar to FlexFEC in the encoding operation but has the additional feature of providing multiple FEC levels for different parts of an application data unit.

- ULPFEC currently is supported in the WebRTC implementation.

- The source packet (called media packet in RFC 5109 [13]) follows the same RTP packet format without FEC, and the repair packet (called FEC packet in RFC 5109 [13]) follow the format shown below. Note that multiple FEC levels (protection levels) are supported.

- Again, the FEC Headers will be invisible in the case of SRTP.

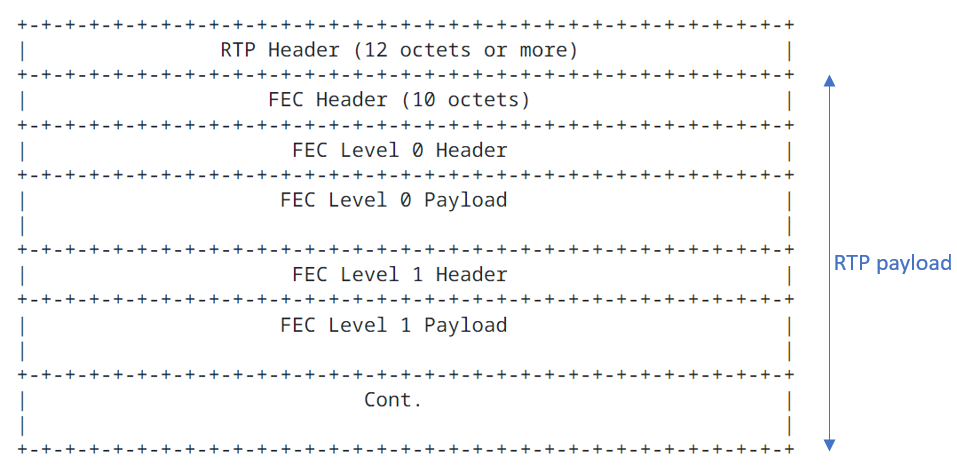


Figure 6.5.2-2: RTP packet format for ULPFEC

- MDS or near-MDS schemes:

- Reed-Solomon (RS) FEC: defined in RFC 5510 and RFC 6865 [16].

- RS FEC codes are MDS.

- They are commercially deployed in for example Meta Messenger.

- The source packet format and the repair packet format are shown in Figure 6.5.2-3.

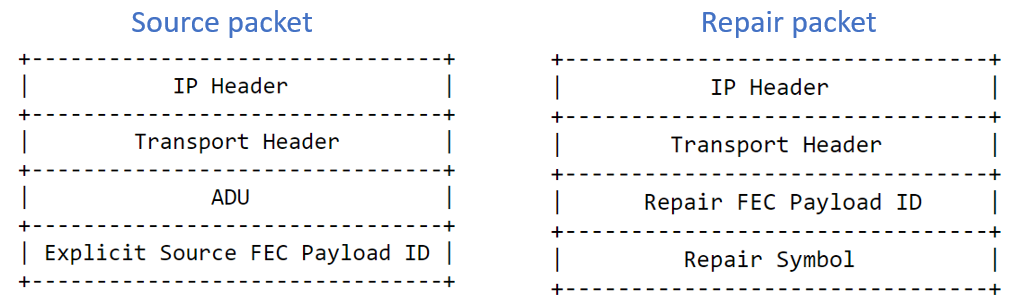


Figure 6.5.2-3: Format of the source packet and repair packet for RS FEC

- Raptor: defined in RFC 5053 [17].

- Raptor is a fountain code, i.e., as many encoding symbols as needed can be generated by the encoder on-the-fly from the source symbols of a source block of data. The decoder can recover the source block from any set of encoding symbols only slightly more in number than the number of source symbols.

- RaptorQ: defined in RFC 6330 [18].

- RaptorQ is a fountain code.

- RaptorQ codes provide superior flexibility, support for larger source block sizes, and better coding efficiency than Raptor codes. The RTP schemes for RaptorQ and Raptor are defined in RFC 6681 [15].

### 6.5.3 Categorization

There has been 3GPP study of AL-FEC (Reed-Solomon codes) for real-time communications, e.g., in 3GPP TR 26.922 [59].

Table 6.5.3-1 categorizes available standardized FEC schemes from IETF based on different criteria.

In addition, for RFC 6681 [15] and 6865 [16], and generally for the underlying FEC framework in RFC 6363 [19], the source data may be modified which may affect backwards compatibility of endpoints not supporting FEC and the application of encryption (i.e., if it happens before or after FEC).

For Raptor RaptorQ different schemes are defined in RFC 6681 [15].

- arbitrary sequence/arbitrary packet flow this needs additional information in the source packets

- single sequenced flow -> there is no change to the source packets

This is why in the fourth column both options yes and no are marked.

Performance is considered good if there is general repair capability for any loss without introducing too much latency. Performance is considered medium if there is general repair capability for any loss but introducing some latency and complexity. Performance is poor when reliability is still not guaranteed.

Table 6.5.3-1: Categorization of AL-FEC schemes for RTP in IETF

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name | RFC | Type | Format of source packets unchanged/ backward compatible | Resilience to Arbitrary packet loss | Flexible redundancy | Overhead (bytes) | Performance (repair capability) | MDS (incl. approximate MDS) |
| ULP FEC | 5109 | Parity/ XoR | Yes | No | Yes | High | Low | No |
| FlexFec | 8627 | Parity/ XoR | Yes | No | Yes | High | Low | No |
| Raptor/ RaptorQ | 6681 | Fountain/ LT | Yes/No | Yes | Yes | Medium | Good | Yes |
| Reed Solomon | 6865 | Polynomial | No | Yes | Limited | Medium | Good | Yes |

## 6.6 Solution #6: Time to next burst extension for the RTP HE for PDU Set marking

### 6.6.1 Key Issue mapping

This solution addresses the key issue #12.

### 6.6.2 Background

TS 26.522 [2] defines a data burst as 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.

TS 23.501 [3] enables an End of Data Burst (EoDB) indication to be added to the last PDU of each Data Burst in the GTP-U header to configure the UE power management schemes like Connected Mode Discontinuous Reception (CDRX). The procedure is as follows:

- PCF may provision the Protocol Description within the PCC rules based on the information provided by the AF and/or the local operator policies.

- SMF should request the UPF to detect the last PDU of the data burst and mark the EoDB in the GTP-U header of the last PDU in downlink, according to the PCC rule and/or the local operator policies.

- UPF identifies the last PDU of a data burst in the downlink traffic based on the End indication according to the Protocol Description and provides an EoDB indication to the RAN in the GTP-U header of the last PDU of a data burst.

If packets are transmitted in a bursty fashion, the idle time between two bursts is largely determined by the video frame inter-arrival time (e.g., ~33 ms for 30 fps). However, it may vary from burst to burst depending on the instantaneous variations in frame rate and when the NAL units comprising a PDU Set is made available by the encoder to the RTP packetizer, which may depend on the scene complexity. Furthermore, encoders that enable frame reordering may pass multiple frames to the RTP sender at once which are transmitted in a single burst that continues over multiple frame intervals.

Paced sending is a technique used in WebRTC to smooth the flow of packets sent to the network by spreading transmission across the frame interval. If paced sending is used, the sending time of the next packet or group of packets is determined by the pacer which may also consider factors such as frame rate and total bytes currently in the queue. In this case, since the transmission of the packets of a video frame is spread over the entire frame, the data burst concept may not apply depending on the amount of packets grouped together.

The EoDB indication informs the UE that there is an opportunity to sleep until the beginning of the next burst and enables the usage of CDRX mechanisms. However, the optimal power state of the UE depends on the time to next burst (TTNB) since the UE requires different transition times to switch to different power/sleep states, as described in TR 38.840 [21]. Therefore, EoDB by itself does not provide enough information for the UE or RAN to determine the appropriate sleep state for maximal power saving.

Table 6.6.2-1 shows the relative power consumption and total transition time (ramping down and up) of each sleep state for FR1 (frequency range 1, up to 7 GHz). Time interval for the sleep need to be larger than the total transition time entering and leaving a power state.

Table 6.6.2-1: Relative power and total transition time for the sleep states defined in TR 38.340 [21]

|  |  |  |
| --- | --- | --- |
| Sleep State | Relative Power | Total transition time |
| Deep sleep | 1 | 20 ms |
| Light sleep | 20 | 6 ms |
| Micro sleep | 45 | 0 ms\* |
| \* Immediate transition is assumed for power saving study purpose from or to a non-sleep state | | |

### 6.6.3 Description

In this solution, the RTP HE for PDU Set marking defined in TS 26.522 [2] clause 4.2 is extended to include a TTNB field. TTNB is the time interval between the transmission of the last packet in the current burst and first packet of the next burst, i.e., inter-burst time.

The TTNB field is 8 bits in length and is expressed in milliseconds. It can be an optional field added by the RTP sender only if the AS is able to obtain the TTNB information and it is subject to SDP negotiation between the sender and the receiver. TTNB is set to 0, if the current PDU is not the last PDU of the burst or if the sender cannot determine the TTNB.

Real-time congestion control algorithms like SCReAM [22] maintain an RTP queue at the sender side to temporarily store the RTP packets pending transmission. Thus, they can adapt the sending rate of packets depending on the congestion level and size of the frames produced by the video encoder. TTNB can be determined by RTP senders that implement such congestion control algorithms since they can choose when the next burst of packets will be sent.

An RTP sender determines the time to next burst based on its wall clock time. The timing is determined based on the sending time of the packets and not the time they are received by the RTP receiver. Optimization based on potential network jitter is not in the scope of this solution but can be implemented in the UPF or RAN.

TTNB can be used by the receiver UE to initiate the most adequate sleep state. For example, if TTNB is more than 20 ms, the receiver UE could achieve the optimal power saving by going to deep sleep.

An example implementation of the TTNB field is shown below for the one-byte version of the RTP HE for PDU Set marking.

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

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

| TTNB |

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

TTNB can be signaled as an optional extension attribute in the SDP signaling. An example usage of extmap attribute for such signaling is shown below.

a=extmap:7 urn:3gpp:pdu-set-marking:rel-18 short ttnb

NOTE: This solution requires coordination with SA2 and RAN2.

## 6.7 Solution #7: PDU Set Size information correction by indicating the remaining PDU Set Size in RTP header extension

### 6.7.1 Key Issue mapping

This maps to Key Issue #1.

### 6.7.2 Description

According to the current TS 26.522 [2], all PDUs of a PDU Set carry the same information in the PDU Set Size (PSSize) field in the RTP header extension for PDU Set marking. Repeating the same information in general is a waste of resource.

We propose to reuse the PSSize field, giving it a new interpretation or a new name, to indicate the remaining PDU Set Size (rPSSize), which includes the size of the PDU carrying the PSSize field, i.e., how many bytes the PDU Set has from this PDU to the last PDU of the PDU Set. For the first PDU of a PDU Set, the rPSSize and the PSSize fields indicate the same value and have the same meaning. As an example, if the PSSize is 4000 bytes consisting of 4 PDUs with 1000 bytes each. The rPSSize field of the first PDU in the PDU Set will indicate 4000 bytes, the same as the PSSize does. The rPSSize field of the second PDU in the PDU Set will indicate 3000 bytes instead of 4000 bytes that would be indicated by PSSize.

This proposal allows a router to compare the indicated size (i.e., the size from the perspective of the application sender) of a PDU (by taking the difference in the rPSSize between two adjacent PDUs) and the observed size of the PDU and derive the PSSize error due to network operations such as NAT46/64 that alter the PSSize. Using the same example, we assume that there is NAT46 in the network unknown to the packet source. The router can derive the indicated size of the first PDU (PDU Sequence Number or PSN=0) by taking the difference between the rPSsize carried in the 1st PDU (PSN=0) and the rPSsize carried in the 2nd PDU, as shown in Figure 6.7.2.-1. The difference will be 4000 – 3000 = 1000 bytes. On the other hand, the router observes that the 1st PDU has an actual size of 1020 bytes. Then, the router knows that the network has added 20 bytes to the packet, and it needs to add 20 bytes for each PDU (this means that the number of PDUs in the PDU Set (NPDS) needs to be signaled in the PDUs) in the PDU Set to get the actual PSSize.



Figure 6.7.2-1 Deriving the indicated size of the 1st PDU by taking the difference in the rPSSize between the 1st PDU and the 2nd PDU

One may argue that in the event of out-of-order delivery, with the current specification TS 26.522 [2], if every PDU carries the PSSize, the first received PDU (whose PSN may not be equal to 0) will provide the PSSize information needed by a router. This is not necessary. First, for low-latency applications, a reasonable design is not expected to lead to severe out-of-order delivery.

Second, if every PDU carries its respective rPSSize, the router can use the PSN filed in the RTP header extension together with the rPSSize to estimate the PSSize. As more packets arrive, the estimate will get more accurate.

Third, even if the packets arrive at a router out-of-order, it may not have a problem. To see this, consider two cases. Case (1) all packets arrive in an ideal burst (i.e., all PDUs arrive within a time slot or a transmission time interval (TTI) equal 1 ms): in this case, the router performs the PSSize correction described earlier, but in a very short duration of an ideal burst, offering even more lead time for scheduling. Case (2) all PDUs are evenly distributed in time until the next PDU Set: in this case, not being able to get the PSSize in the first time slot does not necessarily prevent scheduling the PDUs arriving in the first time slot. Under rare conditions (e.g., the first arrived PDU has the largest PSN among all PDUs of the PDU Set), the rPSSize obtained in the first time slot is less than the total size of the PDUs in the first time slot, the unscheduled PDUs can still be scheduled in the next time slot. What really matters is to timely deliver the PDU Set as a whole, an observation that motivated the notion of Nominal PDU Set Delay Budget (NPSDB) (see clause 6.20 of [6]). As long as the last few PDUs are scheduled on time, which is guaranteed, the timely delivery of the whole PDU set is not affected.

To implement this solution, we can replace the PSSize in the RTP header extension for PDU Set marking with rPSSize, updating the semantics of the field without changing the format.

Alternatively, without changing the name, the PSSize field can be re-interpreted as the remaining PSSize during session setup.

**Pros:** compared to other solutions, it doesn’t incur additional signaling in the user plane or the control plane beyond what is needed for supporting the PSSize in the current TS 26.522 [2].

**Cons:** it assumes that the intermediate routers (e.g., UPF, gNB) use the rPSsize value to correct the PSSize, but when a router serves a large number of traffic flows

- such computation may not be scalable,

- the router needs to maintain a state variable (to store the rPSSize in the most recently received PDU) in the memory,

- it needs the inclusion of the optional Number of PDUs in a PDU Set (NPDS) field in the RTP header extension for PDU Set marking,

- this method cannot correct the PSSize when the first PDU is received, and it needs to wait for at least another PDU before it can correct the PSSize

- this solution would not work with current Stage 2 work when the first PDU of the PDU set is not delivered first (i.e., in order) and requires stage 2 update to address this issue.

- it also requires changing the semantics of PSSize as defined in TS 26.522 [2],

## 6.8 Solution #8: Definition of the PDU Set for Application-Layer FEC

### 6.8.1 Key Issue mapping

This maps to Key Issue #4.

### 6.8.2 Description

In Rel-18, the PDU Set was defined without the consideration of AL-FEC. When AL-FEC is used, the RTP source typically generates both source packets and repair packets. A natural question is whether we need to conglomerate the source packets and the repair packets of an ADU into a single PDU Set or into two PDU Sets. To answer this question, we need to consider how the source packets and the repair packets are multiplexed because the multiplexing has an impact on the QoS provisioning.

Regarding the definition of the PDU Set in the case of AL-FEC, there are two options:

**- Option 1** (separate PDU Sets): A PDU Set includes only the source packets of an ADU and another PDU Set includes only the repair packets of the same ADU

**- Option 2** (the same PDU Set): A PDU Set includes both the source packets and the repair packets of an ADU

There are three ways to multiplex the source packets and the repair packets:

- **Scheme 1** (in a single RTP stream): The source packets and the repair packets of an ADU are sent in the same RTP stream, which is identified by an SSRC. This multiplexing scheme is used for ULPFEC in the WebRTC implementation [28].

- **Scheme 2** (in different RTP streams of an RTP Session): The source packets and the repair packets of an ADU are sent in two separate RTP streams of the same RTP session, and the streams are identified by two different SSRC’s. This multiplexing scheme is used for FlexFEC in the WebRTC implementation [28].

- **Scheme 3** (in different RTP sessions (IP 5-tuples)): The source packets and the repair packets of an ADU are sent in two separate RTP sessions, which are identified by two different IP 5-tuples. This is recommended in RFC 5109 [13], although we are not aware of any such commercial implementation.

In TS 23.501 [3], the QoS for PDU Sets is provisioned on a per QoS flow basis. A QoS flow is typically identified by an IP 5-tuple. The network identifies which IP 5-tuple a PDU Set is associated with and then provisions QoS. With schemes 1 and 2, the source packets and the repair packets of an ADU are still associated with the same IP 5-tuple, allowing for both options for the definition of the PDU Set. However, if option 1 is used, the network needs to correlate the two PDU Sets, and this incurs additional complexity. Therefore, option 2 is preferred.

In contrast, with scheme 3, the source packets and the repair packets of an ADU are associated with different IP 5-tuples. For option 1 of the PDU Set definition, i.e., the source packets and repair packets forming two PDU Sets, the network needs to correlate the two PDU Sets for QoS provisioning. For option 2 of the PDU Set definition, a PDU Set is split into two QoS flows, and it will be difficult for the network to provision QoS to the two QoS flows jointly to meet the QoS for a single PDU Set.

The complexity for PDU Set QoS provisioning is summarized in the table below:

Table 6.8.2-1: Complexity for PDU Set QoS Provisioning

|  |  |  |  |
| --- | --- | --- | --- |
|  | Complexity for PDU Set QoS provisioning | | |
| Multiplexing Scheme 1  (in a single RTP stream) | Multiplexing Scheme 2  (in two RTP streams of an RTP session) | Multiplexing Scheme 3  (in two RTP sessions or with two IP 5-tuples) |
| Option 1: separate PDU Sets | High | High | High |
| Option 2: the same PDU Set | Low | Low | High |

**NOTE:** Scheme 1 and Scheme 2 are deployed commercially, while commercial deployment of Scheme 3 has not been found.

Based on the summary, we see that option 2 of the PDU Set definition has the advantage of having lower complexity for PDU Set QoS provisioning. Therefore, we arrive at the following conclusion:

**Observation 1:** To minimize the complexity for PDU Set QoS provisioning in the case of AL-FEC, a PDU Set is defined to include both the source packets (PDUs) and the repair packets (PDUs) of an ADU.

## 6.9 Solution #9: RTP retransmission aware PDU Set handling

### 6.9.1 Key Issue mapping

This solution addresses the key issue #8.

### 6.9.2 Background

#### 6.9.2.1 RTP retransmission payload format defined in IETF

IETF defined an RTP retransmission payload format in RFC 4588 [31]. The payload format was designed for use with the extended RTP profile for RTCP-based feedback, the RTP/AVPF defined in RFC 4585 [33].

Retransmission packets carry copies of lost packets along with sequence numbers and timestamps to facilitate accurate reconstruction at the receiver. The timing and frequency of retransmission packets are controlled by the sender based on network conditions and feedback from the receiver. This allows a trade-off between reliability and delay; the endpoint may give up on retransmitting after a given buffering time.

RTP retransmission can be performed selectively, meaning that only some of the lost packets are retransmitted, rather than entire data blocks. This selective approach can minimize the overhead associated with retransmissions, as only the packets deemed necessary by the sender may be retransmitted. TS 26.114 [32] recommends senders to retransmit packets which they deem beneficial for timely recovery.

Senders are not required to retransmit and exact copy of the lost source packet. For example, they may retransmit the same encoded data at a lower rate to avoid overloading the network. However, senders must ensure that the receiver will still be able to decode the payload sent in the retransmission packet. Senders can determine the acceptable bit rate and packet rate according to the congestion control mechanism defined in the RTP/AVPF profile.

The format for retransmission packets is shown in Figure 6.9.2.1-1. The sequence number of the source RTP packet, i.e., the Original Sequence Number (OSN), is inserted into the first two octets of the RTP payload as the payload header. The remaining payload corresponds to the original RTP packet payload.

1 
2 
3 
01234567890123456789012345678901 
Header 
OSN 
Original 
RTP 
Packet Payload 

Figure 6.9.2.1-1: Retransmission packet format defined in RFC 4588

Upon detection of a lost packet, the receiver decides whether to request a retransmission or not. The decision may depend on e.g. the media type, tolerable application delay and network conditions. Receivers are expected to use the RTCP NACK feedback message format defined in the RTP/AVPF profile to send retransmission requests. Before sending another NACK to request a new retransmission of a packet, receivers are expected to detect that the previous retransmission failed based on an estimate of the round-trip time (RTT). NACKs can be sent in regular compound RTCP packets or early RTCP packets (as per RTP/AVPF). Format of the generic NACK message is shown in Figure 6.9.2.1-2.

1 
2 
3 
01234567890123456789012345678901 
PID 
Figure 4: 
BLP 
Syntax for the Generic NACK 
message 

Figure 6.9.2.1-2: Syntax for the Generic NACK message defined in RFC 4585

Semantics of the fields are:

- Packet ID (PID): RTP sequence number of the lost packet.

- Bitmask of following lost packets (BLP): Allows for reporting losses of any of the 16 RTP packets immediately following the RTP packet indicated by the PID.

When SDP is used to indicate the use of retransmission for an RTP stream, the mapping is done using the fmtp attribute as follows:

a=fmtp:<number> apt=<apt-value>;rtx-time=<rtx-time-val>

- <number>: payload type of the retransmission stream

- <apt-value>: payload type of the original/source stream

- <rtx-time-val>: time in milliseconds (measured from the time a packet was first sent) that a sender keeps an RTP packet in its buffers available for retransmission. If this parameter is absent, max retransmission time is undefined and but may be negotiated by other means.

TS 26.114 [32] recommends a minimum "rtx-time" value equal to the RTT and the maximum value equal to 400 ms.

RFC 4588 [31] requires the original and retransmission packets to be sent in two separate streams. Two options are given.

1) Session-multiplexing: The streams are multiplexed by sending them in two different sessions. In this case, the original and retransmission streams are sent to different network addresses or port numbers.

2) SSRC-multiplexing: The streams are sent in the same session using different SSRC values. This allows minimizing the port usage since the same port can be used for both streams.

According to TS 26.114 [32], MTSI senders and receivers shall support handling of RTP retransmission packets using SSRC-multiplexing.

WebRTC requires, in RFC 8834 [35], that the endpoints support handling of RTP retransmission packets using SSRC multiplexing and leaves the support of session-multiplexing optional.

#### 6.9.2.2 WebRTC usage

The libwebrtc implementation of WebRTC uses an adaptive NACK/FEC scheme where the configuration is adapted depending on the RTT. If the RTT is below a threshold, the NACK mode is used, i.e., lost packets are retransmitted, and FEC is not used. If the RTT is above another threshold, only FEC is used since the delay incurred by retransmissions may be prohibitive for the application. Another hybrid mode utilizes both FEC and retransmission in an adaptive manner when the RTT lies between the two thresholds.

When retransmissions are used, WebRTC applies a selective retransmission scheme. When a retransmission request (NACK) is received, the RTP sender ignores the request if the packet has been retransmitted in the last RTT msecs. Otherwise, it retransmits the packet if a copy of the packet is still found in its buffer. Retransmission rate is limited according to a bandwidth estimate to avoid sending too many retransmissions and aggravating the congestion.

More information on the adaptive NACK/FEC mechanism can be found in the paper: "Handling packet loss in WebRTC" [26].

#### 6.9.2.3 Feasibility of RTP retransmission in XR applications

RTP retransmission can be a feasible approach for XR media delivery, but its feasibility depends on several factors:

**Network Conditions:** XR applications often require low latency and high bandwidth. RTP retransmission can help recover lost packets, but it introduces additional latency. The network must support low-latency communication to ensure that retransmissions do not degrade the user experience.

XR applications may be highly sensitive to latency. Retransmissions can add delay, which might be acceptable for some types of XR content, but not for others (e.g., highly interactive XR experiences). To accommodate retransmissions, some buffering is necessary. However, excessive buffering can increase latency. Jitter buffers can help manage variability in packet arrival times, but they must be carefully tuned to balance latency and smooth playback.

RFC 4588 [31] clause A.3 provides an analysis of the delay incurred by RTP retransmission. Components of the overall delay (T) are the additional round-trip-time (RTT), packet loss detection time (T2), time to next RTCP report (T3) and processing time (T5), which comprises the feedback processing time at the sender and the queuing time of the retransmission packet:

T = RTT + T2 + T3 + T5

According to the analysis, T2 and T5 can be considered negligible relative to the RTT and T3. T3 is affected by the RTCP interval duration; the worst case would be that we assume that reporting has to wait a whole RTCP interval.

RTP retransmission is more effective in networks with low to moderate packet loss. High packet loss rates can overwhelm the network, leading to increased latency and reduced quality of experience.

For large-scale XR deployments, the scalability of RTP retransmission mechanisms must be considered. Since retransmissions can increase network load, the infrastructure must be capable of handling the additional traffic.

**Application Requirements:** Different XR applications have varying requirements for quality, latency, and reliability. RTP retransmission might be suitable for some applications but not for others.

The authors of [51] examined the network demands and protocols of cloud gaming services Google Stadia, GeForce Now and PS Now and discovered that Stadia has a retransmission stream whereas PS Now has a retransmission or FEC stream. Cloud gaming and XR applications are generally regarded as having similar latency requirements, although the exact thresholds can vary depending on the specific use case and application.

For XR applications, TR 26.928 [52] indicated a motion-to-photon delay upper bound of 20 ms as the requirement for perception of visual presence. However, the delay requirements may not be so stringent in other cases depending on the desired degree of visual presence and application scenario.

**Conclusion:** RTP retransmission can be a feasible solution for XR media delivery under the right conditions, particularly in networks with low latency and moderate packet loss. However, its suitability may need to be evaluated based on the specific requirements of the XR application and the network environment.

### 6.9.3 Description

RTP retransmission is negotiated and configured end-to-end between the sender and the receiver. However, currently, there is no mechanism to indicate to the 5G network whether RTP retransmission is performed and, if yes, how it is configured.

When PDU Set based handling is used, this may lead to suboptimal operation since the 5G network cannot configure the network operations like buffering, scheduling, packet discarding in a way that would benefit from awareness of RTP retransmission. For example, in case of momentary congestion, the RAN could have a higher preference for discarding packets from applications that use retransmission considering that the discarded packets will be retransmitted by the application hopefully within a non-congested period (and assuming that the packet is still found in the sender application buffer).

In this solution, the sender indicates to the network that it has successfully negotiated the use of RTP retransmission with the receiver, and thus retransmissions can take place during the session. The indication can be sent via control plane signaling e.g. in the Protocol Description signalled by the AF.

**Differentiated configuration of PDU Set QoS parameters**

According to RFC 4588 [31], original packets and retransmission packets are carried in different RTP streams, either in the same RTP session or in different RTP sessions. When PDU Set handling is used, PDU Set QoS parameters can be set for each RTP stream by the AF.

PDU Set QoS parameters applied to retransmission streams can benefit from differentiated configuration. For example, the retransmission stream may be assigned a shorter PDU Set Delay Budget (PSDB) so that it becomes more likely that the retransmitted packets reach the receiver before the playout deadline of the media unit they are associated to.

NOTE: It is assumed that the retransmitted PDUs and original PDUs are placed in PDU Sets mapped to different QoS flows such that different PDU Set QoS parameters can be applied.

**Modified PDU Set marking for retransmitted PDUs**

When PDU Set handling is used, RTP senders can insert the RTP HE for PDU Set marking (defined in TS 26.552 clause 4.2) to outgoing RTP packets in order to add the PDU Set Information. However, for retransmitted PDUs, some of the data fields present in the RTP HE for PDU Set marking may not be necessary.

The two optional fields in the RTP HE for PDU Set marking are the PDU Set Size (PSSize) and the Number of PDUs in the PDU Set (NPDS).

PSSize is intended to be used by the RAN for allocation of scheduling resources efficiently to PDU Sets. Until a retransmitted PDU is delivered, most or all of the other original PDUs in a PDU Set will have been transmitted. Therefore, a retransmitted PDU does not need to be marked with PSSize, since this information would no longer provide any benefit to the network.

NPDS is intended to be used by the UPF to correct the PSSize calculation, in case a NAT64/NAT46 conversion has occurred in the network path changing the IP header size and thus invalidated the PSSize calculated at the sender. Thus, NPDS is similarly not necessary once all or most of the original PDUs in a PDU Set have been transmitted.

Among the mandatory fields, the PDU Sequence Number within a PDU Set (PSN) is not necessary for a retransmitted PDU since this information would not provide a correct ordering information in case of a retransmitted PDU.

The End of Data Burst (D) field is maintained since a data burst may contain PDUs from both the original and the retransmission stream and the last PDU of a data burst may correspond to a retransmitted PDU.

