

Evaluation of Error Resilience Mechanisms for 3G Conversational Video

Jegadish Devadoss, Varun Singh, Jörg Ott
Helsinki University of Technology (TKK)
Espoo, Finland
{jegadish, varun, jo}@netlab.hut.fi

Chenghao Liu¹, Ye-Kui Wang², Igor Curcio²
¹Tampere University of Technology
chenghao.liu@tut.fi,
²Nokia Research Center (Tampere, Finland)
{Ye-Kui.Wang, igor.curcio}@nokia.com

Abstract

Communication in 3G networks may experience packet losses due to transmission errors on the wireless link(s) which may severely impact the quality of conversational and streaming video services, with conversational services being most challenging due to tighter delay requirements. Many error resilience mechanisms have been developed that can be applied at the source (codec) level and transport/application layer to address these challenges. Their respective performance varies depending on the network conditions. This paper analyses and compares the performance of four suitable error resilience mechanisms under different realistic wireless link conditions: selective retransmissions, slice size adaptation, reference picture selection, and unequal error protection using packet-based forward error correction. We derive suggestions for the applicability of the individual mechanisms.

1. Introduction

The third generation mobile system provides a *Multimedia Telephony Service for IMS (MTSI)*, particularly including conversational video [2]. The 3GPP standard supports the use of H.264/AVC [6] for MTSI which uses RTP/RTCP for carrying the audio/video (media) traffic. MTSI typically requires end-to-end delay no more than 300 ms for video to be acceptable. Fading, interference and temporary link outages introduce errors in the wireless 3G links. The conversational video service is highly sensitive to packet loss which may result in either frozen video frames or bad display quality leading to degraded user experience.

The 3G Radio Access Network (3G Link) carries traffic from many different applications and the Radio Link Control (RLC) is used to control the link layer mechanisms depending on the service it is used for. The RLC operates above the Media Access Control (MAC) and can provide services in acknowledged, unacknowledged, and transpar-

ent modes. For MTSI, RLC typically operates in the unacknowledged mode to keep link layer delays to a minimum and enable the applications to implement specific error resilience (ER) schemes which are most suitable. RLC parameters such as payload size, header overhead, etc. and link characteristics (delay, error rates and patterns) need to be considered for choosing appropriate error resilience mechanisms.

For conversational video, the video packets generated by the codec and RTP packetizer are typically larger than the RLC frame size used by MTSI. Consequently, these video packets are fragmented at the RLC layer. With this, the loss of a single RLC frame is equivalent to losing all the RLC frames containing data of the same video packet. Therefore, the RLC frame loss rate translates to a significantly higher value at the IP layer. For example, if the IP packet size is 200 bytes and the video conversation uses a total of 1000 packets. With an RLC payload size of 40 bytes, the number of frames that need to be carried at the RLC layer is $\frac{200 \text{ bytes} \times 1000 \text{ packets}}{40 \text{ bytes}} = 5000 \text{ packets}$. With 0.5% RLC frame loss rate, a mean of 25 frames are lost. Assuming that each of the RLC frames carried contents of a different IP packet (worst case), this yields 25 lost IP packets. The IP layer loss rate is $(\frac{25 \text{ packets}}{1000 \text{ packets}} \times 100) = 2.5 \%$, i.e., five times the RLC layer loss rate.

The combination of end-to-end delay requirements, bandwidth constraints and potentially high packet loss rates require specific error resilience mechanism for MTSI. The H.264/AVC codec inherently supports ER mechanisms for video services. These include Slice Size Adaptation (SSA), Reference Picture Selection (RPS), Adaptive Intra Refresh (AIR), the use of Sub-Sequences, and Flexible Macroblock Ordering (FMO) [7, 19]. This may be augmented by transport/application layer mechanisms such as Selective Retransmission (NACK), the use of FEC, among others. The performance of the available ER mechanisms vary with the observed end-to-end delay, link loss, bandwidth constraints, and call scenarios (e.g., UMTS Terrestrial Radio Access Network (UTRAN) to UTRAN, UTRAN to Wireless LAN

(WLAN), wireless to fixed). The aforementioned ER mechanisms have their strengths and weaknesses and none is expected to fit all operating environments. This calls for a study on the applicability of the different ER mechanisms. In this paper, we choose a typical operating environment and evaluate the performance of four promising ER mechanisms. Based on our evaluation, we describe the applicability of the error resilient mechanisms as a function of the characteristics of the operating environment.

