**3GPP TSG-SA WG4 Meeting #127-bis-eS4-240598**

**Online, 8th – 12th Apr 2024**

**Source: Huawei, HiSilicon**

**Title: [5G\_RTP\_Ph2] On documenting key issue #3 FEC enhancement for AL-FEC**

**Document for: Discussion**

**Agenda Item: 10.8**

*Abstract: This contribution elaborates the goals and objectives of FEC enhancement for AL-FEC in order to document it concisely in the technical report.*

# Introduction

The new SA4 Rel-19 study item on “5G Real-time Transport Protocol Configurations, Phase 2” (5G\_RTP\_PH2) has been approved in [SP-240065](https://www.3gpp.org/ftp/TSG_SA/TSG_SA/TSGS_103_Maastricht_2024-03/Docs/SP-240065.zip). The work item lists twelve distinct key issues to improve 5G RTP as defined in TS 26.522, of which key issue number 3 relates to Application layer Forward Error Correction (FEC):

***3. Enhancements for application-layer FEC support.*** *According to clause 5.7.4 of TR 26.926 [6], commercial XR split rendering and cloud gaming services use Application Layer Forward Error Correction (FEC). This clause also introduces several RTP based FEC schemes defined by IETF primarily to be used in WebRTC. It is worthwhile to study if any of these FEC schemes can be added to 3GPP specifications, for example to support split rendering.*

In this paper we propose to elaborate the goals and objectives of this key issue in order to document it concisely in the technical report.

# Key Issue Number 3 General

## 2.1 General

The text of key issue mentions that TR 26.926 already introduces the solutions from IETF and example commercial implementations. With regard to adding FEC to specifications, it is seems AL-FEC is currently not excluded from TS 26.522 so the goal of the key issue should be to provide information about AL FEC in RTP to improve the support and also to start key issue 4.

## 2.2 Text from TR 26.926

The following text is quoted directly from TR 26.926as to document the current level of documentation around AL FEC in XR communications

*“Commercial XR split rendering and cloud gaming services use Application Layer Forward Error Correction (FEC). For example, Nvidia CloudXR™ supports FEC as indicated here* [*https://web.archive.org/web/20240129165957/https:/forums.developer.nvidia.com/t/possible-to-configure-tune-cloudxr-encoding/208977/3*](https://web.archive.org/web/20240129165957/https:/forums.developer.nvidia.com/t/possible-to-configure-tune-cloudxr-encoding/208977/3)*. Other cloud gaming and XR services also report about the use of FEC.*

*In the following a possible implementation for application layer FEC assuming the system model for XR traffic (see Figure 5.2.1-1) is described. Commercially available XR split rendering and cloud gaming services as introduced above follow the same or at least similar principles.*

*In this case, the Application Data Units (ADUs) are not sent directly to the network, but they are added to a source block that then generates packets of basically equal size in order to then distribute the content. The basic concept is shown in Figure 5.7.4-1.*

*Each Application Data Unit (for example a video frame, or an object) has assigned a size F and additional properties, for example the type of the ADU, its importance, its delay constraints and so on. The properties are typically different for each ADU. Each ADU forms a source block with K encoding symbols, each of size T. Typically, the number of K is different for each ADU in a sequence of ADUs. Each of the initial K encoding symbol forms the payload of K source packets, whereby each packet may include some of the properties, and includes the source block size K as well as the encoding symbol id (ESI). The size of the object, F, may be carried as part of the source block as shown in Figure 1. In addition to K source packets, N-K repair packets may be sent as part of this ADU. The repair packets would be assigned to the same ADU, for example using a unique Transport Object Identifier (TOI) for ADU.*

**

*Figure 5.7.4-1 Packet Generation for Application Layer FEC*

*At the receiving end, assuming that the code is maximum distance separable (MDS) as the case for RaptorQ or Reed-Solomon codes, i.e. K out of the N packets are sufficient to recover the ADU, the receiver collects K symbols, determines the symbol size T based on the payload size, applies FEC decoding, recovers the source block, reads the size F from the K-th source symbols and recovers the ADU for the next layer in the protocol stack.*

*Such a system is aligned with Forward Error Correction (FEC) Building Block as defined in RFC 5052 [7].*

*The FEC Payload ID (i.e. the information carried in every packet header), essentially only requires carrying the encoding symbol ID and the source block size K. As an example, for RaptorQ as defined in RFC 6330 [8], the maximum source block size is 56403, i.e. 16 bits are sufficient. It is also expected that to signal the ESI, 1 or 2 bytes would be sufficient for most applications. In addition, a TOI may be carried, again using 1 or 2 bytes. While the above FEC configuration only serves as one reference, it may be considered as typical implementation.*