The PDU Set Sequence Number (PSSN) is maintained since this field allows the network to identify to which PDU Set a retransmitted PDU belongs and thus allows the network to determine whether the entire PDU Set (including the retransmitted PDU) can be delivered on time. For example, the network may estimate the delivery time for the retransmitted PDU and determine whether the transmission time for the entire PDU Set is still within the PSDB. If not, there might be no point of delivering the retransmitted PDU since the playout deadline will likely be missed.

The PDU Set Importance (PSI) field is maintained since retransmitted PDUs may also be subjected to PSI-based packet discarding in case of congestion. However, for differentiated handling of retransmitted PDUs, it could be beneficial to also indicate in the RTP HE whether the marked PDU is a retransmitted PDU. For example, if the network receives two PDUs with the same PSI value and one of them is a retransmitted PDU, that one may be considered to have higher importance and be treated more favorably in terms of resource allocation and scheduling.

NOTE: Lone PDUs may also benefit from a more compact RTP HE for PDU Set marking since parts of the RTP HE for PDU Set marking defined in TS 26.522 [2] such as PSSN, PSN, PSSize and NPDS may not be useful for the network in case of lone PDUs.

An example implementation of the modified RTP HE for PDU Set marking for retransmitted PDUs is shown in Figure 6.9.3-1. The flag indicating that the marked PDU is a retransmitted one is denoted by "X".

A diagram of numbers and letters











Description automatically generated

Figure 6.9.3-1: Modified RTP HE for PDU Set marking for retransmitted PDUs.

NOTE: This solution requires coordination with SA2 and RAN2.

## 6.10 Solution #10: AL-FEC awareness at RAN while considering upstream and downstream packet losses

### 6.10.1 Key Issue mapping

This maps to Key Issue #4.

### 6.10.2 Description

#### 6.10.2.1 Motivation

As explained in clause 5.4.2.2, there is an upstream network segment before the RAN and there may be a downstream network segment after the RAN (e.g., a tethered link). For DL data originated from the AS, when the RAN drops extra PDUs of a PDU Set encoded with AL-FEC, it needs to consider how many of the PDUs have already been lost in the upstream network segment. The RAN also needs to consider how many packets may get lost in the downstream network segment. Otherwise, the application receiver may not receive enough PDUs for reconstructing all the source PDUs.

NOTE 1: Although the solution below is on how the RAN drops extra PDUs, it applies to a generic network device if the network device receives the downstream packet loss probability.

#### 6.10.2.2 Solution



Figure 6.10.2.2-1: The RAN drops extra PDUs by considering packet losses in the upstream network segment and downstream network segment.

For DL data, the upstream is from the AS to the RAN, and the downstream is from the UE to the XR tethered endpoint.

The RAN derives the PDUs that have been lost in the upstream network segment, e.g., by taking the difference between the number of PDUs in the PDU set (NPDS) in the RTP header extension for PDU Set marking for a PDU Set and the number of PDUs of the PDU Set that it has received.

The downlink network segment informs the RAN of the packet loss rate over the tethered link. This is shown in Figure 6.10.2.2-1.

NOTE 2: Although in Figure 6.10.2.2-1 it is shown that the UE provides the packet loss feedback, the XR Tethered Endpoint can also provide the packet loss feedback.

The RAN then decides on how many PDUs are extra PDUs given (1) the upstream packet losses, (2) the packet losses at the RAN (not dropped as extra PDUs), and (3) the downstream packet loss rate.

The following example shows how the RAN decides on the number of extra PDUs that it can drop while still ensuring that the receiver can reconstruct all the source packets with a high probability. Denote the following:

(1) the upstream packet losses: *N1*

(2) the packet losses for the network segment from the ingress of RAN to the ingress of the UE (not dropped as extra PDUs): *N2*

(3) the downstream packet loss rate: *p*

It is assumed that the packet losses in the downstream are independent. Consider a PDU Set with be *K*=30 source PDUs, and 15 repair PDUs, hence a total of *N*=45 PDUs. Let the packet losses for the network segment from the ingress of RAN to the ingress of the UE (not dropped as extra PDUs) be *N2*. Let the AL-FEC code be a MDS code. For simplicity each PDU is assumed to be sent in a separate transport block over the RAN air interface. Let *N1*=2, *p*=2%. Let the minimum reconstruction success probability be 0.995. Then the allowed values for the number of extra PDUs *N3* that the RAN can drop must satisfy the following inequality:

0.995, where

In the above, the summand gives the probability that packets out of packets are a success and the remaining packets are a failure. The probabilities are summed across all values of which allow the application receiver to reconstruct all the source packets.

A graph with a blue line

Description automatically generated

Figure 6.10.2.2-2: The RAN can drop up to N3=6 extra PDUs while still meeting the reconstruction probability threshold 0.995 with the packet losses in the upstream, at the RAN (due to wireless link error, not dropped as extra PDUs), and in the downstream.

NOTE 3: Although the solution presented above is for DL traffic, a similar solution can be devised for UL traffic, in which case the tethered link becomes upstream and the gNB to the AS segment (5G core network and N6 segment) becomes downstream.

NOTE 4: For UL data, the downstream is from the AS to the RAN, and the upstream is from the XR tethered endpoint to the UE.

For the uplink data traffic, the UE needs to determine the number of PDUs that it needs to successfully deliver over the air between the UE and the gNB. The number depends on the packet loss rate for the gNB to the AS segment. It is difficult to directly measure the packet loss rate. One possible method is to send probing packets (e.g., ICMP ping messages), but the AS may not recognize the IP address of the gNB and may reject the probing packet for security reasons. Even if transmitting probing packets is enabled, the probing packets may be treated differently in the network from the media packets, resulting inaccurate estimation of the packet loss rate. As an alternative, the packet loss rate may be estimated indirectly: the gNB provides feedback on the uplink packet reception, the AS provides feedback on the end-to-end packet reception, and the UE estimates the packet loss rate for the gNB to the AS segment, denoted as , which is the ratio of the number of packets lost as observed by the AS but not observed as lost by the gNB to the number of packets successfully received by the gNB. This is illustrated with an example in Figure 6.10.2.2-3.

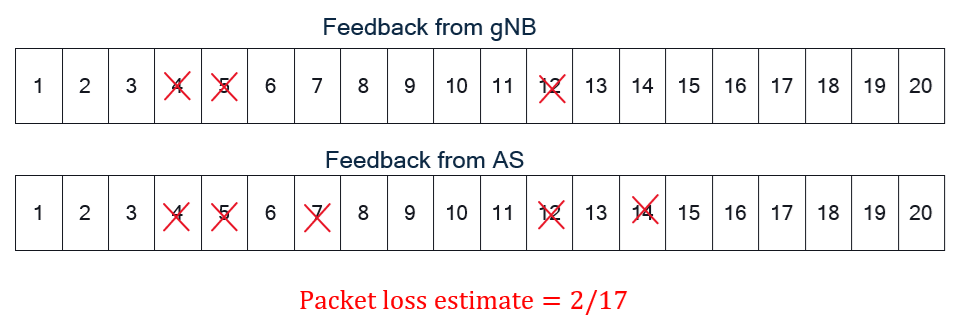


Figure 6.10.2.2-3: Indirect method for calculating the packet loss rate for the gNB to AS segment, where an X sign means the corresponding packet is lost

NOTE 5: The feedback that the gNB provides on the uplink packet reception may be in the form of HARQ feedback (using New Data Indicator (NDI)). RLC Status PDU (in the case of RLC AM), or PDCP Control PDU.

Once the packet loss rate for the gNB to the AS segment is estimated, a similar formulation can be used to derive the number of PDUs that the UE can drop to meet a minimum reconstruction success probability at the AS.

To count the RTP packet losses from the gNB feedback, the UE needs to keep a mapping between the RTP packets and the lower-layer packets whose status is conveyed in the gNB feedback.

The future packet loss rate for the network segment from the gNB to the AS may be greater than the estimate. To minimizes the probability of failing to reconstruct the source PDUs, the UE may add a margin in the estimate, i.e., using a larger packet loss rate, in the same approach taken by a traffic source that sets the redundancy for AL-FEC based on end-to-end packet loss rate estimation.

The above assumes that the extra PDUs dropped at the RAN are not taken as signals of network congestion. Network congestion can occur in the upstream network segment, at the RAN or in the downstream network segment.

NOTE 6: It is FFS under what conditions the RAN can drop extra PDUs considering network congestion in the upstream network segment, at the RAN or in the downstream network segment.

NOTE 7: For RLC UM, the NDI design may not that accurate since the gNB may decide to give up the TB transmission due to retransmission time limits, which means the calculation is still not accurate. This may need cross-layer support, e.g. mapping between TB and the PDCP PDU may be needed in order to derive the lost application layer packets. The probability of such events is FFS.

NOTE 8: There is no clear relationship between the packet loss rate in Uu interface between UE and RAN (i.e. ) and the E2E packet loss rate, but is expected to be larger than . Additional details of this relationship are FFS.

**Cons:** One drawback of the solution is that the underlying system may take some decisions that are detrimental to the application. If the RTP sender estimates the E2E loss rate wrongly, then that’s the fault of the application layer itself. Otherwise, if the 5G system will be responsible. Therefore, when implemented in 5GS the estimates need to be accurate.

## 6.11 Solution #11: PSI indication to optimize RTP retransmission

### 6.11.1 Key Issue mapping

This solution addresses the key issue #8.

### 6.11.2 Description

Senders have the best view on which packets are sufficiently important to be retransmitted. Selective RTP retransmission prioritizes the packets that contain crucial data, such as keyframes in video streaming or important audio segments in voice calls.

In this solution, the sender signals a threshold/upper bound to the receiver indicating the range of PSI values assigned to the PDU Sets that are deemed critical for the application. The PSI threshold can be sent by the sender in an SDP negotiation, e.g., by means of a new SDP attribute "psi-thr" that can be set a value between 0 and 15 (inclusive).

By this signaling, the sender indicates that such PDU Sets are critical for the session and need to be retransmitted in case they are lost. For example, a PSI threshold of 8 means that the sender asks the receiver to consider requests retransmissions for the PDU Sets with PSI between 0 and 8 (inclusive).

The receiver uses the indicated PSI threshold to determine for which lost packets it will send retransmission requests (NACKs). Upon detection of a lost packet, the receiver inspects the PSI values of the received packets within the same PDU Set (i.e., have the same PSSN) and have RTP sequence numbers (SN) adjacent to the lost packet. From these PSI values, it can derive the PSI value of the lost packet. The receiver then sends a NACK, if the PSI value of the lost packet is lower than or equal to the PSI threshold.

NOTE 1: While making retransmission requests, receivers may also consider other factors such as the possibility of timely arrival of requested packets, as described in TS 26.114 [32], clause 9.3.2.

NOTE 2: In case a PDU Set consists of a single PDU Set, the solution does not apply, since PSI cannot be inferred from the adjacent PDUs in the same PDU Set. However, for typical video applications and bitrates, it is expected that a PDU Set comprises multiple PDUs.

The sender may also consider other factors (e.g. network conditions, number of lost packets) to decide whether to retransmit packets falling into the indicated PSI range during the session.

NOTE 3: This solution assumes that the PDU Set integrated handling is not used, i.e., the network does not discard the whole PDU Set when one PDU of a PDU Set is lost.

Figure 6.11.2-1 illustrates the solution with two example cases. The labels in the boxes show the PSN values of the PDUs with subscripts showing the PSI values for the respective PDU Sets.

In example 1, the PDU with RTP SN=14 (PSN=2, PSSN=3) is lost. When the receiver detects that loss, it can look at either the previous or the next PDU to infer the PSI since they are both in the PDU Set 3.

In example 2, the PDU with the RTP SN=15 (PSN=3, PSSN=3) is lost. In this case, the next PDU would not provide the correct PSI since it belongs to PDU Set 4, which has a different PSI value 9. The receiver can first inspect the PSSN value to check whether the next PDU is in a different PDU Set. If that is the case, it can instead use the value from the previous PDU with RTP SN=14 to obtain the correct PSI value 7 for the lost PDU.

A group of squares with numbers

Description automatically generated

Figure 6.11.2-1: Example cases illustrating the solution. Subscripts denote the PSI values.

**Benefit of the solution:** With guidance from the sender on the PSI range that is assigned to critical PDU Sets, the receiver can make more informed retransmission requests and improve bandwidth usage.

## 6.12 Solution #12: MID packet filtering

### 6.12.1 Key Issue mapping

This maps to Key Issue #9 and Key Issue #14.

### 6.12.2 Description

When multiple RTP streams are associated and multiplexed into a single RTP session, they can be grouped together called BUNDLE. IETF RFC 9143 [37] provides SDP BUNDLE framework using SDP offer/answer mechanism to negotiate which "m=" sections will become part of a BUNDLE group. In this framework, each "m=" section is associated with its identification-tag (the values of "mid" attribute) and a BUNDLE group is defined as the SDP 'group:BUNDLE' attribute having identification-tag list. All RTP-based media within a single BUNDLE group belong to a single RTP session. It means that all "m=" sections representing RTP-based media within a BUNDLE group share a single synchronization source (SSRC) numbering space. Additional rules and restrictions to be applied on that single RTP session are given in the section 9.1 of RFC 9143 [37].

IETF RFC 7941 [38] provides the RTP header extension method for the RTCP Source Description (SDES) items. The Figures 6.12.2-1 and 6.2.12-2 show the format of 1-byte and 2-byte extended headers, respectively. IETF RFC 9143 [37] defines the MID RTP HE registered by IANA in the "RTP SDES Compact Header Extensions" subregistry and carrying the identification-tag of the associated "m=" section. The MID RTP HE enables a receiver to associate each RTP stream with a specific "m=" section.

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

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

| ID | len | SDES item text value ... |

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

Figure 6.12.2-1: One-byte Header extension format for SDES items

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

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

| ID | len | SDES item text value ... |

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

Figure 6.12.2-2: Two-byte Header extension format for SDES items

Figure 6.12.2-3 shows an example of SDP answer when the answerer accepts the creation of the bundle group. In this example, three RTP media streams are bundled together by announcing from "a=group:BUNDLE" and each of stream is identified by using "a=mid:XXX".



Figure 6.12.2-3: Example of SDP for bundled media streams

As addressed above, MID value which was exchanged in SDP negotiation can be inserted into the extended header of each RTP packet, it can be used to identify how each of RTP packets are associated with the RTP media stream. Figure 6.12.2-4 illustrates the QoS mapping structure using MID packet filter which maps and video streams (RTP stream 1 & 2) to QoS Flow 1 and an audio stream (RTP stream 3) to QoS Flow 2, respectively.



Figure 6.12.2-4: QoS flow mapping by MID packet filtering

NOTE: TR 23.700-70 [6] has conclusions on the additional packet filter to identify each media flow. Therefore, it is desired to have a coordination with SA2 for normative work.

## 6.13 Solution #13: Dynamic traffic characteristics of RTP sender implementations

### 6.13.1 Key Issue mapping

This is a solution to KI #12 Enhancements of Data Burst Marking.

### 6.13.2 Description

#### 6.13.2.1 General

A data burst indicates a set of multiple PDUs generated and sent in a short period of time as defined in clause 3.1 of TS 23.501 [3]. Data burst is a common dynamic traffic characteristic in communication networks.

The source of traffic in this study is the Real Time Transport Protocol (RTP) as used for conversational or Extended Reality applications and conversational applications. Most experiments use traffic originating from a camera (conversational application), but also some synthetic sources are used that may be more suitable for the case of extended reality traffic.

The data burst identification could be beneficial for the 5GS network, e.g., power saving and efficient radio resource management. In Release 18, the End of Data Burst indication has been introduced to enable the UE power saving in the 5GS, i.e., the NG-RAN node can configure to move a UE into CDRX for power saving after transmitting the end PDU of the data burst. In Release 19, the data burst size has been concluded to enable the RAN radio resource management as described in clause 8.5 of TR 23.700-70 [6].

This solution aims to observe dynamic traffic characteristics from popular RTP senders and possible implications for low latency and real-time communication.

In this solution we explore dynamic traffic characteristics of the following implementations:

- The open source WebRTC implementation of an RTP Sender in a browser implementation.

- The cross platform GStreamer [39] implementation of RTP senders for peer-to-peer (P2P) transmission.

- The sending and receiving measured of two popular real-time streaming servers.

#### 6.13.2.2 WebRTC and paced sender implementation

##### 6.13.2.2.1 WebRTC and paced sender implementation

WebRTC (Web Real-Time Communication) [28] is an open-source project providing web browsers and mobile applications with real-time communication (RTC) via application programming interfaces (APIs). It allows audio and video communication and streaming in web pages by direct peer-to-peer communication, eliminating the need to install plugins.

It can be implemented in web browsers and is a likely source of real-time conversational RTP traffic in practice.

Based on the documentation available [28] from November 2021 in the WebRTC source code this clause describes the way packets may be generated and transmitted in a WebRTC environment.

WebRTC does not send out data packets generated by an encoder directly, instead a pacing module is used that limits and smoothens the packet train that is transmitted out to the network.

The idea is that, say for example, a 30 fps stream of 6 MBps is sent to the network, in the ideal case this would result in equally sized frames of around 25 kiloBytes in around 21 equally sized packets. While in practice, over a 1 second sliding window the average bit-rate of 6 Mbps might be measured, on a shorter timescale there may be periodic bursts that overshoot this average bit-rate with a much larger instantaneous bit-rate. One of the causes could be a sudden movement in a frame.

To avoid potential impact of these cases on the network and application, the WebRTC solution introduces the paced sender. The paced sender introduces a buffer in which the media packets are queued before being sent out. This avoids, for example, that large video frames are split into a large burst of packets that will disrupt other streams such as audio streams that are critical for intelligibility. WebRTC uses multiplexed RTP transport with different streams sent on the same port and IP tuple, thus potentially such an overshoot in video packets could stop other packets from being sent out in time as the socket may be blocked when transmitting the video packets.

NOTE: The standard approach in RTP is to use different ports for different media streams, but in practice this could have some drawbacks as this increases resource usage and may complicate NAT traversal. In this case, the prioritization would depend on the operating system implementation. WebRTC allows multiplexing audio and video on the same port by establishing a relationship between the media lines included in the SDP using the attribute a=group:BUNDLE.

For sending out the media, a leaky bucket algorithm is used for pacing them on to the network. In this implementation packets are inserted into a buffer first before being transmitted. The buffer contains separate first in first out (FIFO) queues for each media type/track, such that audio can for example be prioritized over video. Equal priority streams can be sent in round-robin fashion, avoiding these streams from blocking each another.

In a leaky bucket, the rate of packets that is sent out is limited, potentially leading to a buffering of the input in case the input is larger than the output rate. A leaky bucket is one way to limit the burst sizes in traffic sent to the network.

In WebRTC this output rate is called the *pacing\_rate.*

The typical way a packet is generated and paced on to the network in WebRTC is as follows:

1. Media frame is packetized into RTP packets

2. The packets are sent for scheduling (transmission)

3. The pacer is called to enqueue the packet

4. The packets stay in the queue until the pacer finds a scheduled moment to transmit them out based on the leaky bucket algorithm

5. At the time that the packet is scheduled, the packet is forwarded to the RTP module for final timestamping

6. The packet is sent on the low level interface (i.e. the UDP socket), and it is now out of scope of the WebRTC implementation

Asynchronous to this, the WebRTC implementation tries to estimate the sender bandwidth in order to set the *pacing\_rate,* i.e. the rate at which packets are sent to the network and secondly the *padding\_rate*, the bit rate (if any) used to send padding in case no packets are available to be sent out*.* Figure 6.13.2.2.1-1 illustrates the WebRTC transmission flow.

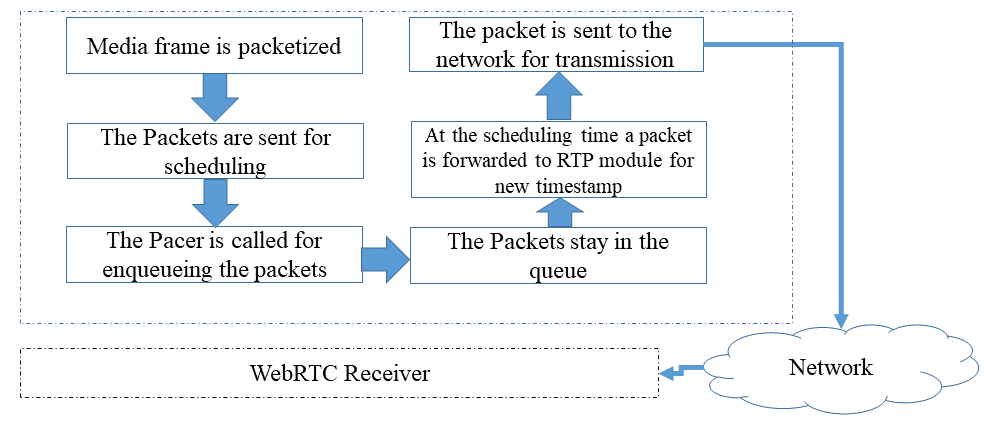


Figure 6.13.2.2.1-1: Packet processing in WebRTC paced sender of an RTP packet

The pacer of WebRTC prioritizes based on different criteria:

a) Packet Type, with most to least prioritized:

1. Audio

2. Retransmission

3. Video and FEC

4. Padding

b) Based on the enqueuing order.

When the queues are empty the implementation will aim to generate padding frames and send them out at the *padding\_rate*. In some cases the *pacer\_rate* may be overridden and ignored, when there is a significant encoding overshoot for example. In addition WebRTC implementation enables setting the maximum time a packet can stay in a queue using a function (*setQueueTimeLimit(TimeDelta limit))*.

Bandwidth estimation in the WebRTC implementation is done by bandwidth probing, i.e. a cluster of packets is requested to be transmitted over the network to gauge if this will lead to increased delay and or loss. The implementation provides a function to do this measurement, enabling one to use this information to update the *pacing\_rate*.

The WebRTC implementation has defined additional API functions to monitor the states and statistics of the pacer.

##### 6.13.2.2.2 WebRTC and paced sender evaluation

To collect traffic characteristics, the following setup is deployed:

- In this case a call is started on a Microsoft edge browser and Wireshark is used to collect the traffic trace. The machine initiating the call is a Lenovo laptop with 16 GB of RAM running windows 11 operating system (referred to as machine A). The network is a Wifi connection on 2.4 GHz band setup by a Samsung Galaxy A54 5G connected to a 5G network using Wifi hotspot feature.

Figure 6.13.2.2.2-1 illustrates the traffic pattern measured from a WebRTC sender that joined a google hangout call on Machine A. This stream has a lower bit-rate as compared to the next experiment. The traffic is clearly separated in burst that corresponds to the frame transmission.

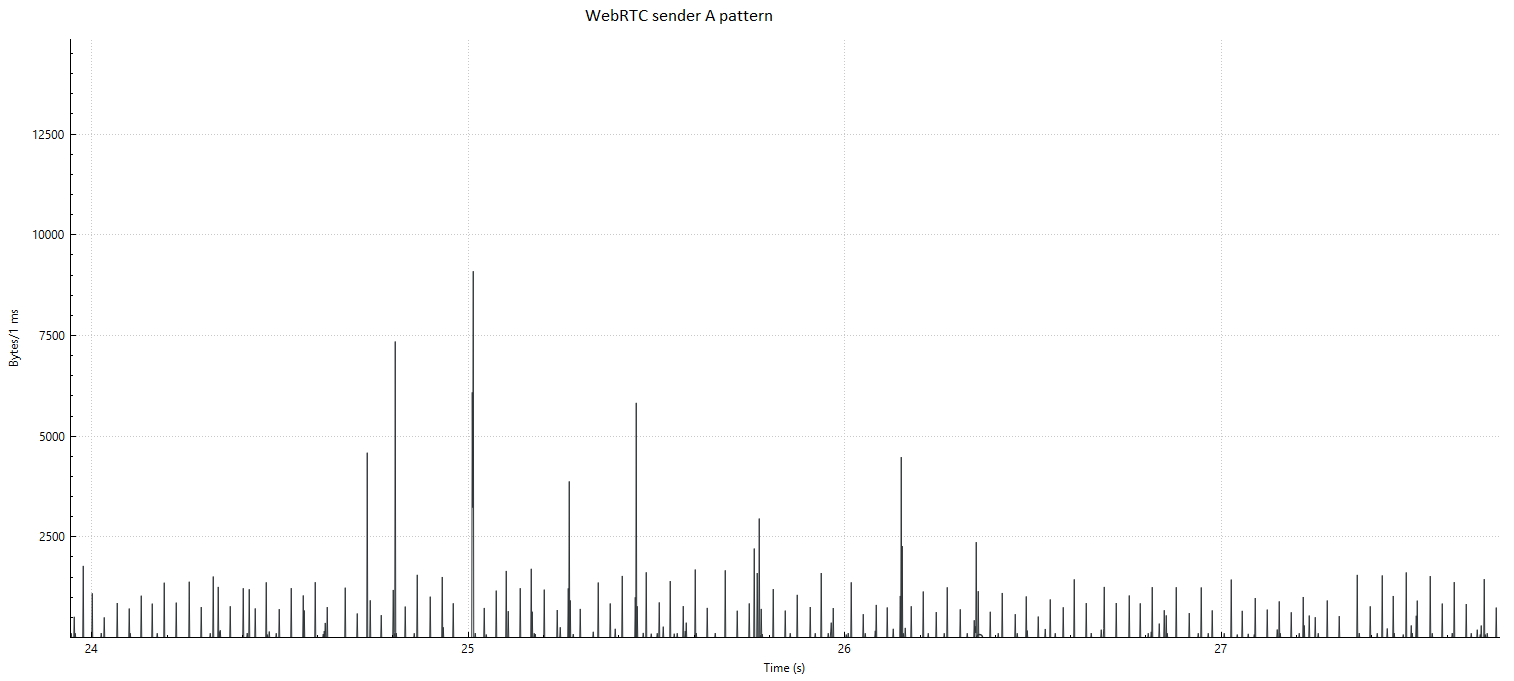


Figure 6.13.2.2.2-1: Sending pattern from WebRTC in a Google hangout call with video only

In a second experiment, we use a machine running Ubuntu 24.4 with 8 GB or RAM and Intel core i5 processor. The Ubuntu machine/Linux kernel enables throttling of the bandwidth and latency. The communication is via a Wifi and a 5G hotspot.

Linux traffic control was used using a token bucket filter (tbf) to slow down the data rate to 200 kbit and only enable 16 kbit bursts. The results are shown in Figure 6.13.2.2.2-2. We see the magnitude of the peaks is reduced as the hangout video quality drops after the traffic control filter is enabled. However the general traffic characteristics appear similar, there is no smoothening effect observed that may result from the paced sender. So, it seems in this case the general rate control of hangout for the video encoding reacted quick enough to avoid critical network operation.

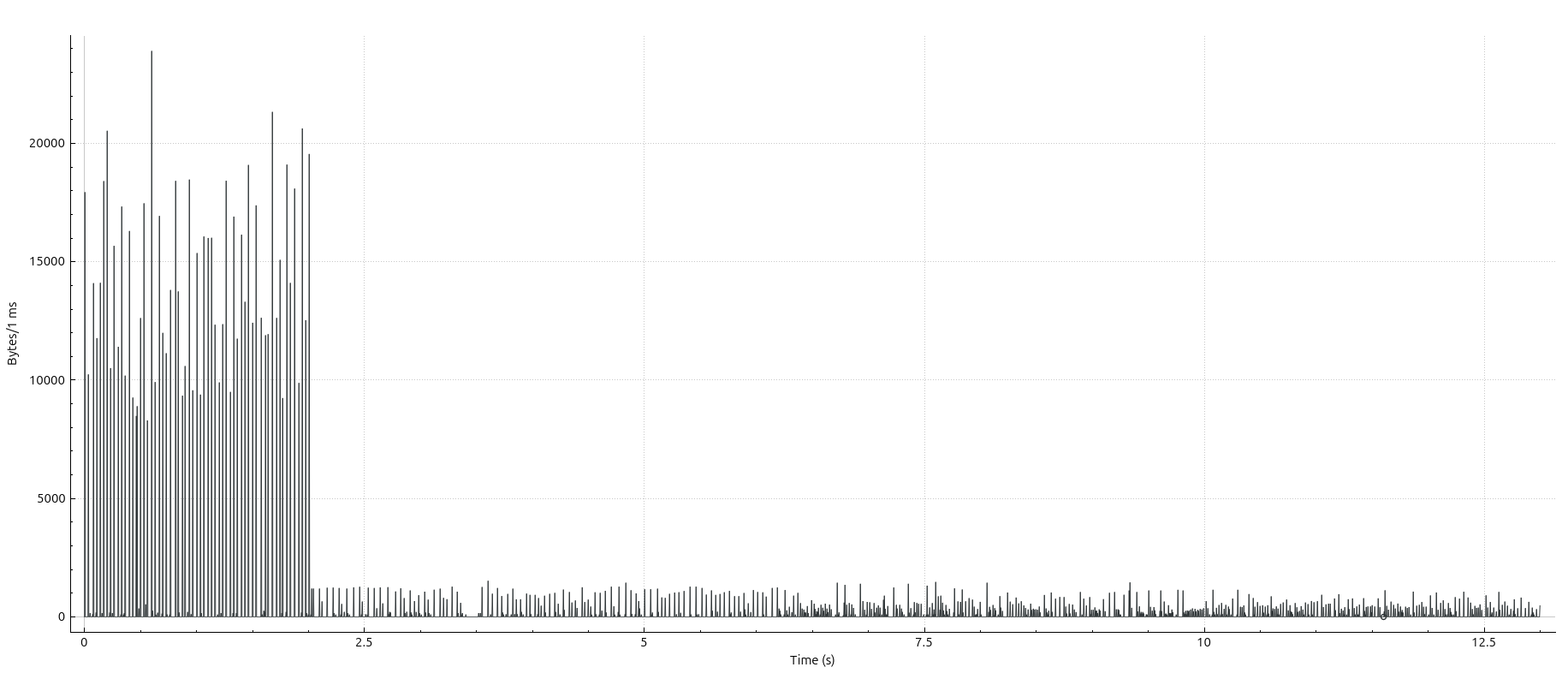


Figure 6.13.2.2.2-2: Resulting traffic characteristics at sender when the network bandwidth is throttled to 200 kbit audio + video

#### 6.13.2.3 GStreamer multimedia framework RTP implementation for P2P

##### 6.13.2.3.1 GStreamer multimedia framework RTP implementation details

GStreamer [39] is a library for constructing graphs of media-handling components. It supports a range of applications from simple playback to streaming up to complex mixing and editing workflows.