The paper is structured as follows: In section 2, we introduce the background of the discussed ER mechanisms and the related work in this area. Section 3 explains the features and configurations of the simulation environment based upon which we evaluate the applicability of the ER mechanisms using the simulation results in section 4. Section 5 concludes this paper with a short summary.

2. Overview Of Error Resilience Mechanisms

The goal of error control methods is to achieve minimum end-to-end distortion under a certain channel condition, by recovering the lost or corrupted video data, reducing or stopping error propagation, and/or masking the damage effect. To overcome loss due to transmission errors, various error control methods have been developed [7, 15, 14, 17]. These methods can be classified into the following three categories: source coding methods, channel coding level methods, and end-to-end transport/application methods. Source coding level methods refer to those methods that are carried out by the video codec (encoder and decoder). Channel coding level methods refer to those methods that operate at the physical and link layers.¹ Transport layer methods refer to end-to-end mechanisms which are codec-independent, application layer methods to those integrating source coding and transport mechanisms.

Source coding level error control methods include error resilient encoding, interactive error control and error concealment. Error resilient encoding injects redundancy into the bitstream. This redundancy may be used to detect data losses, reduce/stop error propagation, and/or assist error concealment. In interactive error control, wherein the encoder and decoder work cooperatively, the encoder utilizes feedback information from the decoder, e.g. correctness of pictures or parts thereof, to adjust the encoding strategy adaptively. For example, the feedback information can be employed in Reference Picture Selection (RPS) [14] to make the current encoded picture select correctly decoded older reference pictures for inter-picture prediction. Thus, error propagation due to corrupted reference pictures can be stopped. With—non-interactive—error concealment mechanisms, the decoder attempts to mitigate the loss impact al-

¹Since our aim is to be independent of specific operator infrastructure, we do not consider channel coding methods further in this paper.

gorithmically from adjacent regions of a frame and/or other frames.

Transport layer methods include retransmissions, Forward Error Correction (FEC), and interleaving [11, 8, 14]. With retransmissions, the receiver communicates the loss of packets to the sender (using Negative Acknowledgements, NACK, but also Automatic Repeat reQuest, ARQ) and the sender responds by retransmitting the packet. Its design leads to an increase in end-to-end delay. FEC adds repair bits to the transmitted packets to correct bit errors (particularly when used in RLC layer) or additional packets to repair packet losses. Both can be used to recover lost or corrupted data. Interleaving, in contrast, may limit the impact of burst losses, but cannot recover lost data.

Application layer methods include Unequal Error Protection (UEP) [14, 8] and robust scheduling [3]. UEP uses FEC [8, 19] or different video encoding [16] to selectively provide better protection to more important parts of the media stream. In robust scheduling, more important picture data is sent earlier than less important picture data, such that even during abrupt network throughput changes, such as cell handovers, a smooth playback, with possibly lower quality or frame rate, can be achieved.

In this paper, we choose four ER mechanisms with different characteristics: NACK-based retransmissions as generic transport mechanism to cope with losses, slice size adaptation (SSA) as a transport-aware source coding mechanism without explicit feedback to minimize the impact of losses, Reference Picture Selection (RPS) as a source-coding mechanism with feedback to limit error propagation, and Unequal error protection (UEP) as an application layer approach without feedback.

2.1. Retransmitting Lost Packets (NACK)

NACK-based feedback may contain generic (transport) or (codec) payload-specific information [10]. Generic NACKs simply refer to RTP sequence numbers of lost packets, whereas payload-specific ones convey feedback information from the receiver to the sender (see section 2.3). Reports are collected at the receiver for a short interval and then sent to the sender. The collection interval period is regulated by the timing rules defined in RFC 4585 [10] which provides mechanisms allowing feedback messages to be sent as early as possible while still adhering to the RTCP bandwidth constraints. For generic NACKs, the packets reported lost are retransmitted by the sender.

