Improved H.264 /AVC video broadcast /multicast

Dong Tian[*a], Vinod Kumar MV[a], Miska Hannuksela[b], Stephan Wenger[b], Moncef Gabbouj[c]
[a]Tampere International Center for Signal Processing, Tampere, Finland
[b]Nokia Research Center, Tampere, Finland
[c] Tampere University of Technology, Tampere, Finland

ABSTRACT

This paper investigates the transmission of H.264 /AVC video in the 3GPP Multimedia Broadcast /Multicast Streaming service (MBMS). Application-layer forward error correction (FEC) codes are used to combat transmission errors in the radio access network. In this FEC protection scheme, the media RTP stream is organized into source blocks spanning many RTP packets, over which FEC repair packets are generated. This paper proposes a novel method for unequal error protection that is applicable in MBMS. The method reduces the expected tune-in delay when a new user joins into a broadcast. It is based on four steps. First, temporally scalable H.264 /AVC streams are coded including reference and non-reference pictures or sub-sequences. Second, the constituent pictures of a group of pictures (GOP) are grouped according to their temporal scalability layer. Third, the interleaved packetization mode of RFC3984 is used to transmit the groups in ascending order of relevance for decoding. As an example, the non-reference pictures of a GOP are sent earlier than the reference pictures of the GOP. Fourth, each group is considered a source block for FEC coding and the strength of the FEC is selected according to its importance. Simulations show that the proposed method improves the quality of the received video stream and decreases the expected tune-in delay.

Keywords: Video streaming, H.264 /AVC, MBMS, 3GPP, FEC

1. INTRODUCTION

Video coding and transmission have been widely studied in recent decades and a several successful standards have been developed. The latest video coding standard, H.264 /AVC, was jointly developed by the ITU-T and MPEG community and its first version was ratified in 2003[2]. It has proven its superiority over its predecessors in terms of coding efficiency and error resiliency. H.264 /AVC continues to be based on the traditional technique of motion compensation and transform coding of the residual signal. However, a number of advanced features have been added, such as the possible use of multiple reference pictures, and variable block sizes for the motion prediction. Excellent compression efficiency and network-friendliness make H.264 /AVC a competitive candidate for future applications such as 3GPP's Multimedia Broadcast /Multicast Streaming service (MBMS). Due to the anticipated high demands for the video streaming over the mobile network, video streaming based on H.264 /AVC has been one of the focuses in the 3GPP standardization community.

MBMS is a point-to-multipoint service in which data is transmitted from a single source entity to multiple recipients. It has been standardized in 3GPP release 6. A general description of MBMS systems can be found in the technical specification [1]. As depicted in Figure 1, the content delivery of MBMS is conceptually divided into three layers: bearer, delivery method, and user service.

The MBMS bearer defines the architecture to transport data from a single source to multiple receivers. Two delivery methods have been specified: download and streaming. Software update is an example application of the download delivery method, whereas live video is an example using the streaming delivery method. This paper is concerned only with the streaming delivery method.

MBMS uses IP/UDP/RTP transport without underlying guarantied delivery. The packet loss rates perceived at the receiver are highly variable, and depend primarily on the signal quality of the wireless link. This quality is influenced by factors beyond the network's control, such as the physical location and the speed of the receiver relative to the base station. Since the broadcast nature of MBMS does not allow for ARQ-type repair techniques, and since the expected error rates are too high to depend solely on source-coding based tools, a packet-based forward error correction (FEC) scheme has been introduced. Utilizing FEC allows reducing the packet loss rate as perceived by the media decoder to

[*] Email: dong.tian@tut.fi, Phone: +358-40-8282128

zero in virtually all cases. Only at extreme error rates, or when an insufficient FEC strength has been chosen, a packet loss rate above zero may be observable at the media receiver. Assuming FEC at an appropriate strength has the distinct advantage of allowing encoded media without any bits being spent for source-coding based error resilience, which in turn leads to higher coding efficiency and an overall better quality of experience.

In order to be efficient in a highly bursty packet lossy scenario, a FEC block must be as large as possible. Under consideration are FEC block sizes of several dozen packets, which require several seconds for the transmission over the (comparatively slow) links. While efficient from an error recovery point-of-view, such large FEC blocks have negative properties from a user experience point-of-view: since a whole FEC block needs to be received before repair can commence, the tune-in delay is at least as long as the duration of the FEC block – unless FEC repair is not used during the tune-in phase.

This paper proposes a smart delivery order of packets to reduce the tune-in delay and a novel method for unequal error protection to improve error resilience.

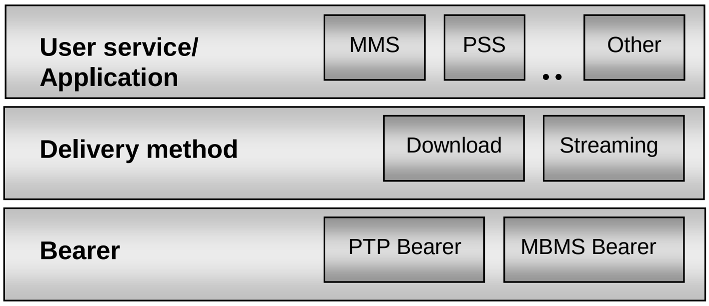

Figure 1. The three layers in MBMS

This paper is organized as follows. Section 2 analyzes the problem with the use of FEC, including the abrupt degradation in quality and the tune-in delay introduced. In section 3 we propose solutions along with examples to protect H.264 /AVC bitstreams unequally and a novel transmission order of coded video data to reduce the tune-in delay. Section 4 provides the details about the simulations. Conclusions are presented in section 5.