GStreamer works on all major operating systems such as Solaris, Unix etc.

It enables graph-based pipeline construction and has a broad coverage of different multimedia technologies for codecs, protocols, file format encapsulations etc.

The GStreamer 1.x API has been stable since 2012 and the library is quite likely to be found as a sub-component in many practical media systems implementations, especially open-source implementations.

In addition, the design of light weight data passing implies high performance and low latency, very applicable to real-time communications.

Due to the stability and long track record and wide deployment of this library, some sample pipelines utilizing the RTP features are developed in this study for collecting details on the traffic characteristics of this framework in simple RTP workflows.

The setup uses a camera followed by an H.264 encoder tuned for ultra-low latency and optimal speed settings, and the output is fed to an RTP pay-loader linked to a network sink. The receiver receives the RTP packets, decodes the streams and renders them to the screen enabling a 1-way conversational application.

##### 6.13.2.3.2 GStreamer RTP sender evaluation video

In this section the RTP video sender and receiver are evaluated.

To collect traffic characteristics, the following setup is used:

- Two Machines A and B (Lenovo and HP, both AMD Ryzen 5 processor, 16 GB RAM are connected over Wifi network (2.4 GHz) setup by a Galaxy A65 5G phone using hotspot function. Machine A runs the real time-frame capture, encoding, packetization and UDP transmission. Machine B receives the packets and depacketizes, decodes and renders the frame (receiver pipeline from the previous clause). Wireshark is used to collect the traffic data. In this setup a relatively high bit-rate of 16 MBps is used. GStreamer 1.24.5 is used for the experiment.

Figure 6.13.2.3.2-1 outlines the experimental setup.

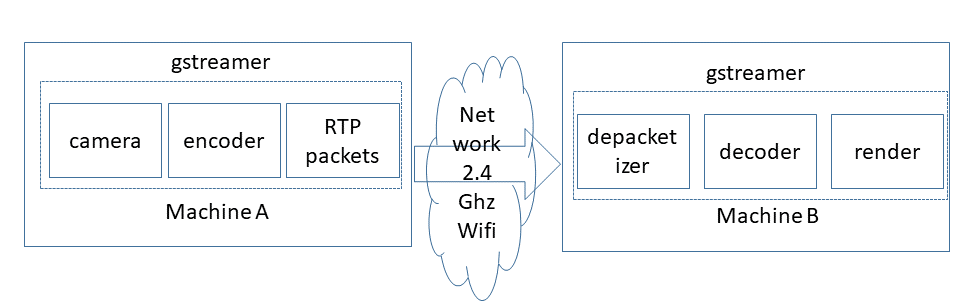


Figure 6.13.2.3.2-1: Experimental setup flow

Figure 6.13.2.3.2-2 illustrates the sending data traffic pattern in bytes per millisecond on a logarithmic scale. It is clear that the traffic is bursty and that the bursts occur on the timespan of 1-3 milliseconds.

Figure 6.13.2.3.2-3 illustrates the receiving data traffic pattern in bytes per millisecond on a logarithmic scale. The traffic is bursty and the burst occur in the timespan of 1-3 milliseconds, however already a little bit of dispersion can be seen due to transmission and delays experienced in the transmission.

Table 6.13.2.3.2-4 demonstrates a snapshot from the data collected from the sender A, one interesting observation is that the inter burst time is not constant. Between some burst there is approximately 20 milliseconds of idle period, while for other bursts the inter burst time is about 40-45 milliseconds.

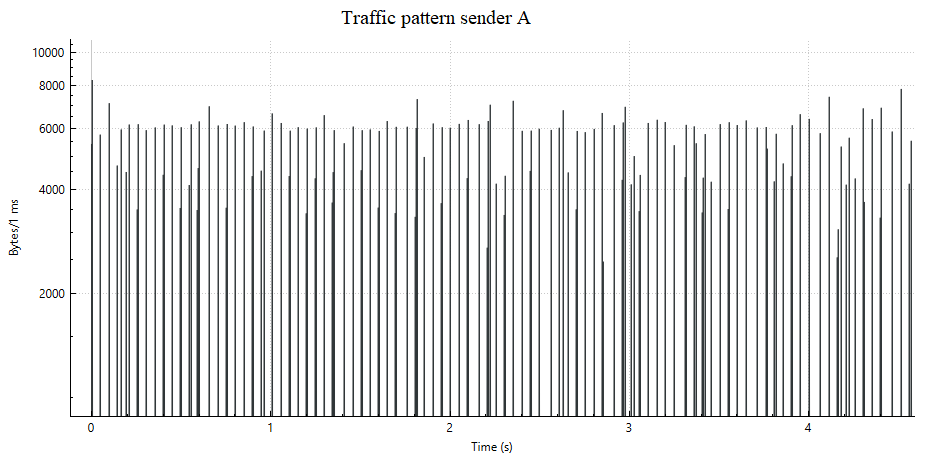


Figure 6.13.2.3.2-2: Traffic pattern in GStreamer RTP sender on machine A

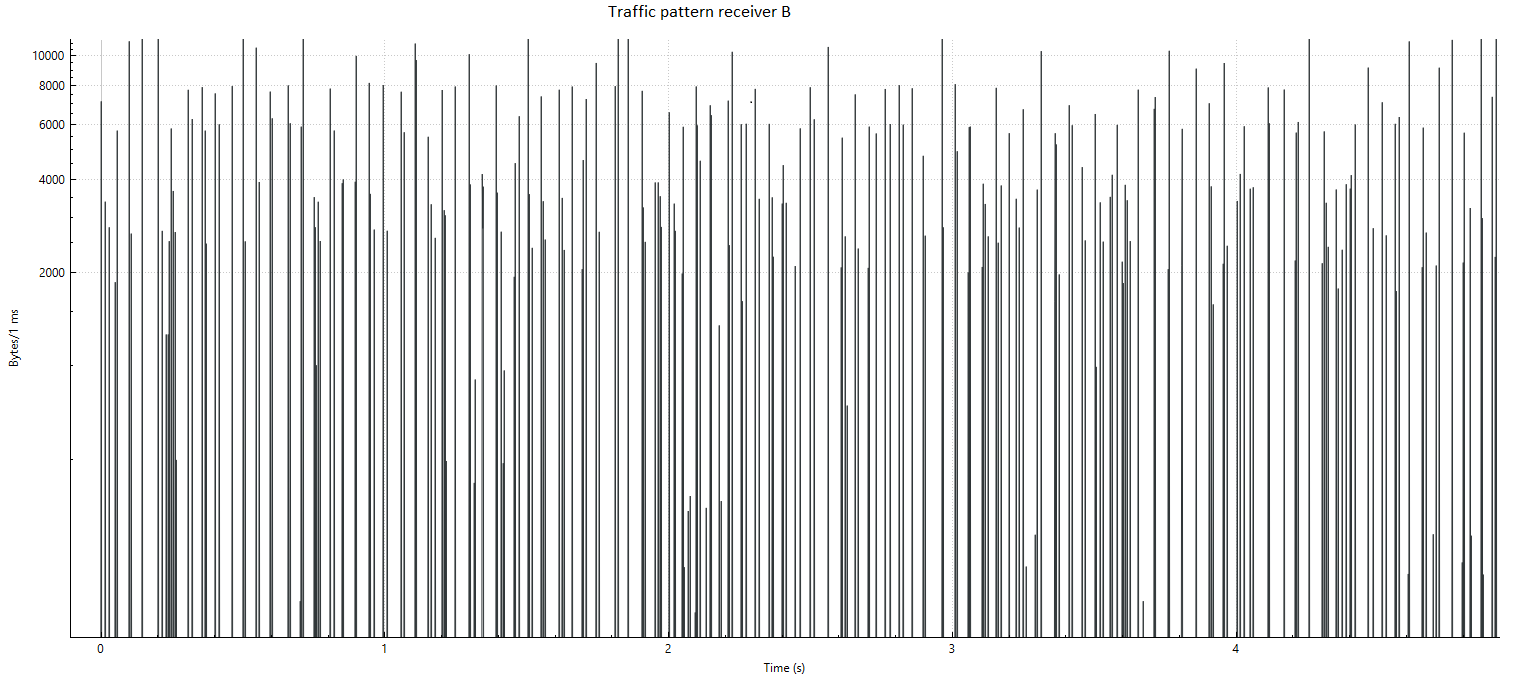


Figure 6.13.2.3.2-3: Traffic on machine B at the RTP receiver

Table 6.13.2.3.2-4: Snapshot of data from the sender, illustrating the inter-burst times

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Packet Number | Time | Source IP | Receiver IP |  | Length |
| 108 | 0.192814 | 192.168.178.229 | 192.168.178.110 | UDP | 766 |
| 109 | 0.192830 | 192.168.178.229 | 192.168.178.110 | UDP | 767 |
| 110 | 0.192846 | 192.168.178.229 | 192.168.178.110 | UDP | 845 |
| 111 | 0.192859 | 192.168.178.229 | 192.168.178.110 | UDP | 520 |
| 112 | 0.192872 | 192.168.178.229 | 192.168.178.110 | UDP | 448 |
| 113 | 0.210483 | 192.168.178.229 | 192.168.178.110 | UDP | 56 |
| 114 | 0.210584 | 192.168.178.229 | 192.168.178.110 | UDP | 565 |
| 115 | 0.210627 | 192.168.178.229 | 192.168.178.110 | UDP | 985 |
| 116 | 0.210655 | 192.168.178.229 | 192.168.178.110 | UDP | 944 |
| 117 | 0.210687 | 192.168.178.229 | 192.168.178.110 | UDP | 1110 |
| 118 | 0.210716 | 192.168.178.229 | 192.168.178.110 | UDP | 1027 |
| 119 | 0.210742 | 192.168.178.229 | 192.168.178.110 | UDP | 758 |
| 120 | 0.210770 | 192.168.178.229 | 192.168.178.110 | UDP | 722 |
| 121 | 0.256040 | 192.168.178.229 | 192.168.178.110 | UDP | 56 |
| 122 | 0.256087 | 192.168.178.229 | 192.168.178.110 | UDP | 317 |
| 123 | 0.256104 | 192.168.178.229 | 192.168.178.110 | UDP | 416 |
| 124 | 0.256125 | 192.168.178.229 | 192.168.178.110 | UDP | 584 |
| 125 | 0.256143 | 192.168.178.229 | 192.168.178.110 | UDP | 812 |

##### 6.13.2.3.3 GStreamer RTSP sender/receiver evaluation PCM audio + video

In this section the RTP video + audio sender and receiver are evaluated.

The video is as in previous section, while the audio is transmitted as PCM audio.

To collect traffic characteristics, the following setup is used:

- Two Machines A and B (Lenovo and Asus, first with AMD Ryzen 5 processor and second with Intel i5, 16 GB RAM/8 GB RAM are connected over Wifi network (2.4 GHz) setup by a Galaxy A65 5G phone using hotspot function. Machine A runs the real time-frame capture, encoding packetization and UDP transmission (sender pipeline from the previous clause). Machine B receives the packets and de-packetizes, decodes and renders the frame (receiver pipeline from the previous clause). Wireshark is used to collect the traffic data.

The results are shown in the figures below, the dynamic traffic characteristics bear similarity with those in the previous section, but the audio packets are much more frequent peaks with a smaller size.

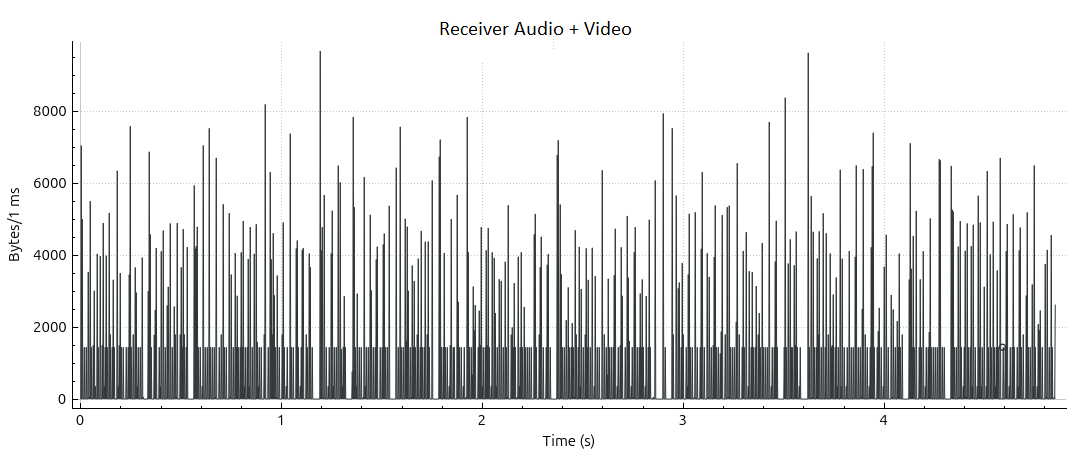


Figure 6.13.2.3.3-1: Receiver audio + video

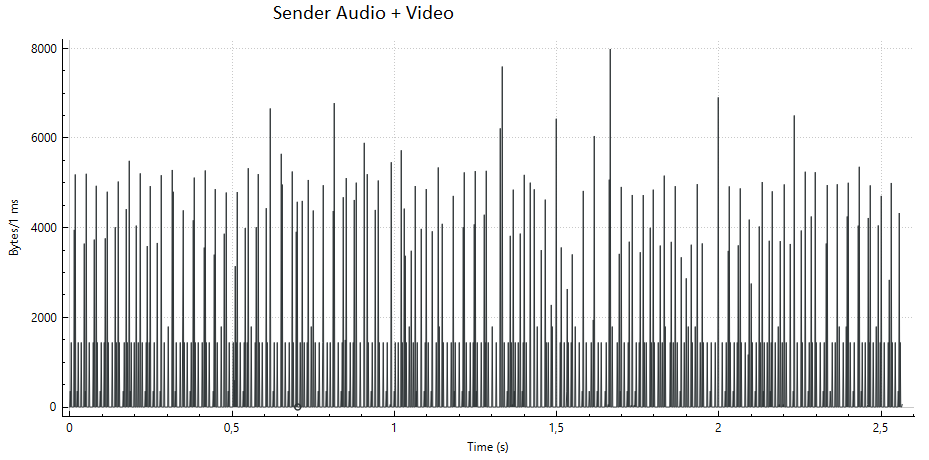


Figure 6.13.2.3.3-2: Sender audio + video

#### 6.13.2.4 Server-side senders

##### 6.13.2.4.1 Server sender implementation details

In this clause we consider server implementations of RTP senders using RTSP and/or RTP.

Both a popular commercial server and a popular open-source server implementation are evaluated.

The commercial server used for evaluation is Wowza Streaming Engine™ [40] (before Wowza Media Server™), one of the popular streaming servers that can support a wide range of transmission protocols. Wowza Streaming Engine™ is a Java application that runs as a service. The server can be used for a variety of live and on-demand streaming applications and it supports RTP/RTSP and low latency delivery.

The open-source server used in this evaluation is the popular Media MTX (before called simple RTSP server) [41], Media MTX is a real time media server/proxy that can support different protocols, including RTSP and RTP, but also WebRTC and RTMP [42]. The GitHub statistics on [41] show this as one of the most popular open source RTP/RTSP server implementations. It is possible, similar as in Wowza to ingest camera and other streams into the server for redistribution.

##### 6.13.2.4.2 Server side implementation evaluation

Wowza Streaming Engine™ version 4.8.27 was used deployed on a Lenovo laptop with 16 GB or RAM with AMD Ryzen 5. Another laptop with Intel i7-6600 2.6 GHz CPU and 16 GB of RAM is used to receive the stream from the server. A camera stream is ingested over RTP to the Wowza Streaming Engine™ using a GStreamer pipeline similar as the P2P sender in the previous section. In Wowza Streaming Engine™ a predefined session description file is used to ingest the stream. The stream is distributed using RTP/RTSP. The machines are connected over a 2.4 GHz Wifi network setup by a Galaxy A54 5G phone using the hotspot function. The receiver laptop uses VLC media player 3.0.8 for receiving the stream using RTSP and RTP.

Media MTX version 1.8.24 was used deployed on a Lenovo laptop with 16 GB or RAM with AMD Ryzen 5. Another laptop Microsoft surface with Intel i7-6600 2.6 GHz CPU and 16 GB of RAM was used to receive the stream. A camera stream is ingested over RTP to the wowza streaming engine using gstreamer, instead of RTP + SDP in this case WebRTC + WHIP was used for the ingest, which automizes the session description exchange needed for the ingest. The stream is distributed by Media MTX engine using RTP/RTSP. The machines are connected over a 2.4 GHz Wifi network setup by a Galaxy A54 5G phone using the hotspot function. The receiver laptop uses VLC media player 3.0.8 for receiving the stream over RTSP.

The flow of the experiment can be shown in Figure 6.13.2.4.2-1, GStreamer is used for the camera ingest to MTX Media or Wowza Streaming Engine™, then the data is transmitted to the client by the server, and Video Lan Client (VLC) is used to receive the stream and play it back. In this scenario RTSP + RTP is used.

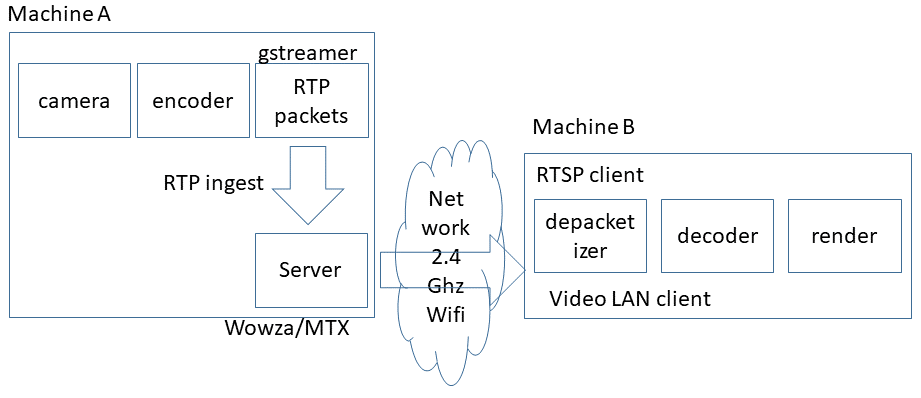


Figure 6.13.2.4.2-1: Setup of the experiment using server

The results for the different configurations are plotted in Figures 6.13.2.4.2-2 - 6.13.2.4.2-6 collecting traffic measurements. Only short intervals of time are measured, and the measurements from sender and receiver do not correspond timewise. The main point of investigation are the dynamic traffic characteristics relating to the burst size and periodicity, in each of the setups the burstiness corresponding to frame transmissions can be observed. It seems the MTX media the times between the bursts vary a bit more compared to the Wowza setup.

From the experiments it seems the lower framerate mostly results from the camera capture, tests with an artificial test source generating content at higher framerate results in the higher burst frequency (see Figure 6.13.2.4.2-4).

In Figure 6.13.2.4.2-6 an artificial video source is used to ingest to MTX with a higher framerate and quality. The results show that in this case the burst frequency of traffic transmitted by MTX media is also increased.

In addition, in 6.13.2.4.2-7 also shows when both audio (PCM) and video (H.264) are transmitted from MTX media server, in this case we see the audio bursts are more frequent and smaller in size compared to the video burst.

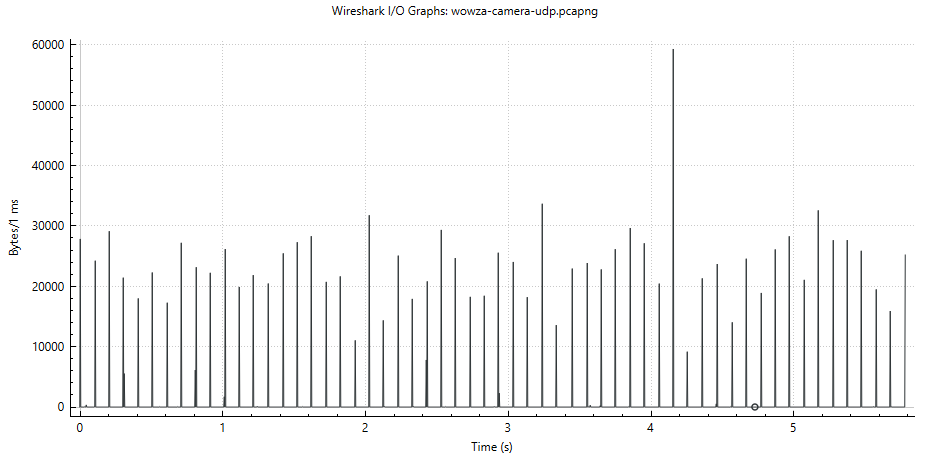


Figure 6.13.2.4.2-2: Wowza server side traffic sending to a VLC receiver using RTSP/RTP

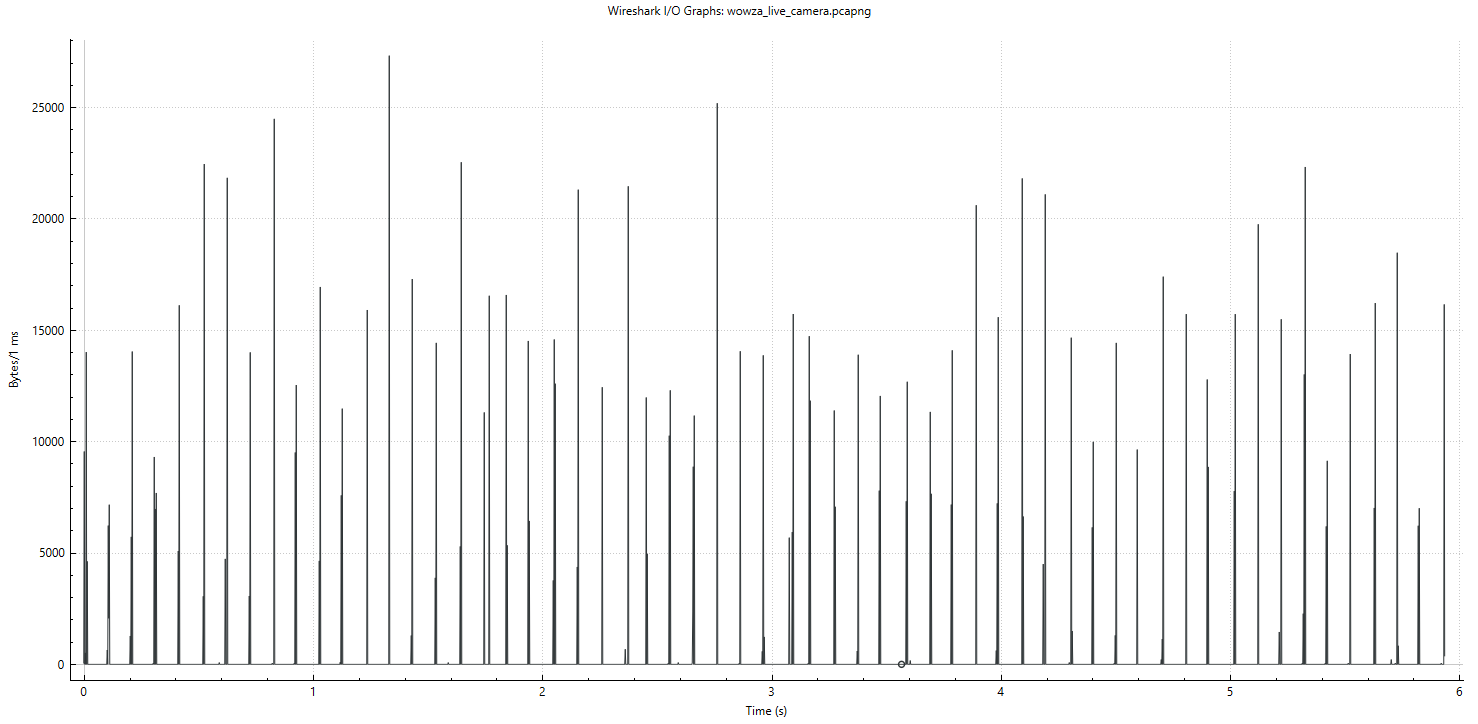


Figure 6.13.2.4.2-3: VLC receiver side traffic from received RTP packets from Wowza Streaming Engine™

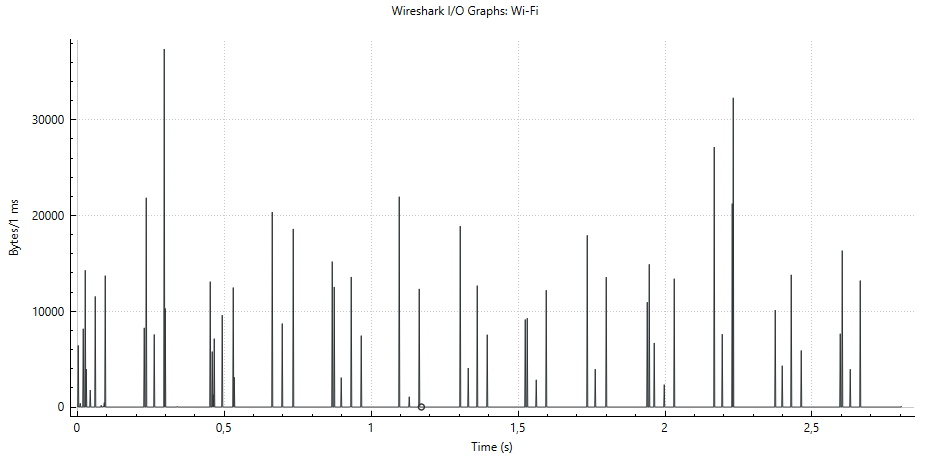


Figure 6.13.2.4.2-4: Camera RTP Traffic monitored from sender from MTX Media

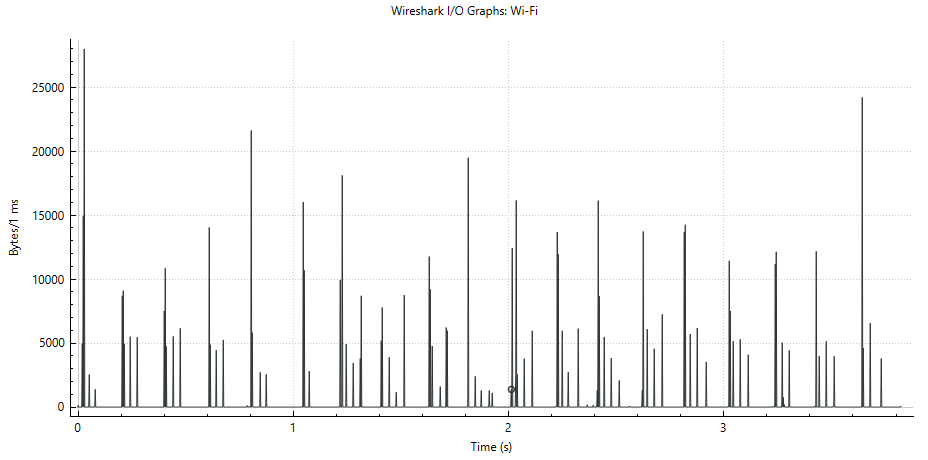


Figure 6.13.2.4.2-5: Camera RTP traffic monitored at VLC receiver from MTX Media

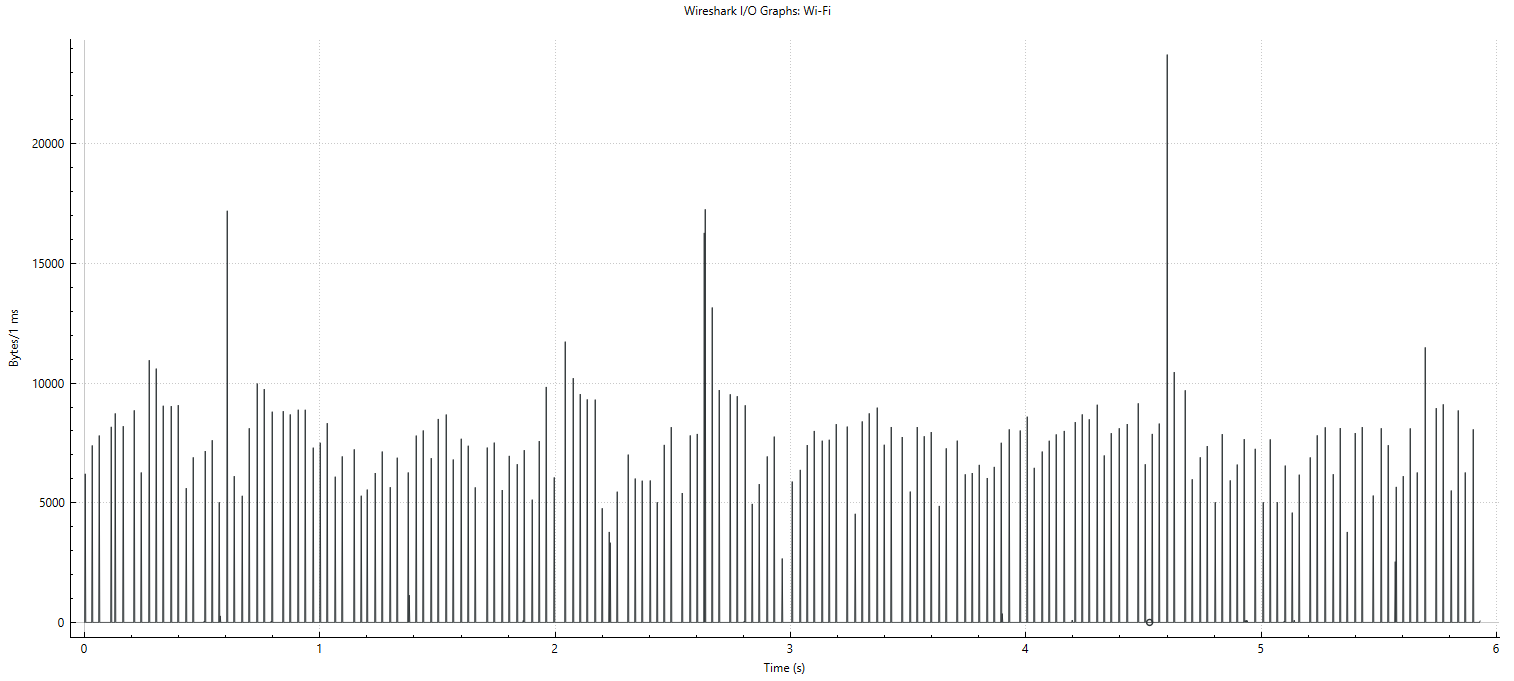


Figure 6.13.2.4.2-6: Synthetic test source RTP Traffic monitored from sender from MTX Media server