2.2. Slice Size Adaptation (SSA)

This mechanism modifies the slice-sizes of encoded pictures based on the channel characteristics: as extremes, when the channel is lossless, there can be one full picture

per slice (up to path MTU size) and when the channel experiences high losses, slices can be as small as the RLC layer payload. Larger slices improve encoding efficiency, but are more vulnerable to RLC frame losses because of RLC fragmentation. Figure 1 shows the variation of average PSNR with respect to different slice-sizes in varying loss scenarios: it can be seen that there is direct correlation between packet loss and slice size.

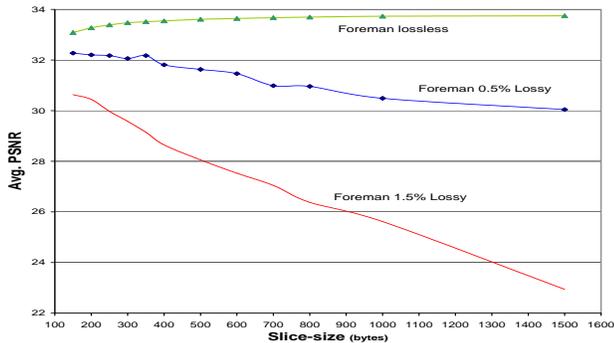


Figure 1. Effect of slice size on PSNR under different RLC frame loss conditions

2.3. Reference Picture Selection (RPS)

The video sender and receiver cooperate to provide video error resilience: The receiver (decoder) detects the loss of a slice (or picture) and sends a feedback message carrying information about either missing (NACK mode) or correctly decoded (past) pictures (ACK mode) available at the decoder. Using this information, the sender (encoder) can choose one of these—previously buffered—pictures as a reference picture for subsequent inter-picture prediction encoding. This method stops the temporal error propagation caused by an earlier packet loss. In both modes, the encoder is able to retrieve the required picture loss data. The mode of operation can be decided depending on the observed packet loss rate to minimize the feedback overhead.

Feedback messages can be encapsulated according to RFC 4585 [10] and ITU-T H.271 [5] and are carried in RTCP reports. Similar to the NACK-based retransmissions, the RTCP reports are guided by the timing and messaging rules defined in RFC 4585 [10] and RFC 5104 [18].

2.4. Unequal Error Protection (UEP)

Unequal Error Protection (UEP). Forward Error Correction (FEC) is an ER mechanism that is applicable in scenarios where extra bandwidth is available and the cost of retransmission is high. However, in application scenarios like

MTSI, the wireless channel capacity is scarce and expensive so that using extra bandwidth for FEC is problematic. Unequal Error Protection (UEP) tries to strike a balance by protecting only a chosen set of the media packets.

H.264 supports flexible temporal scalability by means of sub-sequences and sub-sequence layers [4]. The simplest form is to use reference pictures and non-references as in conventional video coding standards. The reference pictures are used in the future inter-picture prediction chain and, therefore, have a higher significance than non-reference pictures. A loss of a reference picture may reduce the decoded video quality due to error propagation, as subsequent pictures may refer to the lost reference picture. Losing non-reference pictures does not affect the prediction chain and its impact will thus be limited.

Subsequence encoding offers an opportunity for using FEC. As the reference pictures are more important, FEC can be applied only to them, yielding a type of UEP mechanism. This approach reduces the FEC overhead to an acceptable level. In this paper, RFC 2733-based FEC [12] is used to protect the reference pictures in the encoded video.

3. Simulation Environment

This section focuses on the characteristics and configurations of the simulation environment for evaluating the error resilience mechanisms described above. At its core, it includes ns2² as a network simulator with the extensions described in the following.

The simulator needs to interface with a video codec, so that adaptive encoding mechanisms can be evaluated. The decoder is extended to support the generation of feedback messages. Similarly, the encoder is extended to react to the received feedback messages. Since the codec and the simulator need to interact, both of them need to synchronise their clock. For this purpose, we have implemented an interface named “REAL” (figure 2). It is used to exchange control and data packets over a TCP/IP connection between the codec and the simulator.

Figure 2 depicts an overview of the simulator system. The RTP Traffic Generator module is responsible for timing the RTP packet injection into the network and for sending/receiving media packets to/from the codec. The RTP Agent module is responsible for generating RTCP reports, also implementing the timing rules defined in RFC 4585 [10]. The link characteristics are incorporated in a module called 3GLink. The 3GLink module takes care of fragmenting packets into RLC frames, reassembling the received ones into IP level packets. It also introduces link losses and delays as specified. The 3G Link implementation exhibits the same behaviour as the 3GPP Simulator [1].