2. PROBLEMS

2.1. Background

H.264 /AVC provides a friendly interface between the video coding layer (VCL) and network transmission layer. Each picture is segmented into slices, whereby a slice encompasses an integer number of macroblocks, between one and all macroblocks of a picture. Each slice is encapsulated into one NAL unit which can be considered as the smallest independently decodable unit. All data referring to more than one slice is part of a parameter set, and may be transported by means different from those used to transport slices.

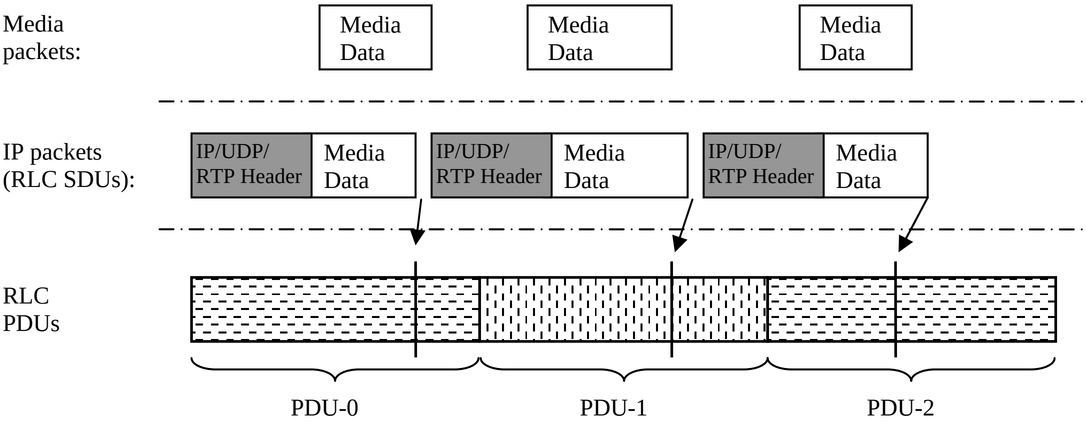

Figure 2. Illustration of the formation of PDUs from media data

The NAL units are encapsulated into RTP packets. RFC3984[5] defines the RTP payload format for H.264 /AVC. It specifies three packetization modes: single NAL unit mode, non-interleaved mode and interleaved mode. The single NAL unit mode and non-interleaved mode are targeted towards conversational systems, in which NAL units are transmitted in NAL unit decoding order, thereby minimizing delay. The interleaved mode is targeted towards systems with relaxed end-to-end latency demands, e.g. broadcast /multicast systems like MBMS. The interleaved mode allows transmission of NAL units out of NAL unit decoding order. A decoding order number (DON) - a field in the RTP payload header or a derived variable - is employed to re-establish the decoding order. In this paper, only the interleaved mode is considered.

RTP packets are transported over UDP/IP. On the sub-IP layers, we assume that one Radio Link Layer (RLC) SDU consists of a single IP packet including its uncompressed header. RLC SDUs are framed and mapped into RLC PDUs (radio data blocks) for the delivery over the MBMS bearer service. It should be noted that the PDU size are constant across the whole transmission session while the SDU sizes are varying due to the variable sized slices in video, and thus the PDUs are not aligned to the SDUs (IP packets) as shown in Figure 2. Loss of a single RLC PDU would cause destruction of all the involved SDUs.

The protocol overhead can be assumed as follows: 12 bytes for RTP, 8 bytes for UDP, and 20 bytes for IPv4 or 40 bytes for IPv6. In addition there is a small overhead per NAL unit due to the use of the sophisticated payload header of RFC3984's interleaved mode. Header compression may be employed to reduce the size of the IP/UDP/RTP headers, but is not further considered here.

In the 3GPP technical specification related to MBMS[1], as illustrated in Figure 3, the FEC is implemented as a meta-payload hierarchically located between RTP and the media payload. The processing can be outlined as follows: an RTP packet, generated by the media encoder, is modified by inserting a FEC payload ID (in the form of a payload header) that indicates the position of the bits of the packet in the to-be-formed FEC block. Furthermore, the RTP payload type is modified so to indicate the presence of the FEC payload ID. The modified RTP packet is sent using the normal RTP mechanisms. In addition, the original RTP packet is also copied into a data structure over which the FEC encoding is run. Once a sufficient amount of data is collected (the FEC block is filled up with variable length RTP packets), the FEC algorithm is applied to calculate a number of repair packets. Those repair packets are also being sent using RTP, and SSRC multiplexing is employed to identify the two different streams.