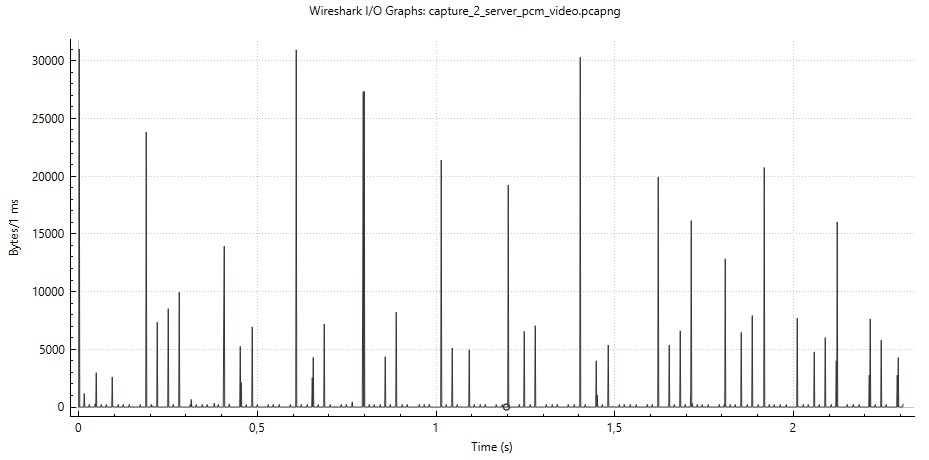


Figure 6.13.2.4.2-7: Traffic pattern from MTX sender server audio + video

#### 6.13.2.5 Aggregate Statistics

This clause presents some aggregate statistics for dynamic traffic characteristics are retrieved from the packet traces collected in this solution.

Table 6.13.2.5-1 illustrates the identified metrics.

Deterministic metrics, averages, standard deviations and percentages are considered.

The metrics are coded with a short abbreviation that is used when reporting in subsequent tables.

In Table 6.13.2.5-2 the scenarios described in the previous clauses are summarized and coded with a short abbreviation that is used then reporting the aggregate statistics in subsequent tables.

Table 6.13.2.5-3 illustrates the results of the metrics derived from the packet traces.

For brevity we can leave out the deterministic metrics bt, tbs, #bp, #lp, #bb and #lb that depend heavily on the length of the trace. Instead, the focus is on the ratio between lone packets/bytes and burst packets/bytes and other metrics that are independent of the length of the trace.

Some observations of the statistics in Table 6.13.2.5-3:

- A large percentage of the traffic and packets are part of a burst, highlighting the relevance of burst handling

- Burst durations in the traces are shorter than a millisecond

- PCM audio is tested, transmitted in lone packets, this increases the share of lone bytes (compared to more efficiently compressed audio)

Table 6.13.2.5-1: Metrics and statistics considered for dynamic traffic characteristics

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Code | Metric | Meaning | Additional notes | Statistics |
| Bt | Burst Time (Bt) | The time of the burst (corresponding to the first packet), measured in seconds | Bursts are measured as instantaneous transmission of 2 or more packets | Deterministic |
| Ibt | Inter Burst Time (Ibt) | The average duration between 2 consecutive bursts (burst), measured in seconds | This does not consider single ‘lone’ packets transmitted between the bursts | Average and Standard Deviation |
| Bd | Burst Duration (Bd) | The duration between start and end of a burst, measured in seconds | Measures the duration of the burst (in a maximum timeslot of 2 milliseconds ) | Average and Standard Deviation |
| Bs | Burst Size (Bs) | The Size of the Data Burst measured in bytes | Measures the size of the entire burst (in a maximum timeslot of 2 milliseconds) | Average and Standard Deviation |
| Ppb | Packets per burst (Ppb) | Number of packets in each burst | Only bursts of 2 packets or more are considered | Average and Standard deviation |
| #bp | Total Burst packets (#bp) | Number of packets in a burst | This is the number of packets in the trace that is part of a burst | Deterministic |
| #lp | Total Lone packets (#lp) | Number of packets not in a burst | This is the number of packets in the trace that is part not part of a burst | Deterministic |
| #bb | Total Burst Bytes (#bb) | Number of bytes that are part of a burst | This is the number of bytes that is part of a burst | Deterministic |
| #lb | Total Lone Bytes (#lb) | Number of bytes that are not part of a burst | This is the number of bytes in lone packets | Deterministic |
| %lp | Percentage of lone packets (%lp) | Ratio of lone packets to all packets |  | Percentage |
| %bp | Percentage of burst packets (%bp) | Ratio of burst packets to all packets |  | Percentage |
| %lb | Percentage of lone bytes (%lb) | Ratio of lone packet bytes to all bytes |  | Percentage |
| %bb | Percentage of burst bytes (%bb) | Ratio of burst packet bytes to all bytes |  | Percentage |

Table 6.13.2.5-2: Scenario descriptions and coding in solution #13

|  |  |
| --- | --- |
| Scenario code |  |
| w\_h | Windows hangout without network throttling duplex case (Figure 6.13.2.2.2-1) [1] |
| u\_h | Ubuntu Hangout with network throttling simplex case (Figure 6.13.2.2.2-2)[1] |
| g\_v\_a | Gstreamer video only measured on machine A (Figure 6.13.2.3.2-2) [1] |
| g\_v\_b | Gstreamer video only measured on machine B (Figure 6.13.2.3.2-3) [1] |
| g\_va\_a | Gstreamer pcm audio + video at receiver (Figure 6.13.2.3.3-1) [1] |
| g\_va\_b | Gstreamer pcm audio + video at sender (Figure 6.13.2.3.3-2)[1] |
| w\_s | Wowza at server measured at server (Figure 6.13.2.4.2-2) [1] |
| w\_r | Wowza stream measured at player receiver (Figure 6.13.2.4.2-3) [1] |
| m\_rv | MTX server measured vlc player receiver (Figure 6.13.2.4.2-5) [1] |
| m\_sav | MTX server video + PCM (Figure 6.13.2.4.2-7) [1] monitored at server |
| m\_rav | MTX server video + PCM (Figure 6.13.2.4.2-7) [1] monitored at vlc client |

Table 6.13.2.5-3: Statistical results of experiments in different scenarios in scenario #13

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Scenario  (direction) | ibt  (mean) | Ibt  (stdev) | Bd  (mean) | Bs  (mean) | Ppb  (mean) | %lp  (%) | %bp  (%) | %lb  (%) | %bb  (%) |
| w\_h (a) | 0.0352 | 0.01437 | 0.0002 | 3463 | 3.55 | 23.2 | 76.7 | 6.6 | 93.4 |
| w\_h (b) | 0.01592 | 0.019 | 0.00016 | 3756 | 3.3 | 19.2 | 80.8 | 17.4 | 82.6 |
| u\_h (a) | 0.03132 | 0.00873 | 0.00017 | 13895 | 12,3 | 40.2 | 59.8 | 22.6 | 77.4 |
| u\_h (b) | 0.42321 | 0.37918 | 0.00007 | 213.92 | 2 | 72.2 | 27.8 | 73.8 | 26.2 |
| g\_v\_a | 0.01642 | 0.01175 | 0.00052 | 2864 | 7.73 | 2.4 | 97.6 | 0.7 | 99.3 |
| g\_v\_b | 0.01796 | 0.00967 | 0.00029 | 3109 | 8.6 | 0.2 | 99.8 | 0.2 | 99.8 |
| g\_va\_a | 0.01252 | 0.0114 | 0.00071 | 4144 | 6.6 | 14.2 | 85.8 | 20.1 | 79.9 |
| g\_va\_b | 0.01414 | 0.00955 | 0.00033 | 4472 | 7.5 | 14.8 | 85.2 | 24.7 | 75.3 |
| w\_s (a) | 0.08732 | 0.03483 | 0.00079 | 24822 | 42.3 | 0.2 | 99.8 | 0.0 | 100.0 |
| w\_r (a) | 0.03778 | 0.04627 | 0.00054 | 10671 | 18.24 | 0.7 | 99.3 | 0.1 | 99.9 |
| m\_rv | 0.02749 | 0.0404 | 0.00084 | 6630 | 6.38 | 2.5 | 97.5 | 0.5 | 99.5 |
| m\_sav (a) | 0.04777 | 0.03838 | 0.00056 | 11906 | 12.18 | 12.6 | 87.4 | 3.3 | 96.7 |
| m\_rav (a) | 0.02573 | 0.03412 | 0.00086 | 6692 | 6.8 | 12.2 | 87.8 | 3.2 | 96.5 |

### 6.13.3 Discussion and conclusion

**Observation 1:**

The data from the GStreamer RTP and from WebRTC browser implementation show bursts of data being transmitted in short time intervals. There is no evidence that the encoder/packager gradually produces and transmits packets, as in the experiments only short bursts were observed. It could happen on a per frame(s) basis in a short time interval. It may also be that some frames are combined in a single burst, but this cannot be observed from the current results.

Similar observations were made when an intermediate server was used for redistributing the streams.

It is therefore expected to be possible for these applications to calculate the burst size a priori before sending it out. The added latency is expected to be limited to around a few milliseconds even in the worst case.

In this scenarios, a large percentage of the total traffic (packets/bytes) is part of a burst.

In this study only burst of size more than 1 packet were considered.

The results show that in cases a large percentage of the bytes/packets (generally over 90%) are transmitted in bursts.

This highlights that improvement of transmission of bursty traffic can be beneficial as this is such a large percentage of traffic.

The dynamic traffic characteristics could be the

NOTE 1: The latency is related to sending the packets on the network, not to the encoder or RTP packager generating the packets. Detailed study of the cause of the delay is FFS. The preliminary conclusion is that a few milliseconds of delay may be introduced is only a worst-case estimate.

**Observation 2:**

The inter-burst time interval seems regular, but not constant. Therefore, signaling of the time to next burst, if known by the application, may be suitable for signaling as a dynamic traffic characteristic.

Aggregate metrics for inter burst times have been derived (average/standard deviation), but these do not relate to the potential to predict or measure these in an application.

NOTE 2: This requires more study of different patterns and situations.

**Observation 3:**

It can be derived that when the network is not overloaded the WebRTC and GStreamer implementations do not differ too much and the influence of the paced sender module is limited.

In addition, when the network bandwidth changes all of a sudden, the rate control may adapt so well that the effect of the paced sender is not significant.

**Observation 4 (extra on P2P versus Server based):**

When comparing the end-end latency of the peer-to-peer setups to the server based setups, for the peer to peer setups latencies < 1 seconds are achieved while in the server routing setup latencies of around 3 seconds are observed.

**Conclusion:**

Short periodic traffic bursts in short intervals occur in typical real time conversational applications using real-time video + audio.

Given the observed traffic behaviour and the observed application behaviour it seems achievable to include information about a burst size before it is being sent out as the durations of the burst are in the order of less than1-2 milliseconds (mostly less than 1 millisecond), but it may require some changes in the sender implementation to achieve this.

A large percentage of the transmitted bytes/packets are part of a data burst.

### 6.13.4 Conclusion

Given the observed behaviour of common RTP Sender implementations, the conclusion and recommendation can be made as following:

- Extend the signaling of dynamic traffic characteristics to signal data burst size and time to next burst in the RTP Header Extension.

- Complementary signaling/additions to the above RTP Header Extension.

- Transmission of bursty traffic is important as in RTP senders a large percentage of the bytes/packets (> 90%) is transmitted in bursts.

## 6.14 Solution #14: Candidate RTCP messages and RTP header extensions to support XR services in 5G

### 6.14.1 Key Issue mapping

This maps to Key Issue #7.

### 6.14.2 Description

The following RTCP messages and RTP header extensions are relevant for XR services.

#### 6.14.2.1 RTCP messages

To understand the RTCP messages that may be used for supporting XR applications, we need to know what RTCP messages have been defined in standard specifications such as IETF RFCs and what RTCP messages have been used in commercial systems. For this purpose, we look at the WebRTC implementation [28] and IETF RFCs.

RTCP messages defined in RFCs:

- **Receiver report (RR)** (RFC 3550 [44]): The packet type (PT) is 201. It provides reception quality feedback to the other RTP endpoint on a per source SSRC basis. Among the reported information is

1) the fraction lost: The fraction of RTP data packets from a source SSRC lost since the previous SR or RR packet was sent

2) cumulative number of packets lost: The total number of RTP data packets from a source SSRC that have been lost since the beginning of reception

3) interarrival jitter: An estimate of the statistical variance of the RTP data packet interarrival time

- **Sender report (SR)** (RFC 3550 [44]): The packet type (PT) is 200. It is the same as the RR except that it carries additional 20 bytes of information about the RTP endpoint that originates this report.

- **Application-Defined Packet** (APP) (RFC 3550 [44]): The PT is 204. It is intended for experimental use as new applications and new features are developed, without requiring packet type value registration.

- **Feedback messages**:

1) **Transport layer FeedBack messages** (RTPFB): The PT value is 205.

a) **Generic NACK** (RFC 4585 [33]): The PT value is 205 and the format type (FMT) value is 1. The Generic NACK is used to indicate RTP packet losses, identified by the means of a packet identifier and a bit mask.

b) **TMMBR** (RFC 5104 [45]): The PT value is 205 and the FMT value is 3. The Temporary Maximum Media Stream Bit Rate Request message is used to notify the media sender about the changes in downlink bandwidth allocation with a new current maximum bitrate.

c) **TMMBN** (RFC 5104 [45]): The PT value is 205 and the FMT value is 4. The Temporary Maximum Media Stream Bit Rate Notification message is used to notify the media receiver about the adjusted new media bitrate by a media sender. This message is sent in response to a received TMMBR message.

d) **RTP-ECN-FB** (RFC 6679 [46]): The PT value is 205 and the FMT value is 8. The RTP Explicit Congestion Notification (ECN) feedback message reports reception of an ECN-CE-marked RTP packet so that the RTP sender may perform congestion control.

e) **RTP Congestion Control Feedback** (CCFB) (RFC 8888 [47]): The PT value is 205 and the FMT value is 11. The congestion control feedback message is used to transmit the status of each RTP packet received at the RTP receiver along with the RTP sequence number, packet arrival time and packet Explicit Congestion Notification (ECN) Marking. This message specifies a common RTCP feedback packet format that can be used by Network-Assisted Dynamic Adaptation (NADA) (RFC 8698 [29]), Self-Clocked Rate Adaptation for Multimedia (SCReAM) (RFC 8298 [22]), Google Congestion Control [27], and Shared Bottleneck Detection (RFC 8382 [48]) congestion control algorithms.  
The format and semantics of the RTCP CCFB packet is as 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

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

|V=2|P| FMT=11 | PT = 205 | length |

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

| SSRC of RTCP packet sender |

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

| SSRC of 1st RTP Stream |

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

| begin\_seq | num\_reports |

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

|R|ECN| Arrival time offset | ... .

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

. .

. .

. .

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

| SSRC of nth RTP Stream |

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

| begin\_seq | num\_reports |

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

|R|ECN| Arrival time offset | ... |

. .

. .

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

| Report Timestamp (32 bits) |

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

- begin\_seq [16 bits]: The first sequence number that this feedback message reports on.

- Received (R) [1 bit]: A boolean that indicates whether the packet was received. 0 indicates that the packet was not yet received and the subsequent 15 bits (ECN and ATO) in this 16-bit packet metric block are also set to 0 and MUST be ignored. 1 indicates that the packet was received and the subsequent bits in the block need to be parsed.

- num\_reports [16 bits]: The number of RTP packets reported in this feedback message. The report block contains a 16-bit packet metric block for each RTP packet that has a sequence number in the range begin\_seq to begin\_seq+num\_reports inclusive.

- ECN [2 bits]: The echoed ECN mark of the packet. These bits are set to 00 if not received or if ECN is not used.

- Arrival time offset (ATO) [13 bits]: The arrival time of the RTP packet at the receiver, as an offset before the time represented by the Report Timestamp (RTS) field of this RTCP congestion control feedback report. The ATO field is in units of 1/1024 seconds (this unit is chosen to give exact offsets from the RTS field) so, for example, an ATO value of 512 indicates that the corresponding RTP packet arrived exactly half a second before the time instant represented by the RTS field.

- Report Timestamp (RTS) [32 bits]: This value denotes the time instant on which this packet is reporting and is the instant from which the arrival time offset values are calculated. The value of the RTS field is derived from the same clock used to generate the NTP timestamp field in RTCP Sender Report (SR) packets.

f) **Transport-wide feedback** [43]: The PT value is 205 and the FMT value is 15. This feeds back information about each packet received with a transport-wide packet sequence number.

     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

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

    |V=2|P|  FMT=15 |    PT=205     |           length              |

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

  0 |                     SSRC of packet sender                     |

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

  4 |                      SSRC of media source                     |

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

  8 |      base sequence number     |      packet status count      |

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

12 |                 reference time                | fb pkt. count |

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

16 |          packet chunk         |         packet chunk          |

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

    .                                                               .

    .                                                               .

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

    |         packet chunk          |  recv delta   |  recv delta   |

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

    .                                                               .

    .                                                               .

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

    |           recv delta          |  recv delta   | zero padding  |

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

- base sequence number: The transport-wide sequence number of the first packet in this feedback.

- reference time: it indicates an absolute reference time in some (unknown) time base chosen by the sender of the feedback packets. The first recv delta in this packet is relative to the reference time.

- packet chunk: A list of packet status chunks, indicating the status of one or more packets starting with the one identified by base sequence number.

- recv delta: it represents a time interval in units of 0.25 ms for a packet indicated in the packet chunk relative to the reference time.

2) **Payload-specific FeedBack Messages** (PSFB): The PT value is 206.

a) **Picture Loss Indication** (PLI) (RFC 4585 [33]): The PT is 206 and the FMT is 1. It indicates the loss of an undefined amount of coded video data belonging to one or more pictures.

b) **Full Intra Request** (FIR) (RFC 5104 [45]): The PT value is 206 and the FMT value is 4. The FIR indicates the request of a full intra or Instantaneous Decoder Refresh picture from the media sender.

c) **Region of Interest** (ROI) (TS 26.114 [32]): The PT value is 206 and the FMT value is 9. The ROI message indicates the request by the media receiver for an interested region of the media.

d) **Viewport** (VP)(TS 26.114 [32]): The PT value is 206 and the FMT value is 11. The ROI message indicates the request by the media receiver for a region of the media in an interested viewport.

- **RTCP XR Reports** (RFC 3611 [49]): The RTCP XR report is identified by PT equal to 207, which refers to an extended report block message. The block type (BT) field defined in RFC 3611 is used to identify the block format.

#### 6.14.2.2 RTP header extensions

For application bitrate adaptation and congestion control, it is important for the network to understand the state of the network, i.e., whether the network is in congestion or not. Many congestion control algorithms, e.g., Google congestion control algorithms [28], NADA [29] and SCReAMv2 [30], use the queueing delay as a signal of network congestion. Therefore, it is important to measure the delays and make the measurements available to the RTP sender in an efficient manner.

TS 26.522 [2] defined two RTP header extensions for in-band end-to-end delay measurement. The first RTP header extension that carries only one timestamp, also known as the "Absolute Sender Time" RTP header extension, is already implemented in WebRTC [28]. The second RTP header extension that carries three timestamps returns the measured one-way delay in the direction from the sender to the receiver back to the sender. The current implementation in WebRTC uses RTCP messages to carry the one-way delay back to the sender and that may introduce large delay due to the RTCP bandwidth limitation or large overhead due to the additional IP/UDP packet headers for a separate packet.

TS 26.522 [2] defines an RTP header extension for XR Pose which can be used for signaling either a 6DoF or 3DoF XR pose. The HE can be used either by an RTP receiver to indicate the XR pose used for rendering the media (rendered pose) or by an RTP sender to indicate to an RTP receiver the XR pose to be rendered.

NOTE: Other RTCP messages and RTCP header extensions in 3GPP TS 26.522 [2] and IETF RFCs may also be considered.

## 6.15 Solution #15: PSI signaling for lone PDUs

6.15.1 Key Issue mapping

This solution addresses the key issue #2.

### 6.15.2 Description

#### 6.15.2.1 Background

As of Rel-18, there is no mechanism to mark PDUs carrying protocol data other than RTP. Thus, PDUs belonging to protocols such as RTCP, STUN, etc. cannot be marked.

In Rel-18, SA2 has agreed that the PSA UPF marks, in the downlink, each N6-unmarked PDU ("lone PDU") with PDU Set Information into a PDU Set. If the UPF receives a PDU that does not belong to a PDU Set based on Protocol Description for PDU Set identification, the UPF still maps it to a PDU Set and determines the PDU Set Information by implementation-specific means.

**Observation 1:** **PDUs of non-RTP protocols (e.g. RTCP) are mapped by the UPF into PDU Sets. The associated PDU Set Information is determined by the UPF.**

When the PDU Set Information is not provided by the sender in an RTP HE, the UPF may be able to reliably obtain some parts of the PDU Set Information based on the UPF implementation. Annex A of TS 26.522 [2] describes how a Network Function can obtain the PDU Set Information from the RTP header and RTP payload, respectively.

When the RTP HE for PDU Set marking is not available, the UPF may derive some parts of the PDU Set Information from the (S)RTP header, as described in clause A.2.1 of TS 26.522 [2]. However, PDU Set Importance (PSI) cannot be obtained and PDU Set Size (PSSize) can only be obtained after the reception of the last PDU of the PDU Set. Also, the derivation in clause A.2.1 only applies for the case when a PDU Set is a video frame. It is not applicable to e.g. audio PDU Sets or PDU Sets that are video slices.

When the RTP payload is not encrypted (i.e., SRTP is not used), the UPF may derive some parts of the PDU Set Information from the RTP payload. Clause A.2.2 of TS 26.522 [2] describes how this can be done for RTP payloads carrying H.264/AVC and H.265/HEVC coded bitstreams. In summary, the UPF needs to parse the NAL unit header, which is the first one and two byte(s) of the RTP payload for H.264/AVC and H.265/HEVC, respectively. Similar to the derivation from the RTP header, the PSSize can only be obtained after the reception of all PDUs of the PDU Set. PSI can be obtained by using the same logic the RTP sender uses to populate the RTP HE for PDU Set marking (i.e., parsing the NAL unit header), as described in the relevant guidelines in clause 4.2.6.2 of TS 26.522 [2]. However, this means more operations and thus more processing load for the UPF since it would need to check one or more of the NAL unit header fields. This is not feasible considering that the UPF processes data from several endpoints simultaneously under tight latency constraints.

**Observation 2: For RTP PDUs, if the RTP HE for PDU Set marking is not present, the UPF may derive some parts of the PDU Set Information from the RTP header or RTP payload (if unencrypted), albeit with some restrictions. In particular, PSI cannot be obtained from the RTP header, and deriving PSI from the RTP payload imposes a significant processing overhead for the UPF given its high processing load and tight latency constraints.**

Signaling the PDU Set Information in an RTP HE or deriving it from the RTP header/payload is not possible for PDUs carrying protocol data other than RTP (e.g. RTCP). Such PDUs are considered as lone PDUs by the UPF and placed into a PDU Set in a way that is determined by the UPF implementation. The UPF also has to define the PDU Set Information for the PDU Sets containing the lone PDUs. For some parts of the PDU Set Information (PSN, PSSN, End PDU), this operation is straightforward.

For example, consider an RTCP PDU that is placed by the UPF into its own PDU Set (i.e., the PDU Set contains only that PDU). It is described below whether/how the PDU Set Information can be obtained for that PDU Set.

- Number of PDUs in the PDU Set (NPDS) is set to 1.

- PSN is set to 0 and the End PDU flag is set to 1 (since NPDS=1).

- PSSN is trickier since it depends on the transmission order of the PDU Set by the sender and is monotonically increased by the sender by 1 for each subsequent PDU Set. Since there is no gap between the PSSNs assigned by the sender for the RTP PDUs, the UPF would have to either assign an existing PSSN (i.e., a PSSN that is already used by another PDU Set) to the new PDU Set that contains the lone PDU, or a predetermined value that indicates that PSSN is undefined for that PDU Set.

- PSSize is equal to the size of the RTCP packet since that packet would be the only PDU in the PDU Set.

- To determine the PSI, the UPF must resort to preconfiguration (i.e. use a pre-defined value) since it has no means to obtain the PSI from the packet header or payload for non-RTP PDUs.

Any default PSI setting at the UPF may not be accurate since the importance of different PDU Sets is application- and codec-dependent. For example, some RTCP message types may be considered more important for low latency applications. In another example, RTCP feedback messages for viewport signaling may be crucial for the functionality of an immersive application, and thus it would be beneficial to be able to indicate a low PSI value for the RTCP packets carrying those.

NOTE 1: Other potential benefits and limitations of PDU Set handling for non-RTP protocol types (e.g. RTCP, STUN) is FFS.

**Observation 3: For lone PDUs, some parts of the PDU Set Information must be determined by the UPF. However, the UPF cannot reliably determine the PSI and may only assign a pre-defined PSI value (e.g. by the network operator). Sender applications are in the best position to determine the PSI.**

NOTE 2: Whether the UPF can reliably determine PDU Set Information for lone PDUs based on local configuration needs to be verified with SA2.

#### 6.15.2.2 Solution description

In this solution, the Media Application Provider defines a mapping between a set of PSI values and non-RTP protocols that are used in the media delivery session.

The mapping can be provided using the RTC provisioning feature (TS 26.510 [50], clause 5.2.10) of the media delivery session. In an example implementation, the Media Application Provider adds a property *lonePduInfoList* to the *RTCConfiguration* resource. The property *lonePduInfoList* contains an array of *lonePduInfo* objects as defined below.