²http://nslam.isi.edu/nslam/index.php/Main_Page

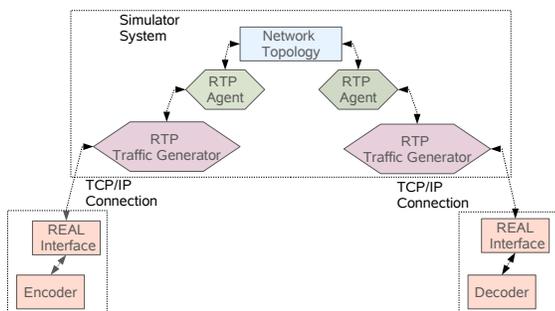


Figure 2. Simulator System Overview

3.1. Network Setup

The 3G link parameters are set as shown in table 1 which is typical for the 3G conversational video service.

Table 1. 3G Link Configuration

Link Bandwidth	128 Kbps
Link Delay	100ms
RLC Payload Length	40 bytes
RLC Header Size	1 byte
Compressed RTP/UDP/IP Header Size	3 byte
PDCP Header Size	1 byte

Table 2 shows an example for RLC framing rules: Assume an RTP packet size of 200 bytes (constant for simplicity). The size of the packet including the UDP and IP header is 228 bytes. At the RLC layer, the RTP/UDP/IP header is removed and replaced with a compressed header of 3 bytes. The PDCP header is added to every media packet. Hence, the effective size of a media packet before fragmentation is $228 - 40 + 3 + 1 = 192$ bytes.

If a particular RLC frame (PDU) is lost, IP packet re-assembly cannot succeed for all those packets of which fragments were contained in the respective frame. Hence, one or more media packets will be lost. For example, if PDU 5 in table 2 is lost or corrupted then both the first and second media packet are lost.

The network topology used for the simulation is depicted in figure 3. In the call scenario under evaluation, both the caller and callee use 3G as the access link. The 3G Core is considered as a well-provisioned reliable private network, no additional errors are introduced inside the core.

3.2. Codec Setup

We use the Nokia H.264 codec [9]. Three well-known media sequences (Foreman, Football and News) are used to study the error resilience mechanisms. The sequences

Table 2. An example of link level framing rules

PDU 1	1 (RLC Header) + 40 (1st packet, 152 bytes left)
PDU 2	1 (RLC Header) + 40 (1, 112)
PDU 3	1 (RLC Header) + 40 (1, 72)
PDU 4	1 (RLC Header) + 40 (1, 32)
PDU 5	1 (RLC Header) + 32 (1, 0) + 8 (2, 184)
PDU 6	1 (RLC Header) + 40 (2, 144)
...	...

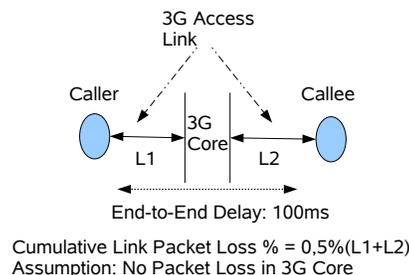


Figure 3. Network Topology

Football, Foreman and News have high, medium and low motion respectively. All the sequences are encoded at 15 frames per second. The slice size is kept at 200 bytes for all evaluated ER mechanisms except for SSA. The bit rates of the encoded sequences vary depending on the type of error resilience mechanisms. In our experiments, the channel bandwidth of the conversational video is taken as 128 kbit/s. Since a part of the channel bandwidth is consumed by RTCP and RLC layer overheads, the maximum media encoding rate is set to 115 kbit/s. For NACK, it is further reduced to occasionally accommodate the overhead for re-transmitting lost packets. In the UEP case, the sequence is encoded at a lower rate considering the 23% overhead that the use of FEC introduces (determined experimentally). For all the experiments, long video sequences of approximately 4000 frames is chosen so as to have randomness in the packet loss events. This yields the following net encoding bit rates for the sequences for the various error resilience mechanisms: 110 kbit/s for NACK, 115 kbit/s for SSA and RPS, and 92 kbit/s for UEP.