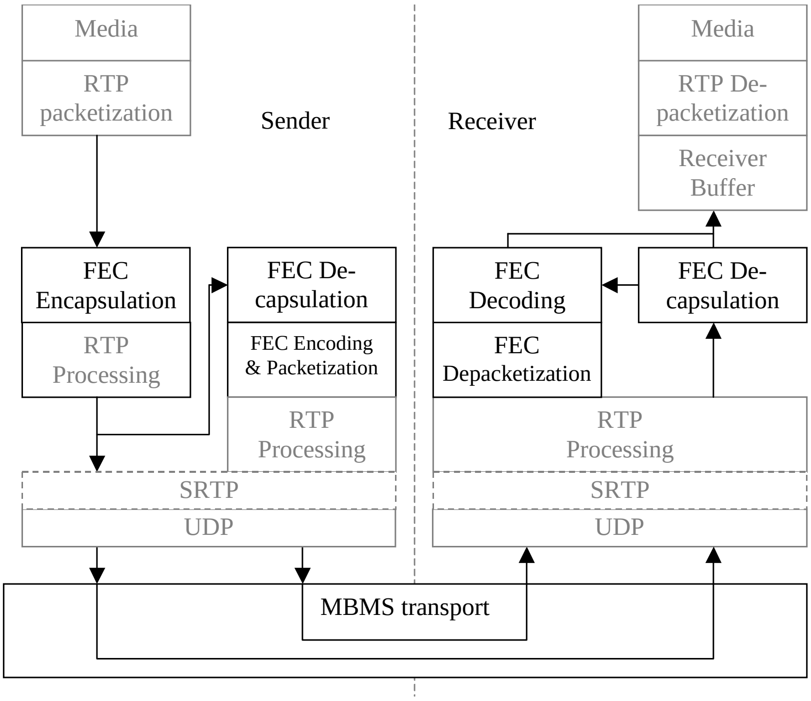

Figure 3. The MBMS system with FEC

At the receiver, first, all media packets and all repair packets belonging to the same FEC block are collected. This is possible through the information of the repair packet payload header and the FEC payload ID. Then, the FEC decoding is applied, resulting in the reconstruction of any missing packets. The special case of insufficient FEC strength, in which not all media packets can be recreated, does obviously not allow to reconstruct missing packets - however, the correctly received media packets are still available and could, after removal of the FEC payload ID, be used for media reconstruction.. The received media packets are transformed into their original state by removing the FEC payload ID and changing the Payload Type to its original value. Finally, the received and the reconstructed media packets are re-sequenced utilizing the RTP sequence number.

2.2. Delay in FEC protected MBMS system

2.2.1. Initial Delay

Hypothetical decoder (HD) models are defined to set up the minimum requirements for the bitstream flows. They are typically composed of a hypothetical buffer and an instantaneous decoder. Such models can be used by the sender to verify the transmitted bitstream does not cause underflow or overflow in the receiver's buffer. A H.264 /AVC HD is defined in Annex C of [2] and a detailed description of an MBMS FEC HD can be found in [7]. In the MBMS system with FEC, since two such HDs are cascaded, additional requirements for the FEC HD are to be met to guarantee the H.264 /AVC HD does not underflow or overflow.

The FEC duration, T_{FEC}, includes not only the actual transmission time of the FEC block, but certain amount of initial buffering delay, D_H, to keep the buffer of media decoder not to underflow or overflow according to the hypothetical decoder models. The value of T_{FEC} may vary from one FEC block to another, and therefore, a variable delay D_C is proposed to be present for each FEC block, allowing for optimization of the initial buffering delay in receivers (see [7] for details). Alternatively, the receiver has to delay the decoding of the media source packets for such a period of the maximum FEC duration, $\max(T_{FEC})$, across the streaming session so as to maximize the probability of correct reception of media samples and to maintain a regular presentation rate of the media samples at the same time [1]. Another factor in the initial delay is the maximum FEC decoding time for a FEC block within the whole streaming session, which is denoted as D_F in Figure 4. The so-called initial delay D_I is therefore can be expressed as,

$D_I = \max(T_{FEC}) + D_F$.

It should be noted that the initial delay is until the decoding of the media source packets and further delay is needed for the rendering of the decoded media samples. For H.264 /AVC, the additional delay for rendering may be signaled in the bitstream, e.g. by the value of num_reorder_frames in the VUI structure or picture timing SEI messages [2]. In this paper we exclude the rendering delay from the consideration.

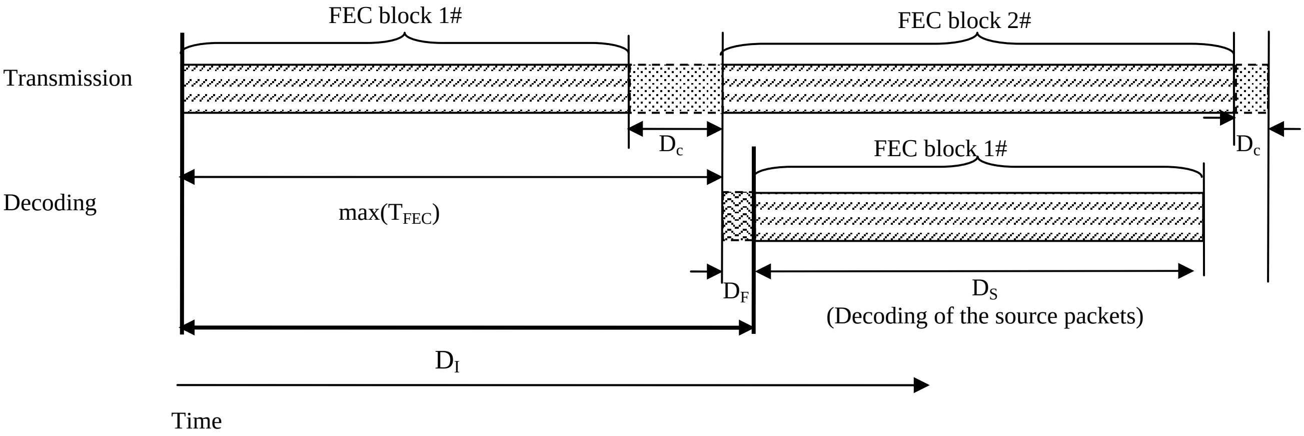

Figure 4. Initial buffering delay

2.2.2. Tune-in Delay

Tune-in delay is defined as the duration from the start reception of packets to the start of correct decoding of packets. It is experienced by a new user who joins the ongoing broadcast, and the tune-in point (the first received packet) is anywhere but at the very start of a FEC block. To successfully tune in, packets representing a random access point (e.g. in the form of an IDR picture, which is assumed henceforth) have to be available. Ideally, all packets after the random access point also have to be available - only this allows for the full user experience. However, even if some packets are missing and the resulting picture degrades, the resulting user experience is perhaps still higher compared to displaying no picture at all.