Table 6.15.2.2-1: Definition of *lonePduInfo* object

| Property name | Data Type | Cardinality | Description |
| --- | --- | --- | --- |
| protocol | string | 1..1 | Protocol information such as RTCP, STUN, etc. |
| packetType | integer | 0..1 | Packet type specific to the protocol. |
| feedbackMsgType | integer | 0..1 | RTCP feedback messages type [33]. Can only be present if the protocol is RTCP. |
| pduSetImportance | integer | 1..1 | PSI value between 0 and 15 (inclusive). |

If a *lonePduInfoList* is provided, the Media AF extends the*mediaTransport‌Parameters* property of the Application Flow Description that it has received from the Media Session Handler with the information in the *lonePduInfoList*. The Media AF then sends the Application Flow Description to the 5G Core, where the*mediaTransport‌Parameters* (asProtocol Description) is passed to the UPF. For example, the Protocol Description may then have the following structure after addition of the property *lonePduInfoList*:

{ "transportProto": "RTP",

"rtpPayloadInfoList": [{"rtpPayloadFormat": "H265", "rtpPayloadTypeList": [96]}],

"**lonePduInfoList"**: [{"protocol": "RTCP", "packetType": 206, "feedbackMsgType": X, psi: 2},

{"protocol": "RTCP", "packetType": 207, psi: 10}],

}

In the example, the first *lonePduInfo* object provides the PSI mapping for RTCP packets that contain viewport feedback messages TS 26.114 [32], clause Y.7.2. The RTCP feedback message is identified by Packet Type = 206, which refers to payload-specific feedback message (RFC 4585 [33]). FMT (feedback message type) is set to the value ‘11’ for viewport feedback messages. The second *lonePduInfo* object provides the PSI mapping for RTCP Extended Reports (XR) messages identified by Packet Type = 207 (RFC 3611 [49]). In this example, the Media Application Provider chooses to assign PSI=2 to viewport feedback messages and PSI=10 to RTCP XR messages.

NOTE 3: There may be PDUs carrying other protocol data such as STUN in the media delivery session. Whether/how a PSI mapping is provided for such PDUs is FFS.

After receiving the Protocol Description, the UPF can determine the PSI for lone PDUs based on the information provided in each *lonePduInfo* object.

NOTE 4: Potential benefits and limitations of PDU Set handling for media streams that do not require both high bitrate and low latency (e.g., audio, haptics) are FFS.

#### 6.15.2.3 Analysis of the solution

The benefit of the proposed solution is that the UPF does not have to rely on a pre-defined value (e.g. provided by the network operator) to determine the PSI for lone PDUs and can make a more reliable decision based on a PSI mapping provided by the Media Application Provider.

In terms of UPF processing, complexity is not increased because the UPF only needs to inspect the packet headers (e.g. the RTCP header) to check for packet type.

Benefits of the solution are summarized below.

- More dynamic solution as it allows applications to set PSI value for specific unmarked traffic instead of using a fixed value for all applications pre-configured at the UPF.

- More reliable as applications are in a better position to determine the importance of PDU Sets.

- Reuses the RTCConfiguration mechanism. Requires the UPF to determine the PSI value for lone PDUs at the start of session based on RTCConfiguration.

- Extensible to other types of unmarked packets in the future.

The impacted entities in the 5G System are:

**Real-time Media Provisioning API:**

- *RTCConfiguration* provided by the Media Application Provider is extended with the lone PDU information.

**Media AF:**

- Receives the extended *RTCConfiguration* and adds it to the Application Flow Description.

**UPF:**

- Receives the extended Protocol Description and parses the *lonePduInfoList* property to retrieve the PSI mapping for lone PDUs.

NOTE 5: This solution requires coordination with SA2.

## 6.16 Solution #16: RTP header extension for dynamic traffic characteristics

### 6.16.1 Key Issue mapping

This is a solution to KI #12 Enhancements of Data Burst Marking.

### 6.16.2 Description

A data burst indicates a set of multiple PDUs generated and sent in a short period of time as defined in clause 3.1 of TS 23.501 [3]. Data burst is a common transmission characteristic in communication networks.

The traffic characteristics regarding the data burst transmission could be beneficial for the 5GS network, e.g., power saving and efficient radio resource management. In Release 18, the End of Data Burst indication has been introduced to enable the UE power saving in the 5GS, i.e., the NG-RAN node can configure to move a UE into CDRX for power saving after transmitting the end PDU of the data burst. In Release 19, the data burst size has been concluded to enable the RAN radio resource management as described in clause 8.5 of TR 23.700-70 [6].

For marking dynamic traffic characteristics, the RTP HE for Dynamic Traffic Characteristics is defined in this clause.

A new RTP HE is proposed for the following reasons:

1) Avoiding compatibility issues with release 18 Header Extension for PDU Set marking.

2) Avoiding overhead, RTP HE need not be present in each RTP packet, but for Release 18 Header Extension it is common understanding that usually each RTP packet is marked. The information for dynamic traffic characteristics on the other hand is specifically useful in specific packets at the beginning of a traffic pattern.

Dynamic Traffic Characteristics marking can be performed by an RTP sender, such as an Application Server (e.g., MRF), a sender UE that sends media to an RTP receiver, such as a UE, or other 5G network components.

Endpoints that support the RTP HE for Dynamic Traffic Characteristics can support both RTP HE formats (i.e., the one-byte and the two-byte formats) according to RFC 8285 [7].

If the RTP HE for Dynamic Traffic Characteristics is the only RTP HE used, the endpoints can use the 1-byte header format. If other 2-byte RTP HE elements are used in the same RTP stream, then the 2-byte header can be used, unless the "a=extmap-allow-mixed" is successfully negotiated through SDP offer/answer, as described by RFC 8285 [7].

NOTE: The headers are not shown with padding as this depends on other prospective extension elements in use, as per RFC 8285 [7] alignment specifications.

The IANA registration information for the RTP HE for RTP HE for Dynamic Traffic Characteristics is in 6.16.8.

#### 6.16.2.1 Intended usage in 5GS

The solution of adding dynamic traffic characteristics serves the following key use case:

1) Based on the SDP negotiation, the RTP HE for Dynamic Traffic Characteristics is enabled. The RTP Sender or Application Server adds header extension of a dynamic traffic characteristic in the first few packets of a data burst.

2) The dynamic traffic characteristics header is added by the packet sender potentially for groups of packets to be sent, this may include multiplexed RTP or Multiplexed RTP/RTCP traffic. The sender, may add information such as the Burst Size of the group of packets to be transmitted, or based on its own internal scheduling the time until the next burst can be sent.

3) The UPF detects packets that include the Header Extension for Dynamic traffic characteristics and marks the dynamic traffic characteristics into the GTP-U header of downlink packets, including the End of Data Burst indication and data burst size.

4) As concluded in clause 8.5 of TR 23.700-70 [6], the data burst size carried in the RTP HE can be identified by the UPF and then further sent to the NG-RAN via the GTP-U header to assist the radio resource management. The procedure is as follows:

a) The (RTC-)AF may provision the Protocol Description to PCF directly over N5 interface or via NEF over N33. The Protocol Description indicates that the RTP HE for Dynamic Traffic Characteristics is enabled.

b) PCF may provision the Protocol Description within the PCC rules based on the information provided by the AF and/or the local operator policies.

c) SMF requests the UPF to detect and mark the burst size of the data burst and mark it in the GTP-U header of the first few PDUs in downlink, according to the PCC rule and/or the local operator policies.

d) UPF identifies the burst size of a data burst in the downlink traffic based on the RTP HE according to the Protocol Description and provides the data burst size to the RAN in the GTP-U header of the first few PDUs of a data burst to assist radio resource management.

e) RAN efficiently optimizes the radio resource for the timely data burst transmission based on the data burst size in the GTP-U header.

#### 6.16.2.3 One-byte RTP header extension format

The one-byte RTP HE 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 | R |D| RR | BSSize

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

| TTNB |

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

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

The two-byte RTP HE 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 | R |**D**| RR | BSSize

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

| TTNB |

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

#### 6.16.2.5 Semantics

The semantics of the fields of the RTP HE for PDU Set marking are defined as follows:

- **End of Data Burst [D] (1 bit):** This field is a flag that can be set to 1 for the last PDU of a Data Burst. It can be set to 0 for all other PDUs. A Data Burst may consist of one or more PDU Sets.

- **Reserved [R] 3 bits):** This field is reserved for future usage (e.g., dynamic burst indication). It can be set to 0 by the RTP sender and can be ignored by the RTP receiver.

- **Reserved [RR] 4 bits):** This field is reserved for future usage (e.g., dynamic burst indication). It can be set to 0 by the RTP sender and can be ignored by the RTP receiver.

- **Burst Size [BSSize] (24 bits):** The Burst Size indicates the total size of the burst to be transmitted. The burst size corresponds to the size of the data burst. If the burst size is not known it is set to 0.

NOTE 1: If a packager generates all packets of the burst at once, no additional delay is introduced when setting the burst size, as the packets can be marked with the complete burst size. If this is not the case a delay as large as the burst duration could be introduced by marking the entire burst. Therefore, this approach may not be suitable for all types of packagers/encoders, especially those that gradually produce packets additional latency may be introduced if the size is not known in advance.

- **Time To Next Burst [TTNB] (16 bits):** Indicates the approximate time (within a 1-5 millisecond range) to the next burst in milliseconds. If the time to next burst is not known it is set to 0. This time is relative to the time of the current burst that is the send time in milliseconds of the current burst, taking the packet in the middle of the burst as reference.

NOTE 2: Inaccuracy on the TTNB may occur due to different reasons such as re-ordering or unknown, this number is indicative and can be accurate within 1-5 ms range. An addition solution will be provided to show some typical XR traffic and related burst showing that for Real Time A/V traffic bursts of 1-2 milliseconds can be common.

NOTE 3: This solution has some overlap with the solution for Data burst marking in R18 and PDU Set marking, more discussion is needed on the benefits. As PDU Set marking requires marking each packet while traffic characteristics marking does not, this separate solution is proposed.

NOTE 4: The introduction of this header extension may need some alignment with other working groups such as SA2 and/or RAN2.

NOTE 5: The anchor time of the TTNB is expected to be for further study in case the solution is adopted as a basis for normative work.

NOTE 6: Additional optional fields of this Header Extension are for further study.

NOTE 7: The layout of the dynamic traffic characteristics header extension mimics the RTP Header for PDU set marking enabling re-using of parsing mechanisms.

#### 6.16.2.6 SDP Signaling

An RTP sender capable of sending RTP HE for Dynamic Traffic Characteristics can use the SDP extmap attribute for RTP HE for RTP HE for Dynamic Traffic Characteristics in the media description of the RTP stream(s) carrying the RTP HE for RTP HE for Dynamic Traffic Characteristics. An RTP receiver that does not support RTP HE for Dynamic Traffic Characteristics can ignore that RTP HE when included. The signaling of the Dynamic Traffic Characteristics RTP HE can follow the SDP signaling design and the syntax and semantics of the "extmap" attribute as outlined in RFC 8285.The URN for the PDU Set marking can be set to "**urn:3gpp:dynamic-traffic-characteristics:rel-19**".

The ABNF syntax for the extmap attribute for the signaling of RTP HE for PDU Set marking is defined as follows, extending the ABNF in RFC 8285:

*extensionname = "urn:3gpp:dynamic-traffic-characteristics:rel-19"*

*format = "short" / "long"*

The extension attributes have the following semantics:

- format: indicates if the RTP HE for Dynamic Traffic Characteristics uses the 1-byte (short) or the 2-byte (long) format. This extension attribute can not be included more than once.

NOTE: Regardless if this extension attribute is present or not, the use of long or short format is determined as described by section 4.1.2 of RFC 8285, i.e., based on what format other RTP HEs use in the same RTP session, unless both endpoints announced support for handling mixed format with "a=extmap-allow-mixed" as described by section 6 of RFC 8285 [7].

Below is an example:

a=extmap:7 dynamic-traffic-characteristics:rel-19 long

#### 6.16.2.7 Guidelines fordynamic traffic characteristics signaling

It is recommended that the first several RTP packets and the last packets contain the dynamic traffic characteristics traffic signaling. In addition, some additional RTP packets may contain the RTP Header Extension for dynamic traffic characteristics.

It is recommended that the application signals the presence of RTP HE for dynamic traffic characteristics out of band using SDP signaling as defined in 6.16.2.6.

In addition, dynamic traffic characteristics can only be used if the generated data can be marked for such characteristics, i.e. it contains burst and potentially some periodicity information or it knows the timing to a next data burst.

A sender, that is scheduling to send out a group of packets, may calculate the size of the group of the packets, and then add the overhead of adding the RTP Header and then update the packets to include the RTP Header Extension for dynamic traffic characteristics.

#### 6.16.2.8 Proposed Annex D.3

The desired extension naming URI:

urn:3gpp:dynamic-traffic-characteristics:rel-19

A formal reference to the publicly available specification:

[TS 26.522]

A short phrase describing the function of the extension:

Marking of dynamic traffic characteristics such as burst size and time to next burst

Contact information for the organization or person making the registration:

3GPP Specifications Manager

3gppContact@etsi.org

+33 (0)492944200

#### 6.16.2.9 Discussion of the solution

This solution presents a way forward to enable dynamic traffic characteristics signaling in release 19. The main advantages are that:

1) Backward compatibility is achieved by not changing the Release 18 HE for PDU Set marking. The default behaviour is to ignore unknown RTP headers. By not changing the release 18 RTP HE for PDU Set marking this can still be used the same way. In case efficient data burst marking is needed in the first packet as requested by RAN2 the current solution can be used.

2) Sparsity and reduced overhead are achieved as recommended in RFC 8285. Not every RTP packet has to carry the Header Extension for Dynamic Traffic Characteristics.

3) Can be implemented separately and independently from the PDU Set marking Header Extension, a sender or scheduler that sends out a group of packets in a burst such as multiple PDU Sets, can apply marking for the traffic characteristics.

4) Well defined usage within the 5G System context is documented.

NOTE: How to combine bursts sent over different IP Tuples is for further study.

## 6.17 Solution #17: Analysis of AL-FEC awareness in 3GPP

### 6.17.1 Key Issue mapping

This solution addresses key issue #4 on AL-FEC awareness for PDU Set handling and Key Issue #3: Enhancements for application-layer FEC support.

### 6.17.2 Description

#### 6.17.2.1 Common attributes of AL-FEC deployments

AL-FEC codecs, as the schemes documented in Solution #5, are used in ensuring low latency media delivery for networks with bursty losses and RTT delays comparable or higher than the jitter buffer delay constraints of an application. AL-FEC is not an exclusive packet loss mechanism and may be used interchangeably with retransmissions (e.g., as documented in Solution #8 in section 6.8 as media source bit rate adaptation or other flow control mechanisms. AL-FEC encoding is in fact utilized in practice in supplement to RTCP NACK indication and the AL-FEC redundancy level is usually dynamically adapted [25], [26] to network conditions.

The advantage of AL-FEC over other schemes is that error recovery is proactively realized by redundant packets and a decoder is able to recover from any packet losses without any additional transport-related delays (e.g., as for retransmissions-based mechanisms). However, the cost to pay for using AL-FEC is additional bandwidth utilization. A dynamic AL-FEC controller (or media optimizer [26]) ensures usually in practice an optimal trade-off between AL-FEC bandwidth utilization and QoS is achieved by balancing the bandwidth split between the source media content and the AL-FEC redundancy added. Figure 6.17.2.1-1 outlines potential operation points for an AL-FEC encoded media stream (e.g., a video stream) as adapted by a dynamic AL-FEC controller based on network conditions. Operation point (1) illustrates an unprotected media stream at 30 Mbps, operation point (2) illustrates an AL-FEC protected media stream with a high AL-FEC redundancy rate of 100% (i.e., equal bandwidth split between media and repair packets), and operation point (3) illustrates an AL-FEC protected media stream with a low AL-FEC redundancy rate of 25%.

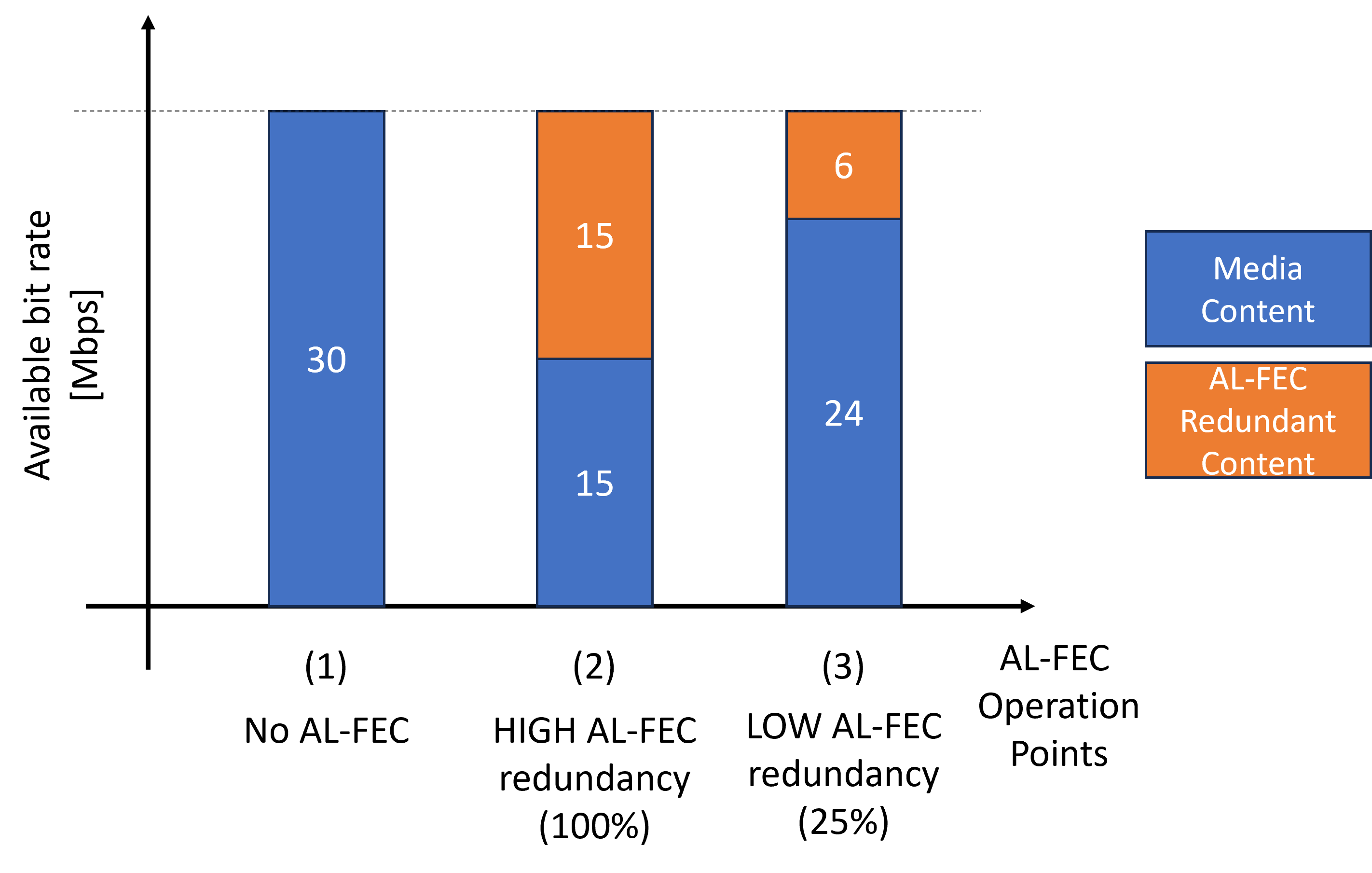


Figure 6.17.2.1-1: Example operation points of AL-FEC encoded media

The network metrics and statistics that influence the operation of AL-FEC controller are usually diverse and may involve at least [23], [24], [25], [26]:

- available bandwidth estimation/information

- packet loss statistics

- packet loss feedback

- RTT delay estimation/information.

Congestion events impact therefore the operation of the AL-FEC dynamic behavior. At a high-level in low packet loss conditions the AL-FEC redundancy rate is reduced considerably (e.g., operation point (3) in Figure 6.17.2.1-1), or even eliminated (e.g., operation point (1) in Figure 6.17.2.1-1) as per application configuration and preferences. On the other hand, for some higher packet loss values, given that the link has necessary bandwidth, the AL-FEC redundancy rate is increased to provide more redundancy and protection against network losses. However, in case congestion events persist and available bandwidth degrades the AL-FEC redundancy rate is usually backed-off and the media source rate is adapted to a lower source rate to account for the lower bandwidth available. The high-level control loop detailed above implies frequent network conditions monitoring (e.g., every second, [26]) and corresponding reactive AL-FEC redundancy rate and media source rate adaptation. This achieves an elastic and robust transport mechanism for low latency media delivery even over bursty lossy networks.

NOTE: Typical bursty losses are usually comparable to the network conditions monitoring times (e.g., couple of seconds resolution) and may affect multiple frames in a row.

#### 6.17.2.2 End-to-end transport perspective

Figure 6.17.2.2-1 illustrates the end-to-end perspective of AL-FEC encoded media streams over a 3GPP network. The Application Server (AS) is situated into a data network with potentially no QoS guarantees. It serves DL AL-FEC encoded XR media traffic to a UE connected to a 3GPP network. The DL traffic is transported over the data network to the UPF (ingested at reference point N6), then over the core network to the NG-RAN (ingested at reference point N3) and finally reaches the UE over the UE air-interface. Optionally, in case the UE is not the XR endpoint, it may relay over a tethered connection the media content to an XR tethered device.



Figure 6.17.2.2-1: End-to-end transport path for AL-FEC encoded PDU Sets.

The following points are worth remarking if AL-FEC awareness is enabled for NG-RAN to actively discard obsolete PDUs out of AL-FEC encoded PDU Sets:

- the AL-FEC obsolete PDU discarding at NG-RAN may be perceived by applications as congestion events, unless applications are fully aware of the NG-RAN behaviour (e.g., based on configuration, e.g., QoS configuration, or other feedback mechanisms);

- a fixed operation point at a static AL-FEC redundancy rate is not advisable from an information rate optimization perspective and media delivery standpoint since the AL-FEC redundancy will inefficiently utilize available bandwidth in detriment of the media source, as described above;

- a dynamic AL-FEC behavior is preferrable;

NOTE 1: Dynamic AL-FEC awareness signaling needs to be supported in supplement to PDU Set awareness, yet how to support it is FFS and may further involve SA2 and RAN2 coordination.

- the AL-FEC obsolete PDUs discarding at RAN will in effect reduce the 5GS operating bandwidth over the Uu interface to the media source rate;

- the AL-FEC encoding may protect against bursty packet losses over any of the data network, core network and air-interface link segments, yet the core network and air-interface link segments are part of the QoS flow architecture ensuring QoS guarantees (e.g., PSDB, PSER, or alternatively, PDB, PER);

- it seems that AL-FEC is mostly effective against the data network link segment, or alternatively, any link segment in the path to N6 that is out of the scope of 3GPP QoS flow architecture;

NOTE 2: How the AL-FEC encoding will impact the QoS configuration if NG-RAN discarding of obsolete AL-FEC PDUs is enabled is FFS.

- the AL-FEC obsolete PDU discarding at NG-RAN cannot protect against losses on tethered links, if present;

The analysis and remarks above implicitly assume NG-RAN feasibility (e.g., fast determination of PDUs ACK/NACK in various RLC modes to enable obsolete AL-FEC encoded PDU discard) and net benefits (e.g., capacity gains/bandwidth savings vs. added complexity) in discarding obsolete AL-FEC PDUs. However, the latter assumptions needs to be further verified in coordination with RAN2.

#### 6.17.2.3 Overhead of AL-FEC

To all the application receiver to reconstruct the source packets, the application sender sets the redundancy for AL-FEC based on the end-to-end packet loss rate in the network. It is important to understand the *overhead*, i.e., the ratio of the number of repair packets to the number of source packets needed to meet a probability that the application receiver can reconstruct all the source packets.

For a small number of source packets, the overhead can be significantly higher than the theoretical limit for the case of an infinite number of source packets (which is equal to *p/(1-p)*, where *p* is the end-to-end packet loss rate). This is because in a realization of the random packet losses, the packet loss rate may be higher than *p* and this effect is more prominent when the number of source packets is smaller. To illustrate, consider an example where *p*=0.1%, 1% and 10%, and the probability of reconstructing all the source packets, denoted , is set to 99.9%. The 10% represents the BLER without HARQ retransmission in typical implementations, and the 1% and 0.1% may represent the BLERs with HARQ retransmissions. Furthermore, it is assumed that the packet losses are independent, and each packet is sent in a separate transport block. The overhead (in percent) as a function of *K* is shown in Figure 6.17.2.3-1. For *K*=20, the overhead is 45%, and even as *K* increases to 100 the overhead still stays at 24%. In contrast, the theoretical limit is 11.1% (the red dashed line). The theoretical limit can be considered as the overhead needed to let the application receiver receive *K* packets on average.

A graph of a number of objects

Description automatically generated with medium confidence

(a)

A graph of a graph

Description automatically generated with medium confidence

(b)

Figure 6.17.2.3-1: The overhead of AL-FEC as a function of the number of source packets *K* for 99.9% probability of reconstructing all the source packets at the application receiver: (a) linear scale; (b) logarithmic scale.

The steps for calculating the overhead are as follows:

1) Find the minimum value (denoted ) of that satisfies , where 0.999

2) The overhead is

Note that in the RAN, a transport block may carry multiple packets. Thus, even if *K* is large, the effective value for the purpose of reconstructing the source packets may be small. For example, if *K*=100 and each transport block carries 4 packets, then effectively we are dealing with 25 transmissions and the overhead would be corresponding to the overhead for *K*=25 rather than the overhead for *K*=100 in Figure 6.17.2.3-1.

**Observation 1:** the overhead of AL-FEC may be much higher than the overhead needed to let the application receiver receive *K* packets on average, where *K* is the number of source packets.

If the RAN transmits every packet (or PDU) of a PDU Set with AL-FEC encoding, then *the overhead at the RAN* (i.e., on average how many packets beyond the number of source packets the RAN needs to transmit normalized by the number of source packets) can be high.

In contrast, if the RAN can drop obsolete packets (or PDUs), the overhead at the RAN can be reduced. With ideal assumptions, e.g., the base station knows immediately and reliably which PDUs are delivered successfully, the overhead at the RAN can drop to the theoretical limit. With practical assumptions, the overhead will be higher than but still can still be close to the theoretical limit.

**Observation 2:** if the RAN can drop obsolete PDUs of a PDU Set with AL-FEC encoding, the overhead at the RAN can be dropped to be close to the theoretical limit.

## 6.18 Solution #18: Real-time communication congestion control algorithms and AL-FEC

### 6.18.1 Key Issue mapping

This solution addresses key issue #4 on AL-FEC awareness for PDU Set handling and Key Issue #3: Enhancements for application-layer FEC support.

### 6.18.2 Description

#### 6.18.2.1 General

This clause introduces the congestion control algorithms for real-time communication, including those currently implemented in WebRTC [28] and two algorithms described in two IETF RFCs [29] [30].

Observations are made on these algorithms and AL-FEC.

#### 6.18.2.2 Google Congestion Control (GCC)

Google Congestion Control (GCC) is one of the two congestion control algorithms supported in the WebRTC implementation. The algorithm determines a sending bit rate, which limits the total bit rate for RTP packets, RTCP packets and RTP retransmissions (if any) and drives the bit rate adaptation of the media codecs. It uses packet losses and delays as signals of network congestion in adjusting the sending rate. The packet loss feedback is based on RTCP reports. The delays are one-way delays measured by the "absolute send time" RTP header extension as described in TS 26.522 [2], and the change in the one-way delay is computed to detect network congestion.

GCC has not been standardized. Although the informational IETF document [27] describes GCC, the description is outdated. In what follows, we present GCC based on the current WebRTC implementation [28].

GCC sets the sending rate to the lesser of a loss-based bandwidth estimation and a delay-based bandwidth estimation. That is, the sending rate at time is set to:

where is the loss-based bandwidth estimation made at time when the lastest RTCP loss report (the *k*th) is received, and is delay-based bandwidth estimation made at time when the lastest one-way delay measurement is taken or reported.

**Delay-based bandwidth estimation:**

This is described in aimd\_rate\_control.cc, based on the detection of overuse. The one-way delay

where is a constant equal to 1200 bytes/second/RTT, is a backoff factor. The state ‘decrease’, ‘hold’ and ‘increase’ are based on the slope of the change in the one-way delay as a function of time. The change in the one-way delay from packet group to packet group is defined as

where is the arrival time of packet group , and is the departuture time of packet group , is the arrival time of packet group , and is the departuture time of packet group . The slope (termed trendline in the WebRTC source code) is estimated and compared with thresholds to determine the state.

**Loss-based bandwidth estimation:**

There are three versions of loss-based bandwidth estimation algorithms.

**Version 0 (static thresholds):**

The loss-based bandwidth estimation is calculated as:

where is the time when the previous RTCP loss report is received, is the loss (fration of loss) received in the most recent RTCP loss report, and is the loss-based bandwidth estimation at the time when the previous RTCP loss report is received.

**Version 1 (dynamic thresholds):**

This version is similar to version 0, except that it uses (1) dynamic thresholds are used, and (2) the factor for increasing the estimation is adaptive to RTT, instead of being a constant 1.08 in Version 1. The loss-based bandwidth estimation is:

where and are dynamic thresholds, ,

is the probability for decreasing the bandwidth estimation, equal to the minimum of the average loss probability and the most recent loss probability,

is the probability for increasing the bandwidth estimation, equal to the average loss probability if and equal to ,

where is time elaspsed, is a time window,

is the bit rate of the acknowledged transmissions,

where is the RTT, is a pre-configured minimum RTT and is a pre-configured maximum RTT. With this dynamic threshold , the increase in loss-based bandwidth estimation will be slower when the RTT gets higher.