4. Evaluation

4.1. Selective Retransmission (NACK)

In this approach, lost packets are reported to the encoder by the decoder using feedback messages. The feedback messages are carried in RTCP reports. The RTCP reports

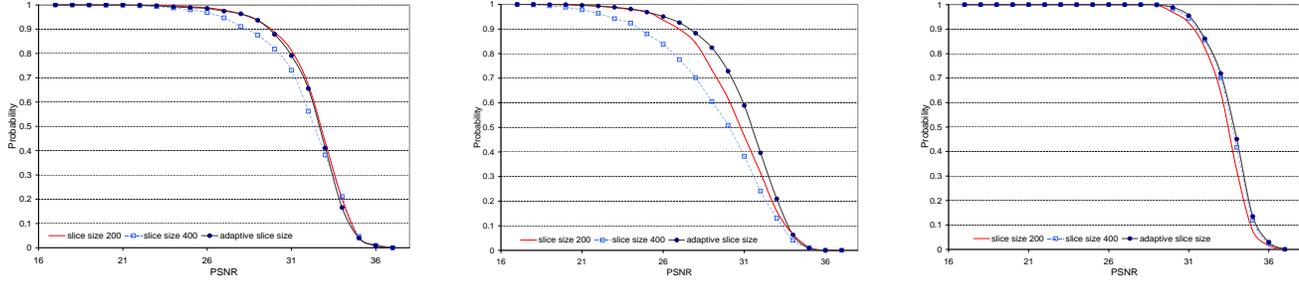


Figure 4. CCDF of PSNR for a) 0.5% RLC frame error rate (0 to 900 frames), b) 1.5% RLC frame error rate (901 to 1950 and 2801 to 3600 frames), and c) error-free link (1951 to 2800 frames)

use 1% of the media bandwidth. This decision is made to use as much as channel bandwidth for the actual video. The 1% RTCP bandwidth roughly translates to about one report being sent every second.

With retransmissions, the delay experienced by a retransmitted packet would be at least three times the one-way delay. For link delays greater than one third of the acceptable end-to-end delay, this method can be effectively ruled out (e.g., which puts an upper limit of 100 ms on the link delay to achieve an acceptable end-to-end delay of 300 ms). As there are call scenarios where the one-way delay can be less than 100 ms., it is still useful to analyse the effectiveness of NACK based approach for low delay scenarios. We have performed simulations using one-way delay of 60 ms.

The tables 3 and 4 summarize the effectiveness of the NACK mechanism. The traces of the Foreman sequence show that close to one third of the lost packets can be recovered for an end-to-end delay of 60ms within the time bound of 300 ms. For Football and News, the recovered packets are approximately 17% and 16%, respectively. This shows that NACK is an effective mechanism for low end-to-end delay scenarios.

Table 3. Summary of NACK effectiveness

	Foreman	Football	News
Number of packets lost	593	689	695
Initial loss rate	2.2123	2.595	2.259
Packets retransmitted	223	237	231
Final error rate	1.479	2.143	1.898

Table 4. Delay distribution of retrans. pkts

	Foreman	Football	News
less than 200ms	16.10%	4.20%	0.40%
200ms to 300ms	64.60%	46.40%	47.60%
greater than 300ms	19.30%	49.40%	51.90%
Efficiency (C)	30.35	17.41	15.954

$$C = \frac{\text{RetransmittedPktsThatReachedWithin300ms}}{\text{TotalLostPackets}}$$

4.2. Slice Size Adaptation (SSA)

For conversational video service in 3G, the typical RLC payload length is 40 bytes. Media packets are fragmented to fit in the RLC layer payload. This directly implies that the larger the media packet are, the larger the perceived loss rate at the IP layer gets. At the codec level, use of bigger slice size reduces slice header overhead and also improves the efficiency of video coding. In the following, we investigate the impact of the tradeoff between the slice size and the packet loss probability and its impact on video quality.