Frequent random access points are desired to facilitate a shorter tune-in delay and enhanced error resilience, but not wanted regarding to the coding efficiency. As a compromise, a consensus within the 3GPP community is that the FEC block boundaries are aligned at the IDR pictures [8].

Based on the thoughts above, we can imagine two different tune-in strategies. In the first strategy, the receiver first synchronizes to the FEC block structure, i.e. waits for the reception and successful processing of one complete FEC block, before attempting the media decoding. In the second strategy, the receiver searches the media packets as received, disregarding FEC block boundaries, for an random access point. Once found, it starts decoding the random access point and any following pictures regardless of the status of the FEC repair engine. The latter approach obviously allows for a shorter tune-in time, but at the expense of the chance of a seriously degraded picture quality due to losses.

In this paper, we do not expect to rely on the FEC decoder to recover the source packets prior to the tune-in point in the tune-in FEC block. Therefore, the second tune-in strategy is used.

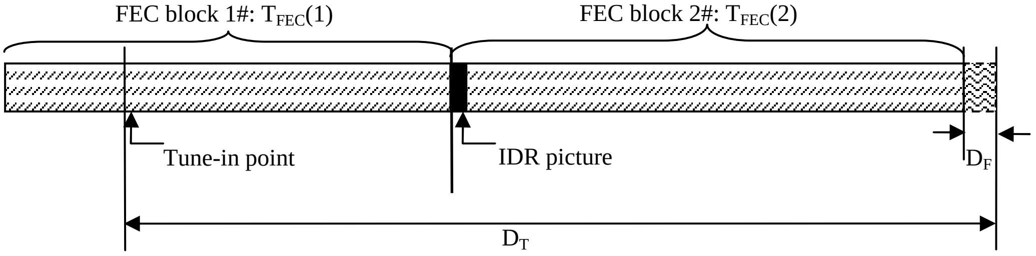

Figure 5. Tune-in delay in MBMS

Conventionally, the packets are sent in the same order of their decoding, in which the IDR picture is sent at the beginning of the FEC block. As a consequence, nothing can be reproduced from the tune-in FEC block and we have to wait until the following FEC block is received. As shown in Figure 5, the delay of D_T can be expressed as,

$$D_T = r * T_{FEC}(1) + T_{FEC}(2) + D_F \tag{1}$$

where r is the percentage of packets being received in the tune-in FEC block, $T_{FEC}(1)$ and $T_{FEC}(2)$ are the durations of the tune-in FEC block and its succeeding FEC block, respectively. Suppose r to be an evenly distributed random variable between 0% and 100%, exclusively. The duration of a FEC block is assumed to be much longer than the hypothetical transmission time of one packet, then r and T_{FEC} can be treated to be statistically independent, and hence we have,

$$E(D_T) = 1.5 * E(T_{FEC}) + D_F, \tag{2}$$

Note that the tune-in delay D_T is not sufficient for a regular decoding rate, but a delay to have a correct decoding of pictures. For a new user, the delay D_P until a regular decoding rate, is called as tune-in period hereinafter,

$$D_P = \max(D_I, D_T). \tag{3}$$

And its expectation is,

$$\begin{aligned} E(D_P) &= \max(E(D_T),\ D_I). \\ &= \max(1.5 * E(T_{FEC}) + D_F,\ \max(T_{FEC}) + D_F) \\ &= \max(1.5 * E(T_{FEC}),\ \max(T_{FEC})) + D_F. \end{aligned} \tag{4}$$

For example, if the T_{FEC} is always of 5 seconds, the expectation of the tune-in delay D_T is then 7.5 seconds, even larger than the initial buffering delay of 5 seconds (with the assumption of an instantaneous FEC decoder, $D_F = 0$). It is irritating for a new user to endure such a long tune-in delay without anything to be decoded and hence to be presented.

In addition it shall be noted that a user may suffer a delay equivalent to the tune-in delay with the FEC synchronization point being lost.

3. GOP STRUCTURE AND PACKET TRANSMISSION ORDER

We propose the combination of two mechanisms to enhance reproduced quality and, simultaneously, reduce the tune-in delay: First, we encode the video utilizing a GOP structure employing sub-sequences and apply unequal error protection for different sub-sequence layers. Second, we remove the requirement of receiving the whole FEC block, by re-ordering the packets in an ascending order of decoding relevance.

3.1. Coding in sub-sequences and unequal error protection

3.1.1. Sub-sequence in H.264 /AVC

Sub-sequences are a form of temporal scalability coding that has been made possible by the introduction of reference picture selection. They were first proposed for H.263+ [6], but have gained popularity only in conjunction with H.264 /AVC. The sub-sequence concept can perhaps best be introduced by an example. Consider Figure 6. A "base layer" and a single "enhancement layer", are depicted. The base layer consist of the IDR picture I0, and the predicted pictures P4 and P8. The IDR picture, by definition, is not predicted from any previous or future picture. P4 is predicted only from I0, and P8 is primarily predicted from P4; however, since H.264 employs reference picture selection on a macroblock level for coding efficiency reasons, some macroblocks may also be predicted from I0 if the encoder determines that this saves bits. This base layer operates with a frame skip of 3 source pictures.