**Version 2 (maximum likelihood):**

To calculate the loss-limited bandwidth, in a nutshell, the sender chooses a tuple of an inherent loss probability (loss probability induced by the channel error rather than network congestion) and a loss limited bandwidth that maximizes the following objective function which is based on the logarithmic maximum likelihood of observing the number of packet losses:

where:

is the number of observations,

=0.9 is a weight,

is the number of bytes lost,

is the number of bytes received,

,

where is the sending rate at the time of the *i*th observation,

is a high bandwidth bais (that depends on ),

is the index of the most recent observation and is the index of the oldest observation.

#### 6.18.2.3 PCC

Performance-oriented Congestion Control (PCC) is the other congestion control algorithm supported in the current WebRTC implementation [28]. The sender adjusts the sending rate and observe the performance metrics including the delay and packet loss, and pick the sending rate that maximizes a utility function:

where is the sending rate, is the gradient of RTT , is the loss rate, and are coefficients.

#### 6.18.2.4 NADA

Network-Assisted Dynamic Adaptation (NADA) is specified in RFC 8698 [29]. This algorithm considers delay, packet loss, and ECN marking as signals of network congestion. Furthermore, it converts packet loss and ECN marking to some equivalent penalty in terms of delay and forms an aggregate congestion signal. Specifically, the receiver calculates:

where is the equivalent delay after non-linear warping,

is the estimated packet ECN marking ratio, is the reference packet ECN marking ratio,

is the reference delay penalty for ECN marking when packet marking is at ,

is the estimated packet loss ratio, is the reference packet loss ratio,

is the reference delay penalty for packet loss when packet loss is at .

The receiver decides whether the sender is to be in the accelerated ramp-up mode (rate update mode ) or in the gradual update mode (). The sender needs to be in the accelerated ramp-up mode if there are no recent packet losses in an observation window (of 500 ms) and there is no build-up of queueing delay. Otherwise, the sender needs to be in the gradual update mode.

The receiver also calculates the receiving bitrate

The receiver sends , , and to the sender.

The sender performs multiplicative increase in the accelerated ramp-up mode using the RTT and , and performs additive decrease in the gradual update mode using .

#### 6.18.2.5 SCReAMv2

Self-Clocked Rate Adaptation for Multimedia 2 (SCReAMv2) [30] uses packet losses, delay, and ECN marking as signals of network congestion.

A congestion window size limits the sending rate and is adjusted based on:

**- Packet losses:** A packet loss causes the congestion window size to decrease, but not as much as a packet loss does in the case of TCP Reno

**- Queueing delay:** The congestion window size decreases linearly when the average queueing delay exceeds a threshold.

**- ECN marking:** When a classic ECN marking is received, the congestion window size decreases by a factor. When L4S ECN marking is received, the congestion window size decreases in proportion to the fraction of packets that are L4S ECN marked.

#### 6.18.2.6 Summary of congestion control algorithms

The interactions between the application and the network are through signaling manifested by packet losses, queueing delay and ECN marking, as summarized in the table below:

Table 6.18.2.6-1: Comparison of congestion control algorithms

|  |  |  |  |
| --- | --- | --- | --- |
| Congestion Control Algorithm | React to packet losses? | React to ECN marking? | React to queueing delay? |
| GCC (with loss based bandwidth estimation v0) | No, if the packet loss rate is between 2% and 10%;  yes otherwise. | No | Yes |
| GCC (with loss based bandwidth estimation v1) | Yes | No | Yes |
| GCC (with loss based bandwidth estimation v2) | Yes | No | Yes |
| PCC | Yes | No | Yes |
| NADA | Yes | Yes | Yes |
| SCReAMv2 | Yes | Yes | Yes |

**Observation 1:** All congestion control algorithms for real-time communication in Table 5.43.5-1 use queuing delay (among other metrics) as a signal of network congestion.

**Observation 2:** Although all congestion control algorithms for real-time communication in Table 5.4.3.5-1 use packet losses as a signal of network congestion, one algorithm is not sensitive to packet losses when the packet loss rate is within the range [2%, 10%].

**Observation 3**: Two of the six congestion control algorithms for real-time communication in Table 5.4.3.5-1 support ECN marking.

Although WebRTC currently implements only GCC (with three versions) and PCC, it does not prevent one from adding other congestion control algorithms such as NADA and SCReAMv2. Since these algorithms support ECN marking, we have the following observation.

**Observation 4:** It is possible to add ECN marking support to WebRTC for congestion control. Updates of RFCs is required to ensure interoperability.

#### 6.18.2.7 Packet loss rate calculation for AL-FEC

When AL-FEC is used, the source packets and the repair packets may be sent in a single RTP stream (identified by an SSRC). This is the case for ULPFEC in the WebRTC implementation [28].

Alternatively, the source packets and the repair packets may be sent in different RTP streams (identified by different SSRC’s) within the same RTP session. This is the case for FlexFEC in the WebRTC implementation [28]. The packet loss rate is calculated individually. The packet loss rates are then combined to form a single packet loss rate as an input to the congestion control algorithms.

**Observation 5:** For AL-FEC, the packet loss rate on the source packets and that on the repair packets can be calculated separately. RFC 5109 [13] (Clause 12) give congestion considerations. However, there are no considerations to handle repair losses differently than source packet losses.

Although RFC 5109 [13] recommended that the source packets and the repair packets may be sent in different sessions (identified by different IP 5-tuples), we are unaware of any commercial implementation of such scheme.

Note that RFC 8085 [34] (UDP usage Guidelines) recommend that an "application SHOULD perform congestion control over all UDP traffic it sends to a destination, independently from how it generates this traffic".

**Recommendation:** SA4 to study how to handle the packet losses for congestion control when the application traffic is encoded with AL-FEC.

## 6.19 Solution #19: Congestion control enhancement to support AL-FEC awareness handling

### 6.19.1 Key Issue mapping

This maps to Key Issue #4.

### 6.19.2 Description

#### 6.19.2.1 Background of using AL-FEC for real-time communication in cellular networks

There are inherent losses in the over-the-air transmission in cellular networks. To recover from the losses, retransmission in PDCP, RLC and MAC may be used. However, the low-latency requirements for XR applications put constraints on the use of PDCP and RLC layer retransmissions.

If retransmission is needed, MAC layer HARQ retransmission is preferred. However, RAN implementations typically have an instantaneous BLER (iBLER) of 10% for high spectral efficiency. That requires a large number of HARQ retransmissions, resulting in large delays. If AL-FEC is used, the need for HARQ retransmission is greatly reduced. This is illustrated in the simulation study below.

**Scenario:** TDD with subframe format DDDSU, 30 kHz SCS, HARQ turnaround time about 5 ms, 100 MHz bandwidth, 60 fps, video frame size following a truncated Gaussian (STD, Max, Min) distribution: (10.5%, 150%, 50%) of average frame size, the average frame size 0.5 Mbits, the average SNR 5 dB, iBLER 10% and the subsequent BLERs for HARQ retransmissions following a BLER correlation model on the successes/failures of the HARQ transmissions based on field data (which gives the probability that the current TB is successfully transmitted in the *n*th attempt conditioned on that the previous TB is successfully transmitted in the *m*th attempt, where *n*=1, …, 5, *m*=1,…5), RLC acknowledged mode (AM) (t-reassembly 25 ms, t-StatusProhibit 10 ms), and MDS AL-FEC code each time applied to the PDUs of a single video frame. RLC AM is used to handle the 0.22% residual BLER resulting from the BLER correlation model given that we evaluate 99.9 percentile latency (otherwise with RLC UM, 0.22% of the PDUs will never be delivered successfully and the 99.9 percentile latency for the "No AL-FEC case" will be infinity).

Table 6.19.2.1-1: Delay without and with AL-FEC

|  |  |  |  |
| --- | --- | --- | --- |
| Scheme | Redundancy ratio | Latency (ms) | |
| 99.9 percentile | 99 percentile |
| No AL-FEC | 0 | 55 | 43.5 |
| With AL-FEC | 30% | 15 | 14 |

We see from the table that AL-FEC reduces the 99-percentile delay from 43.5 ms to 14 ms and reduces the 99.9-percentile delay from 55 ms to 15 ms.

**Observation 1:** AL-FEC can reduce the delay for practical RAN implementations.

#### 6.19.2.2 Potential Benefits of Application-layer FEC awareness for PDU Set handling

When the RTP source adds redundant PDUs for an ADU, the redundancy is over budgeted to account for error in the estimation of the packet loss rate in the network. That is, there are more packets than needed for reconstructing the ADU. At the last hop of the PDU Set delivery, if the base station is aware of AL-FEC, it can drop PDUs that are no longer needed for reconstructing the ADU. This has two benefits:

- Reducing the usage of resources and hence improving the spectral efficiency (the amount of resources per PDU Set)

- Lowering the power consumption of the UE because the network can let the UE go to the sleep mode earlier.

This AL-FEC aware PDU Set handling is illustrated in the Figure 6.19.2.2-1. Packets 0 and 1 are served in the first time slot (which is a ‘D’ slot), and packets 2 and 3 in the second slot, and so on. In Case-1, without AL-FEC awareness, the redundant packets are still transmitted, which wastes network resources and keeps the UE awake longer before the network lets the UE go to the sleep mode. In contrast, in Case-2, with AL-FEC awareness, the use of network resources becomes more efficient, and the UE goes to the sleep mode earlier.

For ease of exposition, we assume that the number of source packets is 20 in Figure 6.19.2.2-1.

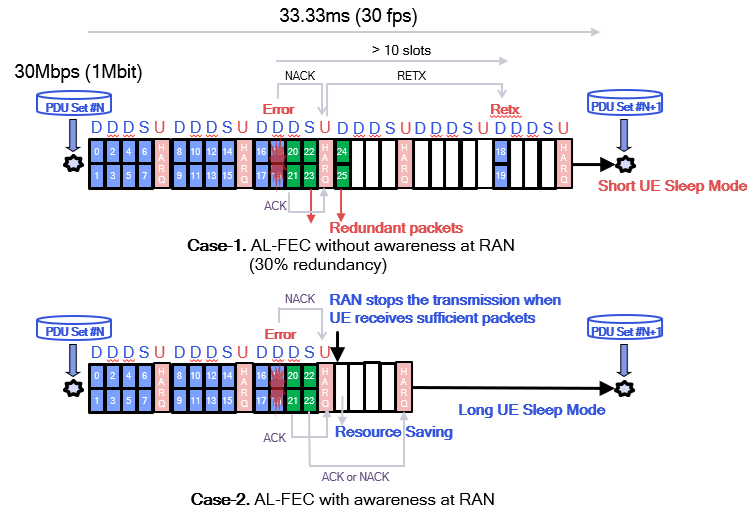


Figure 6.19.2.2-1: Potential Benefits of AL-FEC awareness at RAN

A Simulation study was carried out for the following scenario to quantify the benefits:

The video frame size being fixed at 1 Mbit, TDD with slot pattern DDDSU, 30 kHz SCS, HARQ retransmission (initial transmission and up to 2 retransmissions), 100 MHz bandwidth, iBLER = 10%, the BLER after the 2nd HARQ transmission being 5%, the BLER after the 3rd HARQ transmission being 1%, the minimum value for the PDSCH to HARQ feedback timing indicator (i.e., K1) being 2 time slots (allowing the NACK for packets 18 and 19 and the ACK for packets 20 and 21 to be multiplexed in the same U slot in Figure 6.19.2.2-1), the power saving feature being Rel-18 eCDRX + PDCCH skipping, 30 fps, MDS AL-FEC code each time applied to the PDUs of a single video frame, redundancy ratio 30% (FEC code rate = 22%). The power consumption values are presented by relative power as defined in TS 38.840, and it accounts for the power consumed by the modem for both uplink and downlink.

Table 6.19.2.2-2: Potential Benefits of AL-FEC aware PDU Set handling at RAN

|  |  |  |  |
| --- | --- | --- | --- |
|  | 99-Percentile Latency  (ms) | Power Consumption | Network loading |
| Case 1 – without AL-FEC awareness at RAN | 27 | 312 | 91.08% |
| Case 2 – with AL-FEC awareness at RAN | 27  (0%) | 270  (-13%) | 74.03%  (-19%) |

We see from the table that AL-FEC aware handling reduces power consumption of the UE by 13%, which is significant for the UE. It also reduces the network loading by 19%, and this allows the network to accommodate more users.

**Observation 2:** AL-FEC aware PDU Set handling can potentially reduce the UE power consumption and network loading.

To perform intentional discard, the NG-RAN needs to retain state variables (e.g., how many PDUs have been delivered successfully for each PDU Set). The associated complexity may be acceptable given that a base station typically serves a much smaller number of UEs than the UEs served by a UPF.

#### 6.19.2.3 Implications of Application-layer FEC awareness for PDU Set handling on congestion control

Many congestion control algorithms, such as Google congestion control algorithms [28], NADA [29] and SCReAMv2 [30], use packet losses as a signal of network congestion. Therefore, it is important for the congestion control algorithms to correctly interpret packet losses in the case of AL-FEC awareness handling of the PDU Set.

#### 6.19.2.4 The Proposed Solution

##### 6.19.2.4.1 Case 1: In congestion

When there is congestion, as long as the AL-FEC awareness handling of the PDU Set does not alter the packet loss statistics, there is no impact on congestion control. Examples are given below.

**Example 1:** the network can discard repair packets rather than source packets in the case of FlexFEC without changing the overall packet loss rate, which will not lead to over reduction of the sending rate.

**Example 2:** the network can discard redundant packets across different PDU Sets while still meeting the required redundancy ratios for reconstructing the respective ADUs.

##### 6.19.2.4.2 Case 2: Not in congestion

When there is no congestion, the network can discard obsolete packets (by obsolete, it means that the packets are no longer needed for reconstructing the ADU at the receiver), which will increase the packet loss rate observed by the RTP sender, and to avoid the RTP sender mis-interprets the packet losses as signals of congestion, the network can indicate to the RTP sender that there is no network congestion and such packet losses are not expected to be taken into account by the congestion control algorithm for determining the sending rate.

NOTE 1: It is FFS how the network provides feedback to the application on obsolete packets dropped in the network.

NOTE 2: Although congestion control is currently not in the scope of TS 26.522 [2], it could be studied in this report.

**Conclusion:** The behavior of the application at the sender may be impacted because congestion control that may be triggered by RAN discard needs to be handled differently than in current implementations (i.e., not see RAN intentional discard as an indication of congestion).

## 6.20 Solution #20: Guidelines for PDU Set Marking of Unmarked/Lone PDUs

### 6.20.1 Key Issue mapping

This is a solution to key issue #2.

### 6.20.2 Description

#### 6.20.2.1 General

The RTP Header Extension for PDU Set marking aims to support enabling PDU Set QoS for RTP media traffic in the 5G System as defined in [3].

It helps components in the 5G System to identify PDU Sets and apply PDU Set QoS, i.e. see 5.37.5 in [3].

This clause proposes guidelines for applying and using the RTP Header Extension for PDU Set Marking when lone/unmarked PDUs are present to enable effective PDU Set based QoS handling.

NOTE: This solution covers the case where 5G System cannot apply packet filter based on RTP SSRC and/or RTP PT in a QoS flow (Release 18). The case of using this packet filter is covered in clause 6.2.

For PDU Set based QoS handling, the PDU Set QoS parameters are introduced in TS 23.501 [3] as follows:

- PDU Set Delay Budget, which defines an upper bound for the delay that a PDU Set may experience for the transfer between the UE and the N6 termination point at the UPF.

- PDU Set Error Rate, which defines an upper bound for the rate of PDU Sets that have been processed by the sender of a link layer protocol (e.g., RLC in RAN of a 3GPP access) but that are not successfully delivered by the corresponding receiver to the upper layer (e.g., PDCP in RAN of a 3GPP access).

- PDU Set Integrated Information, which indicates whether all PDUs of the PDU Set are needed for the usage of the PDU Set by the application layer in the receiver side.

Provisioning of protocol description to assist UPF for the PDU Set information identification is defined in TS 29.571 [55].

PDU Set information in the GTP-U header added by the UPF to the NG-RAN is defined in TS 38.415 [56].

If the NG-RAN receives PDU Set QoS Parameters, it enables the PDU Set based QoS handling and applies PDU Set QoS Parameters. When the PDU Set QoS parameters are available, they will supersede the PDU QoS parameters (i.e. PSDB/PSER supersedes the PDB/PER).

#### 6.20.2.2 Guideline for PDU Set marking of lone/unmarked PDUs

An additional guideline is provided to support this case for the following scenario:

- The case of unmarked Packets in a stream to which RTP Header Extension for PDU Set marking is applied (including the marking of data bursts that are not part of a PDU Set).

It is recommended that when the RTP HE for PDU set marking is enabled, the RTP HE is applied to each RTP packet that belongs to a PDU Set. This enables effective identification of all packets belonging to a PDU Set by the 5G System. This can subsequently enable suitable PDU Set QoS based Handling for each PDU Set in the NG-RAN.

Please see clause 6.2 for a detailed gap analysis of when and how lone/unmarked PDU can occur. In the 5G System, such an unmarked PDU will be treated as a separate PDU Set and PDU Set QoS parameters are applied instead of PDU QoS Parameters to a single packet.

NOTE 1: Due to this behaviour of the 5G System, PDU Set QoS parameters such as delay budget, when configured, have to be suitable for the unmarked PDUs as well as the marked PDUs.

A practical example could be an audio and a video stream multiplexed in a single RTP session which is carried on a QoS flow. In this case the video stream RTP packets include RTP Header Extension for PDU Set marking for each RTP Packet but on the other hand the audio stream RTP Packets do not contain the RTP Header Extension. The PDU Set delay budget is set to 20 milliseconds (as example) in order to support the real-time video frame transmission. This delay budget is used in the 5G System for video PDU sets but also for the independent audio packets that are treated as a single PDU Set in the 5G System. In this practical case, a delay budget of 20 milliseconds is also suitable for the audio stream as this delay leads to a satisfactory end-to-end delay performance.

The unmarked packets do not count towards the PDU set size of marked PDU Sets as optionally signalled in the RTP Header extension. In the case of an unmarked packet, it is handled as a separate PDU Set instead, and the size of the packet is identical to the PDU Set Size of this separate PDU Set (which expected tobe equal to the size of the PDU).

End of data burst signaling may still apply in the RTP Header Extension to all packets (both marked and unmarked).

In case the last packet of the data burst is an unmarked packet it may be possible to add RTP HE for PDU Set marking to the packet if possible (i.e. is an RTP Packet). Otherwise, the end of data burst marking may not be valid.

The guideline for determining PDU Set information at the UPF from either RTP HE or unmarked PDU is given in Table 6.20.2.2-1.

Table 6.20.2.2-1: Determining PDU Set information ([55]) at UPF from RTP HE and unmarked PDU

|  |  |  |
| --- | --- | --- |
| PDU Set information | RTP HE | Lone/unmarked PDU (and UPF cannot derive PDU Set based on content type as in Annex A of 26.522 [2]) |
| PDU Set importance | Set by interpreting PSI field in [2] | Set by 5G System to a configured value or set as proposed in solution #15 |
| PDU Set Size | Optionally transmitted in additional PSSize field and derived from this field. | PDU Size |
| End of Data Burst | Can be set by EoDB flag | N/A for lone PDU |
| PDU Sequence number | From PDU sequence number in RTP HE | Set to 0 |
| PDU Set Sequence number | PSSN field from RTP HE with most significant bit is set to 0 | Set by UPF with most significant bit set to 1 |
| end of PDU set | End of the PDU Set in RTP HE | Always 1 |

The PDU Set information can be determined by the UPF as shown in table 6.20.2.2-1 with different PDU Set information in the left most column. The middle column indicates how the UPF can derive PDU Set information for packets that include RTP HE for PDU Set information. The right most column indicates how UPF can derive PDU Set information for unmarked packets (lone PDUs).

The PDU Sequence number could be retrieved from the PSN in RTP HE, or when no RTP HE is present it can be determined by the UPF.

The PDU Set Sequence Number has some additional steps applied by the UPF to enable to generate a PDU Set Sequence Number that can identify the PDU Set as defined in TS 29.571 [55] and TS 38.415 [56] (these steps are an example, UPF could apply different steps, this is just to give how this scenario can be addressed). Alternatively, the PDU Set Sequence Number can also be allocated by UPF following its own implementation to obtain the PDU Set Sequence Number.

As an example, the UPF can only use the 9 least significant bits of the RTP HE, mapping PSSN from RTP HE to PSSN in PDU Set Information and set the most significant bit of PSSN in PDU Set information to 0.

NOTE 2: The RTP HE PSSN cannot map directly to PSSN for PDU Set information when lone PDU's are present

The UPF can then use the 9 least significant bits of PSSN in PDU Set information based on a counter for unmarked PDUs and set the most significant bit to 1. This is an example of how the UPF can deal with both PSSN from RTP HE and unmarked PDU and still get PSSN values that can be used to identify the PDU Set (following definitions from TS 29.571 [55] and TS 38.415 [56] for PDU Set Sequence Number).

NOTE 3: This is solution is to show a possible mapping of PSSN from RTP HE and non RTP HE packets can be done at UPF to enable implementability at UPF. Other solutions can be equally valid and applicable by the UPF.

Signaling presence of unmarked PDUs could be developed in TS 26.510 [50], as currently TS 26.510 [50] explicitly states that signaling requires PDUs to be marked, but for the lone PDU case it is optional, therefore some updates to TS 26.510 [50] can be considered. There is also the case in Annex A of TS 26.522 [2] where the UPF determines the PDU Sets based on codec characteristics for H.264 and H.265 for unmarked packet. This option could be seen as alternative and potentially signaling in TS 26.510 [50] can be developed for both cases.

The definition of PSSN in TS 26.522 [2] and TS 38.415 [56] are currently not aligned, the strict increment by one only applies for TS 26.522 [2] not for PSSN generated by UPF as part of PDU Set information.

NOTE 4: This guideline does not require changes to the definition of PSSN in [2].

#### 6.20.2.3 Discussion

**Pros:**

SA2 has asked SA4 about the possibility of lone PDUs being present and if PDU Set QoS will have an effect on the user experience. The benefits of this solution are:

- Explicit guideline for how the UPF could potentially deal with the case of unmarked PDUs, even though the UPF behaviour is up to implementation. Such a recommendation can still be useful to make sure that RTP HE can be used in practice by the UPF.

- A reference to TS 23.501 can potentially be added to clarify that unmarked and marked PDUs will both be using PDU set QoS.

**Cons:**

This solution proposes informative text to a normative technical specification making adoption of the specification easier and more straightforward. As long as it is clear that it is an example guideline, it is expected to be not harmful as it does not contradict existing specifications. In the case a different PDU Set QoS Parameter is required for unmarked PDUs, this is not covered in this solution. This may require additional coordination with SA2 to enable such handling in 5GS. Other solutions to other key issues such as KI #9 or KI #14 could potentially bring forward solutions that could address this case.

## 6.21 Solution #21: Periodicity and TTNB with the lone PDU

### 6.21.1 Key Issue mapping

This solution intends to give gap analysis on the KI#2: QoS handling requirements for lone PDU.

### 6.21.2 Description

For the downlink direction, the PSA UPF identifies PDUs that belong to PDU Sets and marks them accordingly as described in clause 5.37.5.2 of TS 23.501 [3]. If the PSA UPF receives a PDU that does not belong to a PDU Set based on Protocol Description for PDU Set identification, then the PSA UPF still maps it to a PDU Set and determines the PDU Set Information as described in clause 5.37.5.2 of the TS 23.501 [3]. In this case, both the single PDU and the PDUs belonging to a PDU Set are mapped into the same QoS Flow and the single PDU is delivered to the UE in the DL direction following the PDU Set QoS parameters of the QoS Flow.

When PDU Set marking is activated, there is expected to be no lone PDUs with other PDUs belonging to a PDU Set in the same service data flow. There are other scenarios where lone PDUs and PDUs belonging to a PDU Set from different flows are mapped into the same QoS Flow as follows.

- **Scenario #A**: the RTP and RTCP flows are carried over a single RTP service data flow. When the PDU Set Marking is enabled in the UPF for the RTP flow(s), the UPF takes the RTCP traffic as lone PDUs (e.g. there is no RTP HE for the RTCP packets).

- **Scenario #B**: the RTP packets and retransmitted RTP packets are carried over a single RTP service data flow are carried over the same QoS Flow. When the PDU Set Marking is enabled in the UPF for the RTP flow, the UPF takes the retransmitted RTP traffic as lone PDUs.

- **Scenario #C:** Different RTP media streams multiplexed in a single RTP service data flow. In this scenario, multiple RTP media streams with different payload types are multiplexed in a single RTP service data flow. For example, if the audio and real-time subtitle streams are multiplexed in a single RTP service data flow and the video stream is in a different RTP service data flow, the UPF can use the Payload Type and NAL information to identify the audio PDU and does the PDU Set marking for the audio media and treats the realtime subtitle as the lone PDU.

As can be seen from the above, one key reason for the lone PDU handling is that the PDUs belonging to a PDU Set and the lone PDUs are carried over the same QoS Flow and the 5GS performs the PDU Set marking on all PDUs of the QoS Flow.

**Observation 1: the co-existence of lone PDUs and PDUs belonging to a PDU Set in the same QoS Flow is due to PDU Set marking performed per QoS Flow.**

The UL and/or DL Periodicity information may be provided by the application to the CN and then to NG-RAN to configure UE power saving management scheme for connected mode DRX and radio resource scheduling, normally the Periodicity is for the RTP stream of audio and video (it is unclear whether the Periodicity is provided to RTP stream of the realtime subtitle). If the single/lone PDUs are multiplexed into the PDU belonging to a PDU Set with Periodicity, the DL PDUs of the QoS Flow are received randomly in time by the NG-RAN and it is almost impossible for the NG-RAN to do Periodicity-based radio resource scheduling and UE power saving management.

**Observation 2: the co-existence of lone PDUs and PDUs belonging to a PDU Set in the same QoS Flow makes the Periodicity information provided by the AS useless to the NG-RAN.**

The TTNB is proposed in R19 to improve the UE power saving management scheme for connected mode DRX and radio resource scheduling by the NG-RAN, the TTNB can be provided in the RTP HE by the application. However, similarly as above, If the single/lone PDUs are multiplexed into the PDU belonging to a PDU Set with Periodicity, the DL PDUs of the QoS Flow to be received randomly in time by the NG-RAN and it is almost impossible for the NG-RAN to do Periodicity-based radio resource scheduling and UE power saving management.

**Observation 3: the co-existence of lone PDUs and PDUs belonging to a PDU Set in the same QoS Flow makes the TTNB provided by the AS useless to the NG-RAN.**

To solve the problems described above, the following information can be provided to the AS.

1. **The AS can provide explicit different QoS requirements for different streams to the 5G network. And the 5G network maps different streams into different QoS Flows with different QoS. For the RTCP, the 5G network can provide the same QoS parameters with the associated RTP media stream but with different QoS Flow. NG-RAN will normally bind the different QoS Flows with the same QoS parameters into the same DRB, in such case, the RTCP and RTP share the same radio channel and the SR and RR of the RTCP can have the same results as the RTCP and RTP in the single QoS Flow.**
2. **If different media streams are multiplexed into the same RTP service data flow by the AS, the AS can provide packet filter extension as agreed in R19 TS 23.501 to help the 5G network to demultiplex the different media streams from the same RTP service data flow, the AS also provides explicit different QoS requirements for different media streams. In such a way, the R19 5G network can map these different media streams to different QoS Flows.**
3. **The AS provides the Periodicity and/or TTNB information per demultiplexed media stream (e.g. audio, video) instead of per RTP service data flow.**

Via the potential R19 enhancements in 5GS, it is possible to avoid the co-existence of lone PDUs and PDUs belonging to a PDU Set in the same QoS Flow.

### 6.21.3 Conclusion

Based on the gap analysis in the above, it is proposed to make the following conclusions.

**1) The AS can provide explicit different QoS requirements for different streams to the 5G network. And the 5G network maps different streams into different QoS Flows with different QoS. For the RTCP, the 5G network can provide the same QoS parameters with the associated RTP media stream but with different QoS Flow. NG-RAN will normally bind the different QoS Flows with the same QoS parameters into the same DRB, in such case, the RTCP and RTP share the same radio channel and the SR and RR of the RTCP can have the same results as the RTCP and RTP in the single QoS Flow.**

**2) If different media streams are multiplexed into the same RTP service data flow by the AS, the AS can provide packet filter extension as agreed in R19 TS 23.501 to help the 5G network to demultiplex the different media streams from the same RTP service data flow, the AS also provides explicit different QoS requirements for different media streams. In such a way, the R19 5G network can map these different media streams to different QoS Flows.**

**3) The AS provides the Periodicity and/or TTNB information per demultiplexed media stream (e.g. audio, video) instead of per RTP service data flow.**

## 6.22 Solution #22: Guidelines for PDU Set Marking in Multiplexing Scenarios

### 6.22.1 Key Issue mapping

This is a solution to key issue #9.

### 6.22.2 Description

#### 6.22.2.1 General

The RTP Header Extension for PDU Set marking aims to support enabling PDU Set QoS for RTP media traffic in the 5G System as defined in [3].

It enables components in the 5G System to identify PDU Sets and apply PDU Set QoS, i.e. see 5.37.5 in [3].