*There are several FEC schemes defined in the IETF that either permit or require that the RTP source packets are sent unmodified or are sent to be compatible with the payload format they comply to. This restriction for example applies to IETF RFC 8627 [11], the Flexible Forward Error Correction (FEC). In a similar fashion, for Raptor FEC in the context of FECFRAME as defined in IETF RFC 6681 [10], for example for the single sequenced flow in clause 8 of IETF RFC 6681 [10], the source packets are unmodified. In these cases, the repair packets need to contain sufficient information to form the source block from a sequence of unmodified source packets. For example, IETF RFC 6681 [10] adds to every repair packet the initial sequence number (of the RTP source packet that is included in the source block), the source block length and the encoding symbol ID, summing up to 48 bit in total. FEC encoding ID, encoding symbol size and maximum source block length are signalled as part of the SDP. The information in the SDP and in the repair packets allows the receiver to add each received source packet to the appropriate ESI in the source block at the receiver. Once decoded, the information in the source block (the size F), can be used to recover the length of the included packet.*

*Based on the information in IETF RFC 6681 [10], for an RTP based delivery for which source packets are unmodified, only the repair packets add the headers. To obtain the source block number a source packet belongs to as well as the source block length of this source block, at least one repair packet needs to be received.*

*An example sender configuration for FEC is provided in Table 5.7.4-1*

*Table 5.7.4-1 Example Content Delivery sender configuration for Application Layer FEC*

|  |
| --- |
| *{*  *"S-Trace": {*  *"source": "S-Trace.csv",*  *"startTime": 0*  *},*  *"FEC": {*  *"symbolSize": "1468",*  *"packet-overhead": "46",*  *"fec-overhead-percent": "30"*  *},*  *"Bitrate": "10000000",*  *"P-Trace": "P-Trace.csv"*  *}* |

*At the receiver, if one or more packets associated to the ADU with are lost, then the timestamp of the loss is the time at which the first lost packet is detected. However, as in-order delivery cannot be assumed, a maximum delay of an ADU needs to be set, typically compared to the render time, after which only received packets are processed as part of the ADU. If at least K packets are received for an ADU within the time budget, the ADU can be fully recovered. If less than K packets are received, the ADU cannot be recovered and one of the following two error handling modes can be configured:*

*- ADU loss: If more than N-K of the packets associated to the ADU are lost, the entire ADU is lost.*

*- Suffix loss: If more than N-K of the packets associated to the ADU are lost, then only the correct prefix preceding the first loss of a source packet is used to generate a partially received ADU.”*

*In addition, a maximum ADU delay is set after which the data in the ADU compared to the render time is no longer helpful and the ADU is discarded.*

*A possible configuration is provided in Table 5.7.2-2. In this case the trace data from an P’-Trace is analysed and all buffer S’-Traces are generated. The loss mode is set to ADU to indicate that any lost fragment results in a loss of the entire ADU. ADUs not recovered within the maxDelay of 60 are considered as lost.*

*Table 5.7.2-2 Example Content Delivery receiver configuration*

|  |
| --- |
| *{*  *"Input": {*  *"Pp-Trace": "Pp-Trace-1.csv",*  *},*  *"Output": [{*  *"Sp-Trace": {*  *"buffer": "left",*  *"Sp-Trace": "Sp-Trace-left.csv",*  *"maxDelay": 60,*  *"lossMode": "ADU"*  *},*  *"Sp-Trace": {*  *"buffer": "right",*  *"Sp-Trace": "Sp-Trace-right.csv"*  *"maxDelay": 60,*  *"lossMode": "ADU"*  *}*  *}]*  *}* |

## 2.3 Discussion

The referenced technical report mentions Raptor(Q) and source packet modification is mentioned but in WebRTC flexFEC is supported that only ads parity (XoR) correction in addition without modification of packets. Also the link to NVidia web-archive does not provide the details on the implementation, but it is assumed that webRTC + flexFEC is used. To develop a guideline for RTC it would be helpful to have a quick summary of the different approaches and the adopted approaches for AL-FEC in RTP according to IETF.

It is recommended to elaborate the documentation text of key issue 3 to be more detailed on the specifics of AL FEC in RTP. A categorization can help to understand how application layer can be integrated smoothly. For example, some schemes require modification of the source packets while others do not. In addition some schemes are resilient to any type of packet losses while others are not. Some schemes are more suitable to implement in low complexity and real time scenario than others. Some add more overhead than others.

## 2.4 Documenting Key Issue number 3

The elaborated text of the key issue should be:

Commercial adoptions may use application layer FEC (AL-FEC) as documented in clause 5.7.4 of TR 26.926. 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: The outcome of this key issue should 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: The outcome of this key issue should be the basis for developing solutions for FEC awareness for PDU Set handling in Key Issue #4.

# Proposal

Discuss the details of key issue number 3 and document it according to the text in clause 2.4 in the technical report for FS\_5G\_RTP\_Ph2.