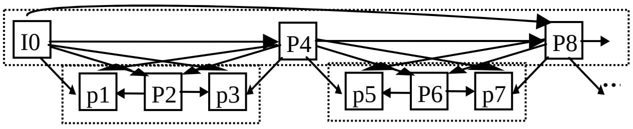

Figure 6. Example of sub-sequences: (*The numbers in the figure indicates the output order*)

Two sub-sequences are further depicted as part of a temporal enhancement layer. In the first of these sub-sequences, picture P2 (note the capital letter P) is predicted only from I0 and P4, but not from p1 and p3. Hence this picture is independent from p1 and p3. However, P2 is used as a reference picture for p1 and p3. Finally, p1 and p3 are predicted from all surrounding pictures including P2. A similar prediction relation consists between p5, P6, and p7. None of the latter pictures have any explicit or implicit prediction relationship with p1, P2, or p3.

Generalized, one can make the following statement: A sub-sequence represents those predictively coded pictures that can be disposed without affecting the decoding of any other sub-sequence in the same sub-sequence layer, or any sub-sequence in any lower sub-sequence layer. The sub-sequence technique enables easy identification of disposable chains of pictures when processing pre-coded bitstreams.

Sub-sequence layers are arranged hierarchically based on their dependency on each other. The base layer (layer 0) is independently decodable. Sub-sequence layer 1 depends on some of the data in layer 0, i.e., correct decoding of all pictures in sub-sequence layer 1 requires decoding of all the previous (in decoding order) pictures in layer 0. In general, correct decoding of sub-sequence layer N requires decoding of layers from 0 to N-1. It is recommended to organize sub-sequences into sub-sequence layers in such a way that discarding of enhanced layers results in a constant or nearly constant picture rate. Picture rate and therefore subjective quality increase along with the number of decoded sub-sequence layers.

While the larger temporal difference between pictures in the base layer does yield a less efficient encoding and hence more bits, it is possible to adjust the QPs of the higher layers to values offering a lower fidelity, very similar to what is commonly done in MPEG-2 B pictures. It was found in [4] that the temporal scalability can be achieved with no significant deterioration in coding efficiency.

3.1.2. Sub-sequence and unequal error protection in MBMS

When employing sub-sequences, the video stream is arranged into so-called super FEC blocks, each of which contains an integer number of consecutive FEC blocks (i.e. source blocks and associated repair packets) in transmission order. All slices in a super FEC block must succeed in decoding order any slice in previous super FEC blocks and must precede in decoding order any slice in succeeding super FEC blocks. In other words, super FEC blocks form self-contained sequences of coded pictures.

To improve the error resilience, we employ unequal error protection (UEP) for different sub-sequence layers. In MBMS, the network conditions as perceived by the individual receiver vary widely. Therefore, the media transmission has to be tailored assuming relatively bad network conditions. It is up to the operator to determine the best operation point according to its business model, i.e. what a failure rate he is willing to accept. Nevertheless, we follow assumptions as used in 3GPP, where a 10% RLC PDU loss rate is considered a worst case for UTRAN streaming [11].

The loss rate, as perceived by the RTP receiver, can be significantly higher than the RLC PDU loss rate, depending on the size and the alignment of the RTP packets to the RLC PDUs. In order to simplify the discussion, we make worst case assumption: each PDU contains parts of three RTP packets - some last bytes of a first RTP packet, a complete second RTP packet, and some bytes of a third RTP packet. Therefore, in a bad case, SDU loss rate is approximately three times of PDU loss rate, i.e. 30%. Let *m* be the number of RTP packets in a FEC block. The expected number of received video packets is *0.7*m*, and the expected number of video packets to be corrected is *0.3*m*. Consequently, a minimum *0.3*m* repair packets should be received for the FEC block. When the same loss rate is applied for repair packets, the transmitted number of repair packets *n* should be such that *0.7*n=0.3*m <=> n=(0.3*m)/0.7 <=> n=(3/7)*m*. To make it an integer number, *n = ceil(3/7*m)*.

In the simulations discussed later, the FEC bitrate for reference pictures is selected as discussed above, so to allow virtually all RTP packets belonging to reference pictures to be repaired. For RTP packets belonging to non-reference pictures we arbitrarily select a FEC strength that is only 1/10th of the FEC strength for reference pictures.

3.2. Transmission order and reduced tune-in delay

The algorithm in this subsection helps to minimize the tune-in delay by avoiding the reception delay of a complete FEC block before beginning decoding. Furthermore, by applying this algorithm, a reduced frame rate may be possible even when only parts of the FEC block are received. We assume here to start decoding as soon as the moment of tune in, without initially relying on FEC-protection. When packetizing pictures in decoding order and assuming a low random access point frequency (e.g. one per FEC block), doing so does not yield meaningful results, since decoding would start somewhere in the middle of a sequence. More than one random access point per FEC block has negative impact on the compression efficiency and is useless except for tune-in, and hence should be avoided.

In order to reduce the tune-in delay, we employ RFC3984’s interleaved packetization mode to place all NAL units belonging to the more important pictures (e.g. the IDR picture, and a few P pictures) towards the end of the FEC block. Even without FEC correction, we assume that in many cases at least the IDR picture (and perhaps a few of the P pictures) can be successfully taken from the packet stream and be reconstructed. This results in having a first visible signal available after a very short tune-in delay (perhaps as short as a few hundred milliseconds). The algorithm, discussed in more detail later, applies to the super FEC block layer and FEC block layer respectively.