This clause proposes guidelines for applying and using the RTP Header Extension for PDU Set Marking in additional multiplexing scenarios to enable effective PDU Set based QoS handling.

NOTE: This solution covers the case where either the 5G System cannot apply packet filter based on RTP SSRC in a QoS flow (Release 18), or it is not desirable to use different QoS Flows for the multiplexed content. In this solution multiplexed streams are handled in a single QoS flow.

For PDU Set based QoS handling, the PDU Set QoS parameters are introduced in TS 23.501 [3] as follows:

- PDU Set Delay Budget, which defines an upper bound for the delay that a PDU Set may experience for the transfer between the UE and the N6 termination point at the UPF.

- PDU Set Error Rate, which defines an upper bound for the rate of PDU Sets that have been processed by the sender of a link layer protocol (e.g., RLC in RAN of a 3GPP access) but that are not successfully delivered by the corresponding receiver to the upper layer (e.g., PDCP in RAN of a 3GPP access).

- PDU Set Integrated Information, which indicates whether all PDUs of the PDU Set are needed for the usage of the PDU Set by the application layer in the receiver side.

If the NG-RAN receives PDU Set QoS Parameters, it enables the PDU Set based QoS handling and applies PDU Set QoS Parameters. When the PDU Set QoS parameters are available, they will supersede the PDU QoS parameters (i.e. PSDB/PSER supersedes the PDB/PER).

In multiplexing scenarios, multiple types of media and/or control packets are carried on a single QoS flow.

#### 6.22.2.2 Unmarked packet handling

It is recommended that when the RTP HE for PDU set marking is enabled, the RTP HE is applied to each RTP packet that belongs to a PDU Set. This enables effective identification of all packets belonging to a PDU Set by the 5G System. This can subsequently enable suitable PDU Set QoS based Handling for each PDU Set in the NG-RAN.

In some cases, packets may exist that do not belong a PDU Set but are instead a single independent packet intended for transmission. Examples of these cases are given in solutions to KI #2 such as in clause 6.2 (e.g. RTCP packet, unmarked audio packet).

The guidelines for handling unmarked packets are discussed in solutions to KI#2.

#### 6.22.2.3 RTP HE for multiplexed content

An RTP sender could also include additional RTP HE for the additional multiplexed streams. This may be useful in the case frames consist of multiple packets that can be grouped in PDU Sets or if setting the PDU Set importance is desired (sc1, sc2). In addition, cases are considered when the content is natively multiplexed and it is hard to distinguish packets based on media type by a sender or receiver, as they may contain multiple media types in an RTP Packet (sc3, sc4). Also, the case where multiple RTP streams of the same media type is multiplexed is considered (sc5, sc6).

To illustrate this, Table 6.22.2.3-1 provides some examples on different multiplexing scenarios and the corresponding guidelines for setting RTP HE are further given in Table 6.22.2.3-2.

Table 6.22.2.3-1: Example of Multiplexing scenarios

|  |  |  |  |
| --- | --- | --- | --- |
| Scenario | Multiplex Type | Description | Implications for RTP Header Extension for PDU Set Marking for sender |
| sc1 | audio + video RTP multiplex [4] | Native Audio and Video streams are carried in separate RTP streams with different SSRC, and different PT Packets contain either audio or video. | Typically, RTP HE is used for the video stream, audio packets can be unmarked or in some cases they can also use the RTP HE (if frames comprise multiple packets). If both audio and video RTP packets are marked into PDU Sets, the RTP HE for PDU Set is applied to video and audio RTP streams separately. RTP video packets and audio packets are marked as separate PDU Sets, not as part of the same PDU Set. |
| sc2 | audio + video [4], RTCP [5] | Same as above, but in this case also RTCP packets exist. Packets contain audio, video or RTCP. | Same as above for audio and video. RTP HE cannot be used for RTCP packets, and these will be handled as unmarked/lone PDUs.  End of Data Burst signal cannot be used in case RTCP packet is the last one in a data burst. |
| sc3 | audio, video native multiplex [36] | Stream packets can contain both audio and video. In this case an RTP packet can contain both audio and video content. In addition, packets can also contain other metadata related to the streams. | In this case, single PDU Sets will contain different media types; additional guidance is provided to handle this case |
| sc4 | audio, video native multiplex [36] + RTCP [4] | same as above adding RTCP | same as above including RTCP packets [4] that cannot carry RTP Header Extension |
| sc5 | video + video or audio + audio [4] | Similar to sc1, but multiple native audio or multiple native video streams are carried in separate RTP streams with different SSRC, either with different PT or sharing same PT. Packets contain content from a single SSRC. | Different RTP streams are marked as separate PDU Sets, not mixing separate RTP streams (SSRC) in a single PDU Set. |
| sc6 | video + video or audio + audio [4] + RTCP [4] | Same as above adding RTCP | Same as above including RTCP packets [4] that cannot carry RTP Header Extension |

Table 6.22.2.3-2: Guidelines for applying RTP HE in different example multiplexing scenarios

|  |  |  |  |
| --- | --- | --- | --- |
| Scenario | Guideline | Additional Comments | |
| sc1 | Video PDU Sets may be assigned for example for video frames or slices and PDU Set importance can be set using guidelines from 4.6.2 of TS 26.522 [2].  Audio Packets can be unmarked or in case audio frames consist of multiple packets they may be marked using RTP HE.  PDU Set importance of the unmarked packet is determined by the 5G System based on a configuration, and this can also be based on the payload type. | | Typically, RTP HE is used for the video stream, audio packets can be unmarked (see the lone PDU case), or the RTP HE can also be used for the audio stream. | |
| sc2 | Same as above.  RTCP packets cannot be marked using RTP HE and are treated as unmarked packet in the 5G System, PDU Set importance can be determined by the 5G system. | | Same as above for audio and video.  End of Data burst signal may not be valid if RTCP is the last packet in a burst. | |
| sc3 | PDU Sets can be identified by the RTP sender based on the presentation time and the RTP HE can be used to support the PDU Set based QoS handling.  The PDU Set importance can be set to a configured value or the value corresponding to the importance of the most important part of the multiplexed stream using guidelines from 4.6.2 in [2] | | In this case, the grouping of PDU sets will contain different media types, and therefore the guidance cannot only be based on one specific media type, which may not be appropriate. Therefore, PDU Sets could be identified and marked by the RTP sender based on other aspects such as the presentation time.  The PSI can be set based on a configuration. | | |
| sc4 | Same as sc3  RTCP packets cannot be marked and are treated as unmarked packet in the 5G System. | Same as above including RTCP packets [4] that cannot carry the RTP Header Extension.  Data burst signal cannot be used if RTCP is the last packet in a burst. | | | |
| sc5 | Video PDU Sets may be assigned for video frames or slices and PDU Set importance can be set using guidelines from 4.6.2 of TS 26.522 [2], separating RTP streams with different SSRC into separate PDU Sets.  Audio Packets can be unmarked or in case audio frames consist of multiple packets they may be marked using RTP HE, separating RTP streams with different SSRC into separate PDU Sets. | Multiple PDU Sets can be "open" at the same time, i.e., some PDUs are received from multiple different SSRC and thus different PDU Sets, which requires the marking to keep track of multiple simultaneous PDU Set contexts. | | | |
| sc6 | Same as sc5.  RTCP packets cannot be marked and are treated as unmarked packet in the 5G System. PDU Set importance can be determined by the 5G system. | Same as above including RTCP packets [4] that cannot carry the RTP Header Extension.  Data burst signal cannot be used if RTCP is the last packet in a burst. | | | |

To support multiplexed content in combination with PDU Set QoS based Handling in the 5G System, groups of packets of different media types (audio, video) but same payload type (native multiplex) may also be grouped as a PDU Set (sc3). This enables frames/groups of packets to benefit from transfer using PDU Set QoS parameters in NG-RAN (delay budget, PSIHI). In this case, each of the RTP packets can set the RTP Header Extension for PDU Set Marking to enable 5G System to identify corresponding PDU Sets.

Different options exist when applying RTP HE for multiplexed content, for which some guidelines are as follows:

- When RTP multiplexing (sc1, sc2, sc5 and sc6) is used, it is possible to separately mark the PDU Sets in different streams. In this case, the PDU Sets may also be indicated with different PDU Set importance as already discussed in [2]. As concluded in TR 23.700-70, the UPF packet filter can be extended to include SSRC, payload type, etc. in order to detect and map each marked media stream to the specific QoS Flow with PDU Set QoS handling enabled. In this guideline, it is assumed that this is not used.

- When packets may combine different media types in a payload type such as in sc3 and sc4 PDU sets can be created around a common media presentation time grouping packets based on timestamps. Additional sender behaviour can be detailed in case such as solution is selected for normative work. In this case the PDU set importance can be set to a derived or default value.

- In case only packets of single stream are marked (e.g. the video stream), the situation as described in the previous sub-clause applies.

- In case packets cannot carry the RTP header extension (e.g. RTCP packet), packets can be handled as lone/unmarked PDU.

The protocol description can be used to indicate to the 5G System that PDU Sets contain multiplexed content. For example, by indicating a payloadType with a number that corresponds to a multiplexed data type (e.g. payloadType 33 for MPEG-2 TS). In this case the transport protocol would still be RTP, but the payload type could be 33 MPEG-2 TS (native multiplex) and/or a combination of format types e.g. (H.264, AAC) (in case of RTP multiplex). In such a case it is optional to provide additional format parameters.

The guidelines for PDU Set Marking for H.264 and/or H.265 can still apply but in this case, but when the PDU Set importance of MPEG-TS RTP packets cannot be identified, it can be set to a PSI value determined by the sender.

Table 6.22.2.3-3: Example protocol description for different multiplexing scenarios

|  |  |  |  |
| --- | --- | --- | --- |
| Scenario | Protocol | rtpPayloadFormat (example) | Rtppayloadtypelist (example) |
| sc1 | RTP/SRTP | H265, PCMA | 97,8 |
| sc2 | RTP/SRTP | H265, PCMA | 97,8 |
| sc3 | RTP/SRTP | MP2T | 33 |
| sc4 | RTP/SRTP | MP2T | 33 |
| sc5 | RTP/SRTP | H264, H265 or just H265 | 97,98 or just 98 |
| sc6 | RTP/SRTP | H264, H265 or just H265 | 97,98 or just 98 |

Note that any of the encoding types or payload types as registered by IANA [57] and beyond can be used, but this table is not fully up to date in TS 29.571 [55].

There is no way to explicitly indicate the presence of RTCP packets in the protocol description.

### 6.22.3 Proposal

The following is proposed to consider for normative work:

a) Develop these guidelines for multiplexed content and add them to TS 26.522 [2] to support both the PDU Set identification by RTP senders and enable PDU Set information derivation by the UPF if needed.

b) The pduSetMarking property in the M1QoSSpecification in clause 7.3.3.4 of 26.510 [50]: "Indicates that packets at reference point M4 are required to include PDU Set marking if the media transport protocol supports this. Default value false if omitted". This requirement may need to be relaxed or updated.

## 6.23 Solution #23: PDU Set Size and Over Provisioning in RTP HE for PDU Set Marking

### 6.23.1 Key Issue mapping

This is a solution to key issue #1.

### 6.23.2 Description

#### 6.23.2.1 General

PDU Set Size is a field in the RTP Header Extension that can be used by the network to retrieve the PDU Set Size in bytes.

The accuracy of this field is a topic studied in KI #1.

Clause 6.4 and clause 6.7 contain solutions to KI #1.

A solution is also provided in TS 26.522 [2] as well to include the IP packet overhead in the signalled PDU Set Size.

The reasons for deviation between PDU Set size at 5GS and sender are discussed in clause 6.4 and related problems in clause 5.1.

Data in 6.4 indicates that for a PDU sets of 22 Kbytes, a total deviation of up to 1000 bytes may occur e.g. up to maximum of 5 percent (to be confirmed via communication with RAN 2).

#### 6.23.2.2 Usage of PDU Set Size in NG-RAN

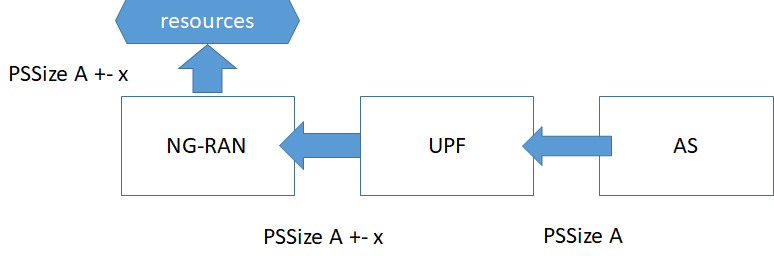


Figure 6.23.2.2-1

Figure 6.23.2.2-1 illustrates the signaling of PDU Set Size A from AS to UPF to NG-RAN.

Application server can add the PDU Set Size to RTP Packets using the RTP Header Extension, signaling PSSize A.

At the UPF after traversal through the network UPF may see PSSize plus or minus X (deviation due to network traversal).

This value A+-x is shared with NG-RAN.

NG-RAN will allocate resources for timely transmission of the PDU Set based in this value.

This operation takes a bit of time, as NG-RANs allocates necessary resources for transmitting the PDU Set.

For this reason, re-allocation is undesirable, as it may introduce additional delay.

Under-provisioning may lead to PDU Set delay budget not being met.

Therefore, for NG-RAN it is better to do a *slight overprovisioning* instead of accidental *under-provisioning.*

Experiments in 6.4 and also comparing the potential header overhead, 5 percent is the upper bound, i.e. up to 1000 bytes for 23 KB PDU Sets.

Highly accurate solutions are attempted in TS 26.522 [2] and clause 6.4 with different mechanisms (out of band signaling) and taking into account specific protocol aspects such as IPv4 versus IP6 usage and IP header overhead.

However, it seems the PDU Set Size is still not fully accurate and *under-provisioning can still occur*. This implies that if an NG-RAN node receives a PDU Set Size that is too small, it allocates resources for it, but when the actually PDU Set is larger it needs to again allocate resources increasing the total delay for transmitting the PDU Set. This may lead to issues especially when the delay budget is small.

**Observation:** Even with the mechanisms in clause 6.4 and TS 26.522 [2] PDU Set Size inaccuracy is still possible, and *under-provisioning* can occur.

Given this observation: *over-provisioning* is an alternative to avoid re-allocation of resources at NG-RAN due to under provisioning.

In this solution we propose to consider *over provisioning* and enable explicit over provisioning.

A) SA4 could recommend over-estimating PDU Set size by a certain percentage (e.g. 5 percent to account for deviations along the way)

**Or**

B) SA4 can align with SA2 and RAN 2 on this matter aiming to adopt a guideline that up to a certain percentage of deviation compared to the *estimated PDU Set Size* from the RTP Header Extension*.*

This would simplify TS 26.522 [2] and avoid problematic *under provisioning*.

#### 6.23.2.3 Discussion

The following analysis of the solution is provided.

**Pro 1:** simpler solution, no out of band signaling, no need for transmission protocol specific operation.

**Pro 2:** explicit about inaccuracy, avoid under provisioning at NG-RAN.

**Con 1**: Some additional resources may be allocated in NG-RAN for PDU Set Transmission, but this is up to RAN implementation and not part of the normative work.

### 6.23.3 Proposal

Inaccuracy in PDU Set Size exists in all current proposals.

Overprovisioning can be a practical approach, mainly targeting to avoid *under provisioning*.

In this solution we propose to consider *over provisioning* and enable explicit over provisioning.

A) SA4 could recommend over-estimating PDU Set size by a certain percent based on RAN requirements and achievability (e.g. 5 percent), to be confirmed based on requirements from RAN.

**Or**

B) SA4 can align with SA-2 and RAN 2 on this matter that up to a certain percentage more resources may be allocated compared to the *estimated PDU Set Size* from the RTP Header Extension*.*

This can enable simplifying TS 26.522 [2] avoiding transport specific operations.

This solution introduces some overhead at NG-RAN, but experimental and evaluation of the overhead in RTP packets, the overhead is bound the percentage of the PDU Set Size (assuming trivial issues like IP fragmentation etc. are avoided).

A benefit is that delay budgets are more likely to be met using this solution as *under-provisioning* can be avoided.

Communication with SA2 and RAN2 will be useful to get feedback on this solution.

With regard to overprovisioning of resources at RAN, this is up to the implementation of the RAN.

## 6.24 Solution #24: Traffic pattern prediction for real-time video communication

### 6.24.1 Key issue mapping

This maps to Key Issue #12.

### 6.24.2 Description

#### 6.24.2.1 The need for traffic pattern indication

The discussion focuses on real-time video traffic. There are two categories: conversational video and XR video (including real time gaming and augmented reality). For conversational video, the delay requirement is 150 ms, while for XR video the delay requirement is 50 ms and 10 ms, as shown in the table 6.24.2.1-1 extracted from [3].

Table 6.24.2.1-1 QoS characteristics for real-time video communications

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 5QI  Value | Resource Type | Default Priority Level | Packet Delay Budget  (NOTE 3) | Packet Error  Rate | Default Maximum Data Burst Volume  (NOTE 2) | Default  Averaging Window | Example Services |
| 2 | (NOTE 1) | 40 | 150 ms  (NOTE 11,  NOTE 13) | 10-3 | N/A | 2000 ms | **Conversational Video** (Live Streaming) |
| 3 |  | 30 | 50 ms  (NOTE 11,  NOTE 13) | 10-3 | N/A | 2000 ms | **Real Time Gaming**, V2X messages (see TS 23.287 [121]).  Electricity distribution – medium voltage, Process automation monitoring |
| 80 |  | 68 | 10 ms  (NOTE 5,  NOTE 10) | 10-6 | N/A | N/A | Low Latency eMBB applications **Augmented Reality** |

Given that the delay requirement is 150 ms for conversational video, the penalty of letting a video frame wait for the next DRX one period at the RAN is likely negligible, whereas this is not the case for XR video, due to the much more stringent delay requirement.

**Observation 1:** For conversational video, the benefit of indicating the delay to the next frame is negligible.

#### 6.24.2.2 Time to the next data burst (TTNB)

For simplicity, we consider a data burst that consists of a video frame. For the more general case of data burst, it will be even more difficult to predict the TTNB. For conversational video, two factors affect the TTNB:

- **Video encoding time:** The video encoding time depends on the complexity of the scene. For the same target frame size, typically the more complex the scene is, the long it takes to complete the encoding. It may be difficult for the sender to predict the scene and the video encoding time.

- **Rate adaptation:** The frame rate is part of the rate adaptation, e.g., triggered by congestion control. With rate adaptation, the delay to the next frame may change. If a rate reduction is requested after the time to the next frame was predicted and sent in the current PDU Set, the prediction may be become obsolete. This is illustrated in Figure 6.24.2.2-1. This issue can be avoided if the sender delays the rate adaptation until the predicted frame is transmitted.



Figure 6.24.2.2-1: Time to the next data burst is obsoleted by a rate reduction request triggered by congestion control

For some XR split-rendering implementations, TTNB may be fixed. In such case, it may be more efficient to use control-plane signaling.

**Conclusion 1:** For some XR split-rendering implementations, it may be beneficial to indicate the time to the next burst using control-plane approach.

#### 6.24.2.3 Burst size

When a data burst consists of a video frame, as noted in Solution #16, if a packager generates all packets of the burst at once, there is no delay in knowing the burst size and there is no error in the data burst size. However, a data burst may consist of multiple PDU Sets according to TS 23.501 [3]. For example, when a data burst consists of multiple application data units (ADUs) such as an audio frame and video frame, to avoid buffering which introduces latency, the sender needs to predict the size of the ADUs that come after the first ADU. This is illustrated in Figure 6.24.2.3-1.

NOTE 1: TS 23.501 [3] allows a data burst to include multiple PDU Sets. The benefit of such a data burst for low-latency communication is FFS.

NOTE 2: The PDU Set is not necessarily the minimum unit of traffic. A data burst does not have to be composed of PDU Sets.

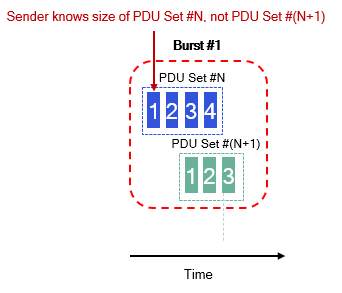


Figure 6.24.2.3-1: The sender needs to predict the size of PDU Set #(N+1) when generating the burst size

Therefore, there is a need for predicting the size of the next ADU. When doing prediction, one way is to use the past frame sizes of a particular media stream to predict the size of the next frame of the same media stream. *This reduces the burst size prediction problem to one of predicting the frame size.* However, it needs to be verified that such prediction is feasible. To test it, an experiment was carried out. The setup is shown in, the HMD is connected to a split rendering server that performs split rendering via Wi-Fi 6 (IEEE 802.11ax). The content is Steam VR with complex graphics. The video codec is a commercial hardware HEVC codec with the IPPP GOP structure. The video frame rate is driven by the display refresh rate, which is 90 FPS. The implementation of the split rendering server is proprietary, not based on WebRTC.

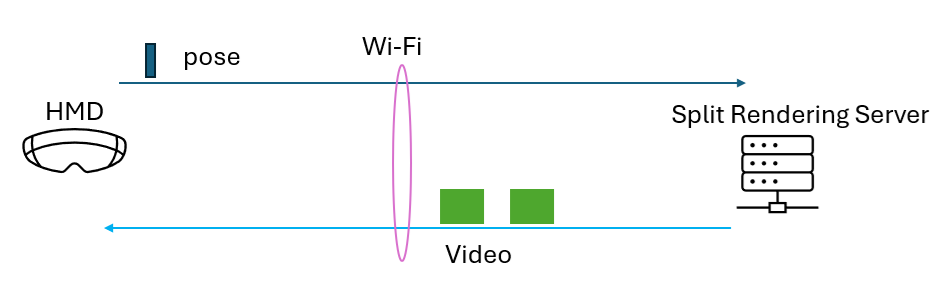


Figure 6.24.2.3-2: Experimental setup for XR video

The video frame size for the left video stream as a function of the frame number is shown in Figure 6.24.2.3-3.

A graph showing a graph

Description automatically generated with medium confidence

Figure 6.24.2.3-3: The frame size of the left video

Some prediction algorithms are tested, and the results are shown in Figure 6.24.2.3-4:

- EWMA (exponentially weighted moving average). The weight is selected to achieve the smallest prediction error.

- MA (moving average): the prediction is set to the average of the frame sizes in a sliding window of most recent frames. The size of the window is 5 and is selected to achieve the smallest prediction error.

- Past observation: the last frame size is used as the prediction of the current frame size.

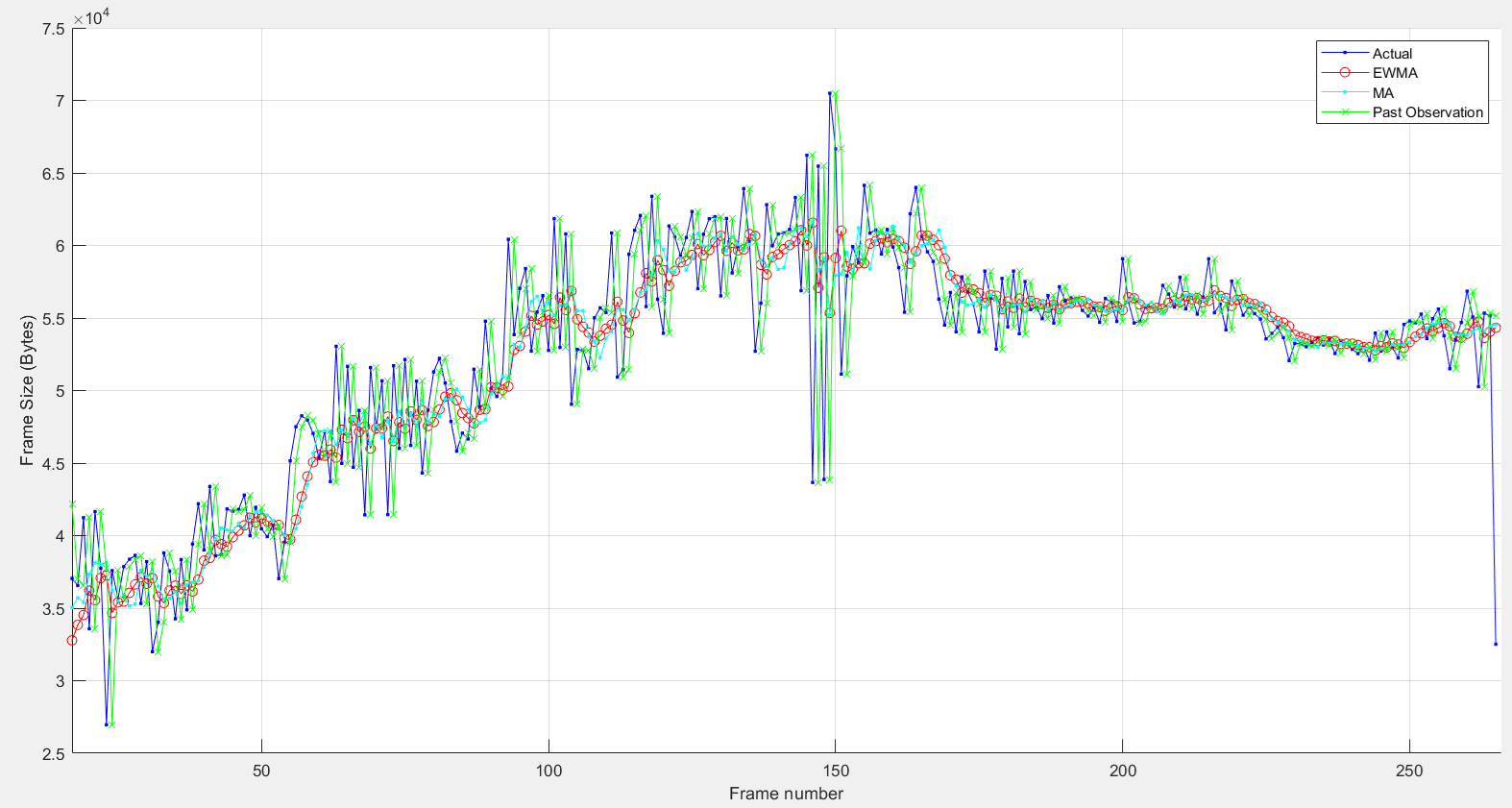


Figure 6.24.2.3-4: The frame size predictions of the left video: actual frame size (blue), EWMA prediction (red), MA (cyan), and past observation prediction (green)

The absolute value of the average prediction error (normalized by the average frame size) is: 4.76% for EWMA, 4.75% for MA, and 5.28% for the past observation method.

The distribution of the prediction error for the best performing method – MA – is show in Figure 6.24.2.3-5. It is seen that although the average error is 4.75%, the error varies a lot, as much as 30%.

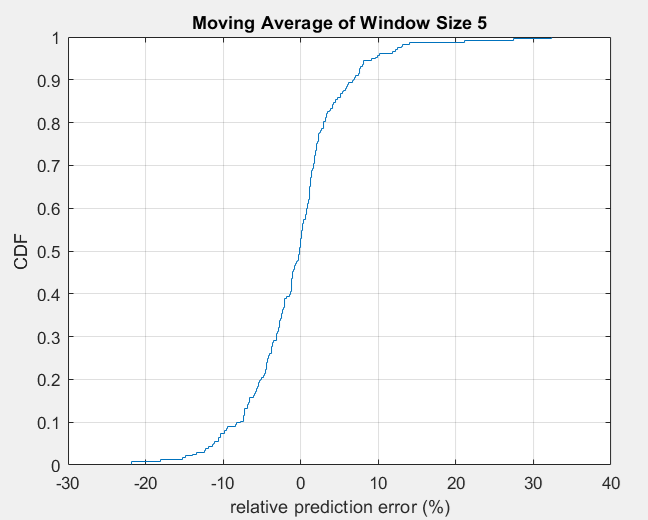


Figure 6.24.2.3-5: CDF of the prediction error (relative to the mean frame size) for the MA prediction method

**Observation 1:** for XR video, the prediction error for the next frame size can be significant.

**Observation 2**: Given the large prediction error, it is not clear what the RAN will do with the predicted size.

**Conclusion 2:** for XR video, if the prediction of the next video frame size is used in the data plane, the prediction accuracy (e.g., the 99% confidence interval) needs to be indicated along with the prediction.

## 6.25 Solution #25: Definition of the time to the next data burst

### 6.25.1 Key Issue mapping

This maps to Key Issue #12.

### 6.25.2 Possible definitions and a comparison

The starting time of the time to the next data burst (TTNB) can be defined in multiple ways:

- Option 1: the starting time is the departure time of the first packet of the first PDU Set. This is shown in Figure 6.25.2-1 (a).

- Option 2: the starting time is the departure time of the last packet of the last PDU Set. This is shown in in Figure 6.25.2-1 (b).

The benefit of Option 1 is that it gives more time for a network node (e.g., base station) to prepare for the next data burst. Also, since the packet carrying the TTNB is the first packet of the first PDU Set, the delay of the transmission of the packet due to contention with other packets is minimized, thus making the indicated TTNB more accurate. The drawback is that the prediction is farther into the future and hence a larger prediction error.