3.2.1. Super FEC block layer

In a super FEC block, the media samples are organized into more than one group according to the layers of the prediction hierarchy. Within each layer, any group can be decoded independently from the other groups, as long as the hierarchically higher layers are available.

To reduce tune-in delay, we arrange the groups into FEC blocks in the order of importance for reproduction - the most important groups are placed into FEC blocks that are transmitted last in the super FEC block. For a two-layered system, as discussed before in section 3.1, the super FEC block consists of FEC blocks of two classes: those which carry the base layer information and those which carry the enhancement layer information. In this example, the FEC blocks with the oldest enhancement layer information would be placed first in the super FEC block, followed by newer enhancement layer information, older base layer information, and newer base layer information. The scheme could be easily expanded to more than two layers following the same rationale.

When tuning in to a stream somewhere two thirds in a super FEC block, this would result in the loss of the enhancement layer and some older parts of the base layer. However, the more recent base layer data (in the form of complete FEC blocks, hence including a random access point) would be still available, allowing displaying a video sequence with reduced frame rate.

3.2.2. FEC block layer

In many cases packets from pictures can be ranked beyond static layering according to their relevance for the decoding process. Referring back to subsection 3.1, it should be clear that even in the base layer, the picture I0 is more important than the picture P4 and P8. P4, again, is more important than P8, because P4 is required to reconstruct P8 but not vice versa. The ordering criteria is the inverse of the decoding order - pictures earlier in decoding order are more important than pictures later in decoding order.

Utilizing RFC3984's interleaved mode, it is possible to put data belonging to less important pictures towards the beginning of a FEC block, and pictures with higher importance towards the end of the FEC block.

The FEC repair packets follow the source packets. Let m be the number of media source packet and n be the number of FEC repair packets.

If only some FEC repair packets are received in the tune-in FEC block, the tune-in delay cannot be reduced compared to traditional transmission order. We discuss how the tune-in delay is reduced with at least one source packet available.

As in subsection 2.2, the first presented picture of a FEC super block is expected to be IDR coded in H.264 /AVC, which is sent after packets from all other pictures. Supposing at least the IDR picture is received, the tune-in delay is,

$$D_T = r * T_{FEC} - D_h, \quad (5)$$

Since the decoding of the FEC code is unnecessary for the tune-in FEC block, we need not wait for the reception of the FEC packets and thus shall reduce the corresponding part in the delay of D_H:

$$D_h = n * D_H / (n+m), \quad (6)$$

where *n* is the number of FEC repair packets and *m* is the number of media source packets.

And the expectation of D_T is,

$$E(D_T) = 0.5\ E(T_{FEC}) - E(D_h), \quad (7)$$

with at least 2/3 reduction in the tune-in delay compared to 1.5 $E(T_{FEC})$ in (2). With the reduced tune-in delay, some pictures can be displayed before the regular presentation rate can be achieved and a better user experience can be expected.

In the example with the FEC duration fixed to be 5 seconds, the expected tune-in delay will be reduced from 7.5 seconds to 2.5 seconds. In the first 2.5 seconds, nothing can be presented and in the subsequent 5 seconds, some pictures can be rendered, and finally (7.5 seconds since the tune-in point) the pictures can be presented at the regular rate.

Additionally, for an old user that loses the FEC synchronization point, more pictures can be rescued with the proposed transmission order compared with the conventional transmission order.

3.3. Examples of transmission order

Let's explore some examples to gain an understanding how the tune-in delay is reduced.

3.3.1. Example 1 with single sub-sequence layer

In this first example, the video is coded in IPP with the IDR frequency to be 16. Each picture is coded into one slice. Traditionally, each FEC block would consist of 15 video packets and 5 repair packets, and they are sent in the following order,

… **[** I_0 **][** P_1 **][** P_2 **][** P_3 **][** P_4 **][** P_5 **][** P_6 **][** P_7 **][** P_8 **][** P_9 **][** P_{10} **][** P_{11} **][** P_{12} **][** P_{13} **][** P_{14} **][** FEC **]** …

Where I_0 is an IDR picture, P_x stand for P pictures, **[** FEC **]** stand for the five FEC repair packets. The conventional transmission order will be the same as the coding order.

However, employing our algorithm, the transmission order looks as follows:

… **[** P_{14} **][** P_{13} **][** P_{12} **][** P_{11} **][** P_{10} **][** P_9 **][** P_8 **][** P_7 **][** P_6 **][** P_5 **][** P_4 **][** P_3 **][** P_2 **][** P_1 **][** I_0 **][** FEC **]** …

To simplify the following discussions, let's assume the transmission time for each packet to be constant. As an example, assume the tune-in point to be the 11th packet in the FEC block. In the conventional transmission order, the tune-in point is at the P_{10}. The decoder cannot decode anything meaningful until it reaches the next IDR in the following FEC block; hence the tune-in delay is 10+20=30 packets of (hypothetical) transmission time.

With our modified transmission order, the tune-in point would be at P_4. After re-ordering, the decoder can decode the packets from I_0 to P_4, and the tune-in delay is the (hypothetical) transmission time of 5 packets. During the reception of the subsequent FEC block, 5 video packets can be decoded and displayed. See Figure 7 for the illustration, where the FEC decoding time is ignored.

3.3.2. Example 2 with two sub-sequence layers

The second example shows how sub-sequence coding, UEP and transmission reordering can be designed jointly. Assume that each picture is coded into one slice and the IDR refresh rate is set to 15. Two non-reference pictures (marked as "p") are coded between two successive reference pictures (either IDR picture, marked as "I", or reference inter picture, marked as "P").

Presentation order is,

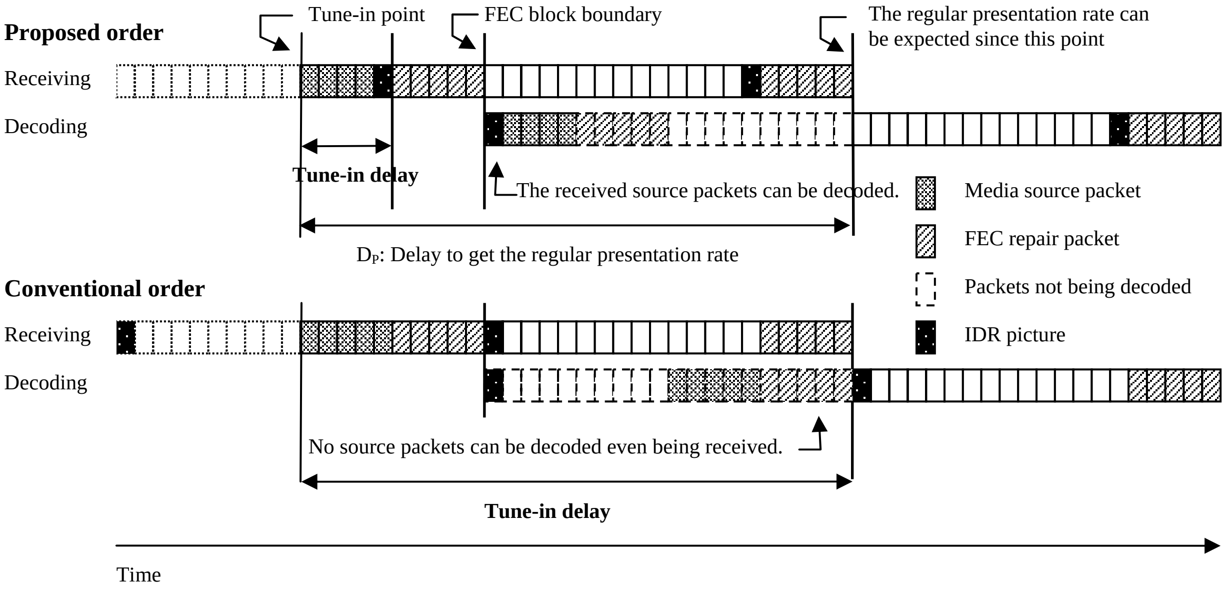
Proposed order
Tune-in point
FEC block boundary
The regular presentation rate can be expected since this point
Receiving
Decoding
Tune-in delay
The received source packets can be decoded.
Media source packet
FEC repair packet
Packets not being decoded
IDR picture
D_P: Delay to get the regular presentation rate
Conventional order
Receiving
Decoding
No source packets can be decoded even being received.
Tune-in delay
Time

Figure 7. How the tune-in delay is reduced in example 1

… [I_0][p_2][P_1][p_4][P_3][p_6][P_5][p_8][P_7][p_{10}][P_9][p_{12}][P_{11}][p_{14}][P_{13}], [I_{15}] … .
Conventional transmission order is,
… [I_0][P_1][p_2][P_3][P_4][P_5][p_6][P_7][p_8][P_9][p_{10}][P_{11}][p_{12}][P_{13}][p_{14}][FEC_1][FEC_2] … .
And the proposed transmission order is,
… [FEC_1][p_2][p_4][p_6][p_8][p_{10}][p_{12}][p_{14}][FEC_2][P_{14}][P_{12}][P_{10}][P_8][P_6][P_4][P_2][I_0], …

"p" stands for a non-ref picture, "P" /"I" stands for a predictive reference picture /IDR picture. In case of all the non-ref pictures are not received, the receiver can still have a 1/2 of the full frame rate. In the worst case, that only I_0 is received, the receiver can display the first picture while waiting for the next complete FEC block during the reception of the subsequent FEC super block, which is better than nothing to be displayed.

For the FEC block of the non-reference pictures, the packets can be transmitted in any order, because every packet does not use any other packet in the block.

4. SIMULATIONS

4.1. Common Simulation Conditions

Simulations were performed following the draft video simulation conditions for 3GPP services[10] as closely as possible. We had to diverge from the condition in [10] in the following areas:

- Due to copyright problems, the Nasa sequence could not be employed. Hence we used the Tour of Glasgow sequence.
- In [10], it is suggested to perform a single simulation run of a 60-second sequence. In order to gather statistically relevant results, we used 50 simulation runs of a 50-sec sequence.

4.2. Video encoding

The picture rate of the Glasgow Tour sequence is originally 12.5 Hz but is considered herein to be 15 Hz (or 30000/2002 Hz, to be exact). Consequently, the duration of the clip is 50 seconds. A constant picture rate of 7.5 Hz was used in all the coded streams. We always code the first picture of a new scene with IDR in H.264 /AVC. There are totally 23 scenes in the Glasgow Tour sequence.

The target bitrate for video and its FEC were calculated by subtracting the FEC bitrate from the channel bitrate. The remaining bitrate was set as the target bitrate for video encoding.