The benefit of Option 2 is that the prediction error can be reduced compared to Option 1 because the prediction is for a shorter time interval. On the other hand, it gives a network node less time to prepare for the next data burst, and the delay of the transmission of the TTNB-carrying packet (e.g., the packet #3 in PDU Set #(N+1)) may be large due to contention with other packets.

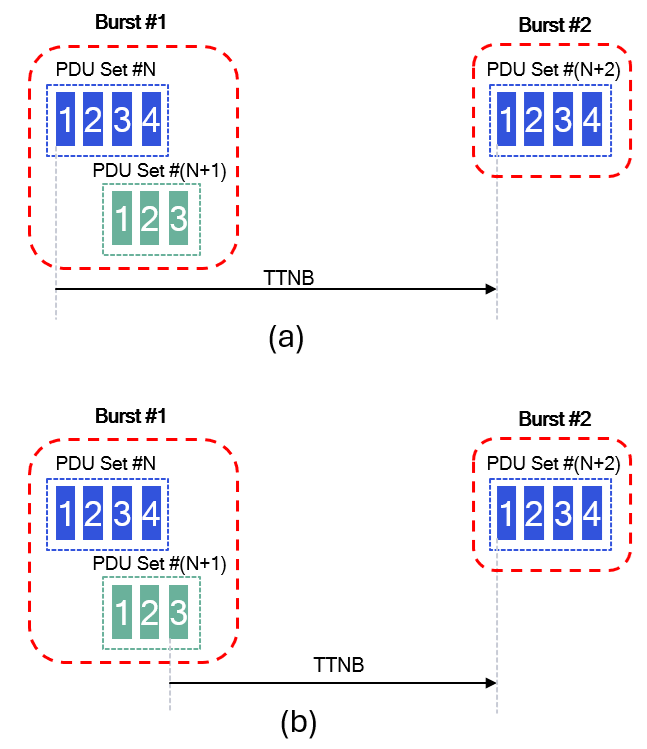


Figure 6.25.2-1: The starting time for the TTNB:  
(a) option 1: the first packet of the first PDU Set,  
(b) option 2: the last packet of the last PDU Set

**Proposal:** consider one of the following definitions of the starting point of the TTNB in normative work:

- Option 1: the starting time is the departure time of the first packet of the first PDU Set.

- Option 2: the starting time is the departure time of the last packet of the last PDU Set.

NOTE: This needs to be coordinated with SA2/RAN2.

## 6.26 Solution #26: RTP HE Enhancements for Data Boosting Indication

### 6.26.1 Key Issue mapping

This solution provides analysis on the matter of DL data boosting and applies to KI#12.

### 6.26.2 Description

In practice, XR and immersive services may experience some scenarios in which the video service is interrupted or halted for a short time e.g. rebuffering the data to continue to video service. An illustrative example may be:

1) The user starts a new XR conversational WebRTC video service from a WebRTC client via the 5G.

2) For an XR conversational WebRTC video service, the WebRTC Framework can continuously monitor the latency and/or the data rate experienced. When the latency or data rate are not good enough to support acceptable QoE/QoS, the WebRTC Framework (e.g., the RTP sender in an RTC AS) can send a Data Boosting indication in the RTP HE to request the 5G system to provide more radio resources to improve the data rate and decrease the latency.

To improve the user experience, one way is to dynamically boost the available bit rate and reduce the latency. To reduce the latency, the new conversational video data needs to be more quickly transmitted to the user. However, the NG-RAN does not know such conversational video data needs to be transmitted to the user more quickly and still transmits the conversational video data in the normal scheduling way.

**Observation 1: In order to reduce the latency of XR conversational video data in some scenarios, the AS needs to provide data boosting indication to the 5G network, and then the 5G network provides direct or indirect data boosting indication to the NG-RAN.**

SA2 has agreed to define that such data boosting (namely "expedited transfer indication") enabled firstly by the AF during the PDU Session Establishment procedure, then the AS send out the data boosting indication in the RTP HE of traffic payload of XR application. Data boosting may be used for non-GBR QoS flow only. SA2 has agreed to use different 5QI for the data boosting, i.e. before the data boosting, the application traffic is transmitted with a given 5QI (e.g. 20 ms PDB). After the data boosting is received in the UPF, the UPF will move the application traffic into a QoS Flow with high performance 5QI (e.g. 5 ms PDB).

One key part of data boosting is how the AS can provide in-band data boosting indication to the 5G network. One simple way is to define the data boosting indication in the RTP HE (e.g., the RTP HE for PDU Set marking). The AS needs to stop providing the data boosting indication to the 5G network after some time if the data rate and or latency is sufficient for the application. However, based on static local policies (as indicated by the PCF), the 5G network can still stop the data boosting even if the DL data carry the data boosting indication.

**Observation 2: The AS can provide in-band data boosting indication to the 5G network with the data needed for boosting in the RTP HE. The AS stops providing the data boosting indication to the 5G network after some time.**

**Observation 3: Based on local policies the 5G network can reject or stop the data boosting even if the DL data is with a data boosting indication.**

### 6.26.3 Conclusion

Based on the gap analysis in the above, it is proposed to make the following conclusions.

1. **Observation 1: In order to reduce the latency of conversation video data in some scenarios, the application needs to provide data boosting indication to the 5G network, and then the 5G network provides direct or indirect data boosting indication to the NG-RAN.**
2. **Observation 2: The AS can provide in-band data boosting indication to the 5G network with the data needed for boosting in the RTP HE. The AS stops providing the data boosting indication to the 5G network after some time.**
3. **Observation 3: Based on local policies the 5G network can reject or stop the data boosting even if the DL data is with a data boosting indication.**

## 6.27 Solution #27: Conveying AL-FEC information to the network

### 6.27.1 Key Issue mapping

This maps to Key Issue #4.

### 6.27.2 Description

As observed in Solution #8 (clause 6.8), the network complexity is reduced if the source packets and the repair packets of the same ADU are grouped into the same PDU Set as opposed to different PDU Sets.

As explained in clause 5.4, exposing AL-FEC information is needed for AL-FEC aware network resource allocation.

There are two approaches.

**User-plane approach:**

It is noted in Solution #5 (clause 6.5) that for the packet formats considered there, the AL-FEC information for FlexFEC codes and ULPFEC codes is invisible to an intermediate network entity such as the UPF when the RTP payload is encrypted. Similarly, if the RTP/UDP is the transport protocol for Reed-Solomon FEC codes, the same invisibility issue arises.

Therefore, it is important to define new RTP packet formats to make the AL-FEC information visible to the network.

For MDS codes (including approximate MDS codes), although successful decoding only depends on how many PDUs are received, there is still a benefit to know whether a PDU is a source packet or a repair packet because in the case where all source PDUs are received, the decoding is trivial.

For the FlexFEC and ULPFEC codes, the dependence information between the source packets and the repair packets is also important for decoding.

Given that the PDU Set information is currently conveyed by the RTP header extension for PDU Set Marking, the same RTP header extension can be enhanced to expose AL-FEC information.

**Control-plane approach:**

The control-plane approach may be an efficient approach to conveying relatively static information and can be considered for conveying the AL-FEC information for MDS (including approximate MDS) codes. However, it is not suitable for carrying highly dynamic information such as the dependence between source packets and repair packets in the FlexFEC codes.

## 6.28 Solution #28: Control-Plane Solution to the Key Issue on Enhancements of Data Burst Marking

### 6.28.1 Key Issue mapping

This maps to Key Issue #12.

### 6.28.2 Description

3GPP TR 26.926 [60] presents a periodic traffic model for XR split-rendering traffic, e.g., in Annex B.2 of [60]. For such traffic pattern, a potentially efficient design option is to signal the traffic pattern in the control plane.

There is an existing mechanism, Time Sensitive Communication Assistance Information (TSCAI) in 3GPP TS 23.501 [3] and TS 38.413 [61], to signal a traffic pattern in the control plane. TSCAI carries the periodicity as a mandatory field, Burst Arrival Time, Burst Arrival time window among others as optional fields. The Burst Arrival Time is the latest possible time when the first packet of the (first) data burst arrives at either the ingress of the RAN (downlink flow direction) or the egress of the UE (uplink flow direction). According to Table 5.27.1.2-1 in 3GPP TS 23.501 the following parameters are available in TSCAI:

1) flow direction (up link or down link),

2) periodicity,

3) burst arrival time,

4) survival time (time the application can survive without the burst),

5) burst arrival time window,

6) capability for BAT adaptation,

7) jitter information (optional),

8) periodicity range.

NOTE 1: Parameters like burst size or other dynamic traffic characteristics are currently not available in TSCAI. TSCAI is constructed by SMF based on information provided by the application to the 5G system. TSCAI can assist the RAN in scheduling. It can also support time-varying traffic patterns. When the traffic pattern changes, SMF can send an updated TSCAI to the RAN.

**Observation 1:** The current TSCAI mechanism can convey a burst traffic pattern to the RAN and, if the traffic pattern changes, can convey an update of the burst traffic pattern.

The RAN can enable UE power saving to different degrees. If the RAN does not want to enable maximum UE power saving, it is not necessary for the RAN to know when the first data burst with the new periodicity will arrive. Instead, the RAN can start detecting the arrival of the first data burst after it receives TSCAI message until it detects a change in the periodicity, as shown in Figure 6.28.2-1 where the periodicity changes from T1 to T2.

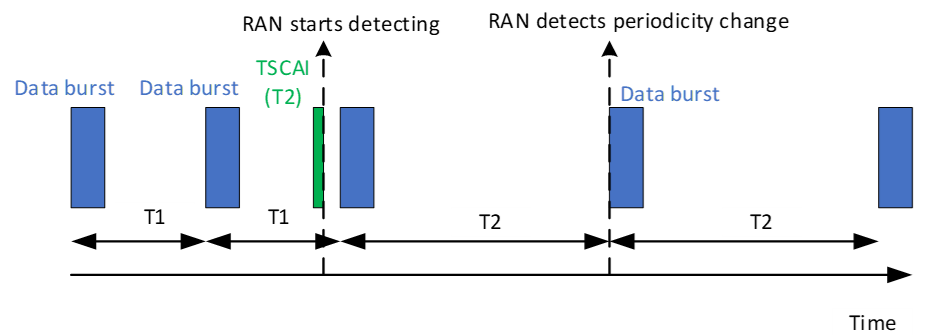


Figure 6.28.2-1: Control plane signaling of change in traffic pattern without signaling the Burst Arrival Time

This leads to the first design option:

**Design option 1:** The TSCAI signals the periodicity to the RAN if the traffic periodicity changes, and the RAN infers the start time of the first data burst after the fact by detecting the new periodicity based on the arrivals of data bursts after the reception of the TSCAI message.

On the other hand, if the RAN wants to enable maximum UE power saving, it needs to know when the new traffic pattern starts. TSCAI can provide the Burst Arrival time for this purpose. However, the drawback is that this requires to synchronize the clock at the application and the 5G clock, which may not be feasible for all real-world implementations. Additionally, for XR deployments, this means that the XR traffic source needs to be able to predict when the data burst will arrive at the gNB with high accuracy, which in turn requires the XR traffic source to know the delay to the gNB, which may be difficult to know in practice.

The operation 5.37.8.2 of TS 23.501 indicates how 5G System can be provided with periodicity information. The following steps are described in 5.27.8.2 of TS 23.501.

The Application Function (AF) can provide periodicity information to the PCF via NEF or directly to the PCF when the AF is trusted. If periodicity information is available at the PCF, the PCF can share this with the SMF based on operator policy including a request to the UPF to perform N6 traffic parameters measurement within PCC rules.

Upon reception of a PCC rule with Periodicity information, the SMF determines the TSCAI and forwards it to the NG-RAN. If the PCC rule indicates to perform N6 Traffic Parameter measurements, the SMF requests the UPF to monitor and periodically report the N6 Traffic Parameters (i.e. the N6 jitter range associated with the DL Periodicity and, if not provided by the AF, UL/DL periodicity) using the N4 Session Modification procedure, see clause 5.8.5.11. If the measurement of N6 jitter range associated with the DL Periodicity is required and the DL Periodicity is available at the SMF, the SMF also sends the DL Periodicity to the UPF. The UPF reports the measured N6 Traffic Parameters to SMF via N4 interface.

The way how the N6 jitter and periodicity (when it is not provided by the AF) is derived by the UPF is implementation dependent.

**Observation 2:** The current TSCAI mechanism with Burst Arrival Time requires time synchronization between the application and the 5G clock and the knowledge of the delay from the application traffic source to the RAN, and these requirements may not be always feasible.

The PDU Set based QoS framework in Rel-18 leads to the definition of the PDU Set RTP header extension which carries the PDU Set Sequence Number (PSSN) [2]. This provides an alternative to the current Burst Arrival Time. Specifically, the application can be able in some cases to indicate at which PDU Set the new traffic pattern will start, and the RAN considers the new traffic pattern starts when the PDU Set arrives. To give the RAN lead time for scheduling, a time offset may also be indicated.

NOTE 2: The PSSN in the RTP Header extension and the PSSN in GTP-U Header provided by the UPF to NG-RAN are not necessarily the same, as the UPF is responsible for deriving the PSSN for different PDU Sets and can modify or adapt the PSSN value from the RTP HE. This is because UPF always needs to add the PSSN in GTP-U Header, for RTP HE packets and for non RTP HE packets (also known as lone PDUs).

NOTE 3: The potential issue (discrepancy between the PSSN signaled in the control plane and the PSSN carried in the GTP-U packet header) is resolvable, e.g., if a predetermined value is used for lone PDUs as proposed in Solution #15 in clause 6.15.

NOTE 4: NG-RAN accessing the PSSN in the RTP packet is undesirable due to security concerns (NG-RAN can typically not inspect L3 or higher layer information).

An example is shown in Figure 6.28.2-2, where the control plane signaling conveys that "The new traffic pattern with new periodicity 16.7 ms starts time\_offset (in ms) after the arrival of the first PDU of the PDU Set with PSSN=8".

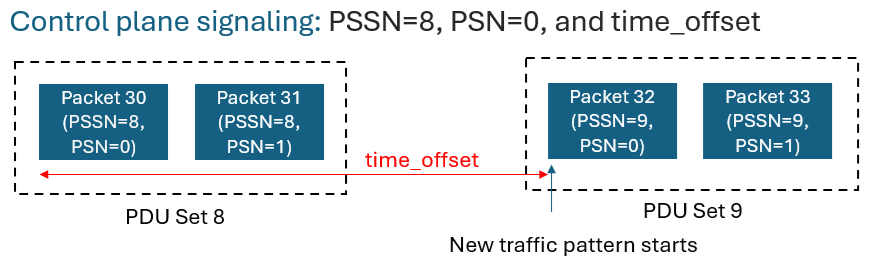


Figure 6.28.2-2: An example on control plane signaling of change in traffic pattern

**Observation 3:** The PDU Set Sequence Number (PSSN) could in some cases synchronize the update of a traffic pattern between the application and the 5G system.

This leads to a second design option:

**Design option 2:** The TSCAI signals the periodicity, the PSSN of the first PDU Set in the new traffic pattern and possibly a time offset. The arrival time of the PDU Set, delayed by the time offset, if present, is considered as the start of the new traffic pattern.

The traffic source may have multiple traffic flows destined to the receiver, e.g., text and video. The packets of low-latency traffic may be in PDU Sets and other packets may not. The traffic pattern is the supposition of the PDU Sets and other packets. Then the question is whether using PSSN will convey wrong information about the burst traffic pattern. This is not the case because what the RAN needs to do is to timely accommodate the bursts of low-latency traffic and can buffer the other packets (which are non-low-latency traffic) until it gets an opportunity to transmit the buffered packets to the UE. The delay from buffering is acceptable for the other packets (which are non-low-latency traffic).

**Observation 4:** The PDU Set Sequence Number (PSSN) based traffic pattern start indication can ensure the low delay for low-latency traffic while at the expense of non-low-latency traffic which is acceptable.

The periodicity may be negotiated at the session setup or updated during a session.

In an XR session, the application server may send multiple streams of media, such as a video stream and an audio stream, which are both periodic but with different periodicities. The superposed traffic pattern is not periodic. However, if the traffic source lets the RAN know the periodicities and the start times, the RAN can predict the traffic pattern accurately. This is similar to the superposition of sinusoidal signals with different periodicities, in which case knowing the periodicities and phases is sufficient to fully determine the superposed signal.

The UPF may map a PSSN in an RTP header extension to a different value in a GTP-U packet header if the UPF creates new PDU Sets for lone PDUs. In this case, the PSSN for the first PDU Set of a new periodicity signalled in the control plane may mismatch the PSSN of the GTP-U packet encapsulating the first PDU Set, and this can cause a timing error when RAN respond to the start of the new periodicity.

NOTE 5: Additional requirements or solutions to prevent this behaviour at the UPF are FFS. A potential solution is to use a predetermined PSSN value for lone PDUs as proposed in Solution #15 in clause 6.15. This means that the traffic source will not use the pre-determined PSSN for regular PDU Sets and the UPF will apply the PSSN value to lone PDUs. The latter involves changes to the UPF behavior. Another potential solution is to divide the PSSN space into two subspaces, one for regular PDU Sets and the other for lone PDUs. Similarly, this will involve changes to the UPF behavior.

The signaling path of the proposed TSCAI enhanced with PSSN is the same as that of the current TSCAI, and therefore the delays will be similar. The communication path of the control plane solution for sharing PSSN or time offset is as follows: UE to AF, AF to PCF, PCF to SMF, SMF to RAN.

To summarize:

**Pros:**

- For scenarios where the traffic burst pattern is periodic and the periodicity changes infrequently, the proposed control plane approach can be more efficient than the user plane approach.

- The proposed control plane approach can be implemented by slightly augmenting the existing TSCAI framework.

**Cons:**

- For other scenarios (i.e., frequent change in the periodicity), the proposed control plane approach is likely to be less efficient than the user plane approach. The TSCAI may have a setup time due to the different steps that may make it unsuitable for frequent and dynamically changing periodicity.

- Burst size is not supported in this approach which is a common traffic characteristic.

- Time synchronization between the application and the 5G clock is needed.

- Issue of consistency of the PSSN (PDU Set sequence number) in the control plane.

### 6.28.3 Proposed solution

The above observations and analysis lead to the following proposal.

**Proposal:** TSCAI without the Burst Arrival Time can be used to indicate a new traffic pattern. Whether TSCAI can be enhanced by incorporating the PDU Set Sequence Number (PSSN) and possibly a time offset to indicate the start time of a traffic pattern is FFS as it is not clear PSSN can be used reliably to support this case. Also, TSCAI enhancement need to be coordinated with SA WG2.

# 7 Conclusions

## 7.1 Key issues progress

Table 7.1-1: Summary of key issue progress

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key Issue # | Short Description | Progress | Objectives met | Normative work proposed | Convergence possible? | Dependencies |
| 1 | PSSize accuracy | Solutions #4, #7 and #23 | Partially | Yes #4, Yes #7, Yes #23 | #4 and 7 are complementary solutions  #23 is not complementary | #7 depends on SA2  #4 is fully contained in SA4  #23 has dependency on RAN feedback |
| 2 | Lone PDU | Solutions #2, #15,  #20 and #21 | Partially | No #2,  Yes #15  Yes #20 | #2 provides gap analysis.  Complementary solutions in #15 and #20 | SA2 |
| 3 | FEC support | Solutions #5, #17 and #18 | Yes | Yes #5, Yes #17, Yes #18 | Complementary solutions FEC schemes, congestion control and FEC awareness | None |
| 4 | FEC awareness | Solutions #8, #10, #17, #18, #19, #27 | Yes | Yes #8 Yes #10 Yes #17 Yes #18 Yes #19 | Complementary solutions | RAN2, SA2 |
| 5 | RTP transport for XR | Not progressed | - | - | - | - |
| 6 | Encryption | Solution #3 | Yes | Yes (but limited) | Single solution | None |
| 7 | RTCP messages for 5G | Solution #14 | Yes | No | Single solution | None |
| 8 | RTP retransmission | Solutions #9 and #11 | Yes | Yes #9, Yes #11 | Complementary solutions | SA2, RAN2 |
| 9 | RTP Multiplexing | Solutions #12  #22 | Yes | Yes #12 and #22 | Complementary solutions | SA2 |
| 10 | Use cases | Not progressed | - | - | - | - |
| 11 | Enhancements to PDU Set HE | Not progressed | - | - | - | - |
| 12 | Data Burst marking | Solutions #6, #13, #16, #24, #25, #26, #28 | Yes | Yes #6, Yes #16,  #13: Reinforces the normative work proposed in #6 and #16,  #24: No  #25: Yes  #26: Yes  #28: No | Some solutions are complementary and convergence needed for others, possible way forward proposed in clause 7.13. | SA2, RAN2 |
| 13 | Applicability of PDU Set HE | Solution #1 | Partially | Yes #1 | Single solution | SA2, RAN2 |
| 14 | QoS flow multiplex | Solution #12 | Partially | Yes | Single solution | SA2 |
| 15 | Media and metadata | Not progressed | - | - | - | - |

## 7.2 Conclusions for Key Issue #1

**Inaccuracy of the PDU Set Size (PSSize) information**

The following aspects are concluded as principles for the normative work:

- Maintain Rel-18 general principle and verify (by means of LS to: RAN2, cc: SA2) the SA4 assumptions regarding the RAN accuracy needs regarding the PDU Set size values.

- Consider towards normative work KI#1 candidate solutions from Rel-18 and Sol#4 from this report, conditionally based on RAN2 feedback as:

o If exact PDU Set Size accuracy is required by RAN consider Rel-18 towards normative work effectively solving the NAT46/64.

o If PDU Set Size accuracy may slightly vary at RAN consider Sol#4 from this report towards normative work based on mechanisms such as RTCP feedback to indicate the correction factor.

The solution on PDU Set Size overprovisioning may be generally regarded as implementation aspect in determining the PDU Set size in the GTP-U header for PDU Set Information. Based on RAN2 feedback, PDU Set Size overprovisioning informative guidelines may be considered.

## 7.3 Conclusions for Key Issue #2

**QoS handling requirements for lone PDU**

The following aspects are concluded as principles for the normative work:

- Extend the RTC provisioning feature in TS 26.510 [50] and TS 26.113 [58] to include PDU Set Importance values for PDUs of protocols that may be treated as lone PDUs in the UPF.

- Consider guidelines for handling lone PDU in TS 26.522 [2].

NOTE: Coordinate with SA2 on whether Protocol Description needs to be extended with the lone PDU information.

## 7.4 Conclusions for Key Issue #3

**Enhancements for application-layer FEC support**

Existing AL-FEC and congestion control schemes were studied in this KI.

No recommendation for normative work to include new AL-FEC schemes in the 3GPP specifications was identified.

## 7.5 Conclusions for Key Issue #4

**AL-FEC awareness for PDU Set handling**

The following aspects are concluded as principles for the normative work:

- Agree on supporting PDU Set handling with AL-FEC awareness in Rel-19 5G\_RTP\_Ph2 normative work.

NOTE 1: The agreement is conditioned by RAN confirmation to feasibility of using content ratio information for discarding DL PDUs during congestion for RLC AM/UM mode based on the above SA2 principles in Rel-19. This would apply for success of delivery of a group of packets.

- Specify any necessary (S)RTP HE enhancements for PDU Set marking with AL-FEC awareness.

NOTE 2: To realize Stage-3 aspects of the agreed SA2 design over 5G-RTC other impacted technical specifications are not precluded (e.g., TS 26.510 [50], TS 26.113 [58]).

- Specify requirements and guidelines for MDS AL-FEC coding schemes necessary for PDU Set handling with AL-FEC awareness by the 5GS.

NOTE 3: A generic mechanism to improve congestion control algorithms for AL-FEC encoded traffic considering intentional packet discarding by the network is FFS.

## 7.6 Conclusions for Key Issue #5

**RTP transport of XR metadata**

This key issue was not progressed, hence no recommendation for normative work.

## 7.7 Conclusions for Key Issue #6

**PDU Set marking for XR streams with RTP end-to-end encryption**

The following aspects are concluded as principles for the normative work:

* Extend the guidelines for support of SRTP in such cases in TS 26.522 [2].

No additional normative work has been identified for Rel-19 since SRTP usage is already supported in Rel-18.

## 7.8 Conclusions for Key Issue #7

**Existing RTCP messages and RTP header extensions to better support XR services in 5G**

Existing RTCP messages and RTP HEs were studied in this KI.

No potential normative work has been identified for Rel-19.

## 7.9 Conclusions for Key Issue #8

**RTP retransmission in supporting XR services in 5G**

The following aspects are concluded as principles for the normative work:

- Coordinate with SA2 and RAN2 on network awareness of retransmitted PDUs as well as core network and RAN handling of retransmitted PDUs based on the information provided by the application.

- Based on SA2 and RAN2 guidance, consider sending information related to end-to-end retransmissions from the application to the 5G Core Network.

## 7.10 Conclusions for Key Issue #9

**Feasibility of RTP multiplexing options for transport of XR media streams**

The following aspects are concluded as principles for the normative work:

- Based on response from SA2, normative work on multiplexed RTP streams may be needed. Furthermore, it is recommended to add guidelines to TS 26.522 [2] for RTP senders that use multiplexing.

## 7.11 Conclusions for Key Issue #10

**Use cases and intended deployment scenarios for enhancements of RTP header extension for PDU Set marking**

This key issue was not progressed, hence no recommendation for normative work.

## 7.12 Conclusions for Key Issue #11

**Enhancements of RTP header extension for PDU Set marking**

This key issue was not progressed, hence no recommendation for normative work.

## 7.13 Conclusions for Key Issue #12

**Enhancements of Data Burst Marking**

The following aspects are concluded as principles for normative work:

- Do normative work for signaling burst size, when deterministically known, from RTP senders in a HE.

- Revisit definition of a data burst in TS 26.522 [2] to indicate what is meant by idle time and if it is required.

NOTE 1:Communication with SA2 and RAN2 might be necessary about the SA4 work on the definition of data burst.

- Enable the application to provide data boosting indication to the 5G network for downlink using RTP/RTCP signaling.

- Define TTNB in coordination with SA2 and RAN2.

NOTE 2: RAN2 has indicated that TTNB may be useful if provided in time and is reliable. SA4 needs further evaluation before proceeding with normative work.

## 7.14 Conclusions for Key Issue #13

**Applicability of the RTP header extension for PDU Set marking to different PDU Set types**

The following aspects are concluded as principles for normative work:

- Consider extending the PSI guidelines in TS 26.522 [2] for the case when a PDU Set is defined as a tile (as opposed to a video frame or slice).

- Coordinate with SA2 and RAN2 on potential benefits of signaling PDU Set type to the 5G network.

## 7.15 Conclusions for Key Issue #14

**Traffic detection and QoS flow mapping for multiplexed media stream data flows**

The following aspects are concluded as principles for normative work:

- Based on response from SA2, normative work on multiplexed RTP streams may be needed. Furthermore, it is recommended to add guidelines to TS 26.522 [2] for RTP senders that use multiplexing. There may be potential normative aspects to be added to TS 26.510 [50].

## 7.16 Conclusions for Key Issue #15

**Media and metadata delivery over multiple sessions**

This key issue was not progressed, hence no recommendation for normative work.

Annex A:  
Change history

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
| 2024-04 | SA4#127-bis-e | S4-240795 | - | - | - | Implementing S4aR240017, S4-240604, S4-240657, S4-240771, S4-240772, S4-240773, S4-240777, S4-240769, S4-240829, S4-240830 | 0.0.1 |
| 2024-05 | SA4#128 | S4-241293 | - | - | - | Implementing S4-240968, S4-241012, S4-241018, S4-241096, S4-241191, S4-241260, S4-241271, S4-241275, S4-241283, S4-241285, S4-241312, S4-241314 | 0.1.0 |
| 2024-08 | SA4#129-e | S4-241486 | - | - | - | Implementing S4aR240032 | 0.1.1 |
| 2024-08 | SA4#129-e | S4-241672 | - | - | - | Implementing S4-241444, S4-241497, S4-241503, S4-241538, S4-241649, S4-241650, S4-241651, S4-241657, S4-241658, S4-241659, S4-241660, S4-241663, S4-241747, S4-241749, S4-241751, S4-241763 | 0.2.0 |
| 2024-09 |  |  |  |  |  | Version 1.0.0 created by MCC for presentation to TSG SA | 1.0.0 |
| 2024-11 | SA4#130 | S4-241939 | - | - | - | Implementing S4aR240054, S4aR240080, S4aR240107, S4aR240108 | 1.0.1 |
| 2024-11 | SA4#130 | S4-242189 | - | - | - | Implementing S4-241903, S4-241906, S4-241909, S4-242048, S4-242124, S4-242127, S4-242137, S4-242141, S4-242142, S4-242143, S4-242144, S4-242162, S4-242183, S4-242184, S4-242185, S4-242190 | 1.1.0 |
| 2024-11 | SA4#130 | S4-242246 | - | - | - | Removing incorrectly implemented S4-242190 | 1.2.0 |
| 2024-11 | SA4#130 | S4-242255 | - | - | - | Implementing S4aR250002, S4aR250003 | 1.3.0 |