In order to produce the bitstream with the required bitrate, we applied a simple rate control method as follows. A constant quantization parameter (QP) value, herein QP_1, resulting into closest bitrate larger than the target bitrate was first found by trial and error. In order to achieve the target bitrate more accurately, the bitstream was then coded with two QP values, QP_1 for the first pictures in the stream and QP_1+1 for the remaining pictures in the stream. An optimal "change-point picture" (the picture in which the change of QPs happens), resulting into closest bitrate compared to the target bitrate, was searched by trial and error.

Two codec configurations were tested:

1. H.264 /AVC Baseline with constraint_set1_flag = 1. All coded pictures are reference pictures. This codec configuration is referred to as H.264 /AVC IPP hereinafter.
2. H.264 /AVC Baseline with constraint_set1_flag = 1. Every other coded picture is a non-reference picture coded similarly to a B picture in conventional video coding. For more details on the use and benefits of non-reference pictures in H.264 /AVC Baseline, please refer to [4] and [9]. This codec configuration is referred to as H.264 /AVC IpP hereinafter.

The maximum slice size of H.264 /AVC was set to 500 bytes.

4.3. FEC coding

We used Reed-Solomon FEC coding and simple source block generation (one media RTP packet to one column of the source block). To minimize initial buffering delay, source block boundaries were made to match scene boundaries, i.e. the first picture of a source block was an IDR picture. We believe that the results are applicable to more complex Reed-Solomon schemes and other FEC schemes as well.

It was assumed that at the time of encoding, the media encoder and FEC encoder do not have knowledge of the prevailing channel conditions but have to tailor the stream according to expected worst case, i.e. 10% PDU loss rate.

To achieve optimal quality for 10% PDU loss rate, we used a small number of trials and errors and some reasoning as follows:

- We tried suggested 1:3 share of FEC and media bitrate in [11]. However, this FEC code rate turned out to be too low for UTRAN 10% resulting into several dBs of quality drop in average luma PSNR.
- We tried using an adaptive intra macroblock refresh (AIR) algorithm in video encoding without any FEC coding. This resulted into inferior performance compared to coded sequences without AIR, protected with FEC coding.
- We made the reasoning in subsection 3.1.2 to find out the number of FEC repair packets.

For H.264 /AVC IpP codec configuration, non-reference and reference pictures for each group of pictures (from an IDR picture, inclusive, to the next IDR picture, exclusive) were arranged such that non-reference pictures were transmitted earlier than reference pictures. This arrangement minimizes the expected tune-in delay as explained in subsection 3.3.2.

4.4. Packet Loss Simulation

At the time of running the simulations, only the PDU loss patterns under UTRAN for 1% loss rate were available in 3GPP SA4. Therefore, we produced the 10% loss rate pattern ourselves (a random error pattern was used).

We ran 50 simulations of the 50-sec coded stream to get statistically reliable results. A random error pattern starting position was generated for each run, and the same starting positions were used for all codec configurations.

4.5. Decoding

Corrupted SDUs were identified and were discarded in the receiver. FEC decoding is applied to recover missing media packets, whenever possible. An error concealment algorithm similar to TCON (of H.263 Test Model Reference) is applied in both cases.

	Total bitrate (video+FEC) (kbps)	FEC bitrate share (%)	Average luma PSNR (dB)		
			Error Free	PLR 1%	PLR 10%
H.264 (IPP)	44.19	42.6%	27.97	27.97	27.38
H.264 (IpP)	44.18	19.0%	28.43	28.43	27.55

Table 1. Simulation results

4.6. Simulation Results

The simulation results are summarized in the Table 1. It can be seen that the H.264 /AVC IpP codec configuration accompanied by the unequal protection of reference and non-reference pictures improves the performance especially in error-free transmission and packet loss rate (PLR) 1% case. Furthermore, the H.264 /AVC IpP codec configuration enables lower tune-in delay.

5. CONCLUSIONS

In this paper, we studied the video transmission of the H.264 /AVC in MBMS over 3GPP. We proposed a novel method for unequal error protection to achieve better error resilience, which requires scalable coding of H.264/AVC and arranging of transmission order of the pictures based on their importance. We also analyzed the delays in the MBMS streaming and introduced a unconventional transmission order of the packets so to a shorter tune-in delay and a better user experience.

REFERENCES

1. 3GPP TS 26.346 V1.7.0, “Multimedia broadcast /multicast service protocols and codecs (Release 6)”, Feb 2005
2. T. Wiegand, G. Sullivan (editors), “Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC)”, document JVT-G050, 2003
3. Tdoc S4-040671, “Video simulations for MBMS streaming”, Nov 2004
4. D. Tian, M. M. Hannuksela, M. Gabbouj, “Sub-sequence video coding for improved temporal scalability”, ISCAS 2005, Kobe, Japan, May 2005
5. S. Wenger, M. M. Hannuksela, et. al. “RFC3984 - RTP payload format for H.264 video”, Feb 2005
6. S. Wenger, “Temporal scalability using P-pictures for low-latency applications”, Multimedia Signal Processing Workshop 1998
7. Tdoc S4-040672, “FEC buffering for MBMS streaming delivery method”, Nov 2004
8. Tdoc S4-050068, “Media alignment to FEC structures in MBMS streaming”, Feb 2005
9. Tdoc S4-040743, “Reduction of tune-in delay in MBMS streaming”, Nov 2004
10. Tdoc S4-040582, “Draft video simulation conditions for 3GPP services”, Aug 2004
11. Tdoc S4-040348, “Simulation guidelines for the evaluation of FEC methods for MBMS download and streaming services”, May 2004