The performance evaluation is performed for three different cases: fixed slice sizes of 200 and 400 bytes and adapting the slice size depending on the observed packet loss. For the adaptive slice size mechanism, the slice size is dynamically adjusted based on the calculated average loss rate. The average loss rate is a sliding window average of the previous three loss rates received in normal RTCP reports. The slice size is doubled when the average packet loss rate goes below 1.0% and the maximum slice size value that can be reached is kept as the MTU size (1500 bytes). Slice sizes remain constant for loss rates below 2.5% to provide stability to the system. However, in high loss scenarios, if the slice sizes are larger than 400 bytes then it is halved and for sizes below 400 bytes, it is reduced in steps of 50 bytes up to a minimum of 150 bytes. An example of the adaptive operation of the algorithm is shown in figure 5.

For evaluating the performance, a time varying loss pattern is used. In the loss pattern used, the first 900 frames experiences 0.5% RLC layer frame loss, the next 1050 frames experiences 1.5% RLC layer frame loss, the next 850 frames experiences no frame loss and the last 800 frames again experiences 1.5% RLC layer frame loss. The observed performance for Foreman sequence is plotted in Figure 4 (a-c). RTCP reporting for SSA is based on the timing rules defined in RFC 3550 ($5s \pm 50\%$) [13]. Figure 5 shows the variation of slice size with the instantaneous

Table 5. Comparison of PSNR for fixed and adaptive slice size in each scenario

	200 bytes	400 bytes	Adaptive
Avg. Total PSNR	31.55	30.78	31.62
0.5 % loss	32.40	31.91	32.30
1.5 % loss	30.46	29.40	30.99
no loss	33.25	33.54	33.66

loss rate and the average loss rate.

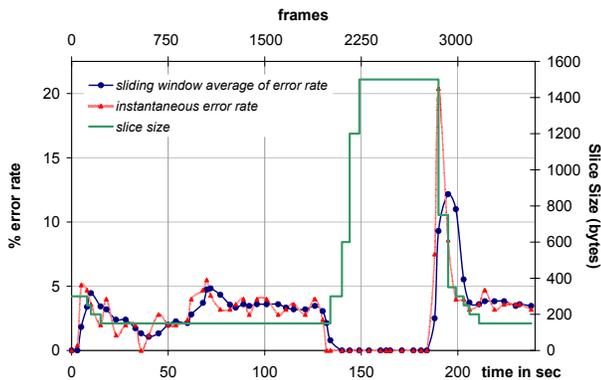


Figure 5. SSA based on loss rate

In lossy scenario (figure 4a and b), the slice size 400 performs badly when compared with the slice size of 200 and adaptive (adaptive mechanism can reach a slice size as low as 150 bytes). But in no loss scenario (figure 4c), the slice size 200 performs badly when compared with the slice size of 400 and adaptive (adaptive mechanism can reach a slice size as high as 1500 bytes). These evaluations show the effectiveness of SSA as an ER mechanism and are also supported by the PSNR values given in table 5.

In heterogeneous call scenarios (with one party on a 3G Link and the other one in the wired public Internet or a WLAN), the applicability of slice size adaptation gains significance. A media packet of 200 bytes adds 40 bytes (RTP/UDP/IP) of overhead when no header compression is used. Many of the available SIP-based VoIP products do not support header compression, nor do WLAN installations. This introduces significant overhead (16.67% for a slice size of 200 bytes) when small slice sizes are used. Furthermore, in a heterogeneous call scenario, it is not possible to identify whether the packet loss occurs in 3G link or in public Internet; WLANs suffer from small packet sizes; and the public Internet introduces different loss patterns. These mixed scenarios are for further study.

4.3. Reference Picture Selection (RPS)

With the RPS mechanism, the feedback messages are generated by the receiver to indicate in case of losses to the sender which pictures have been decoded correctly/incorrectly. The feedback messages are encapsulated in RTCP packet and are reported to the sender. The RTCP reports are sent every 250 ms and consume approximately 2% of the media bandwidth.

Table 6 shows the average PSNR of the RPS compared to an unprotected sequence (serving as a reference) in an error free scenario.³ On an error free link, RPS achieves better PSNR because the encoder uses the best matching latest reference frame in displaying order as reference frame. Table 7 presents average PSNR of the different error resilience schemes after undergoing link loss of 0.5%.

To evaluate the error resilience performance of RPS at frame level, we also present the PSNR of each frame after undergoing a link loss of 0.5% in figure 6 (a-c). In all the test sequences, the PSNR drops down during the interval when a lost frame has been referenced. The PSNR increases immediately after the encoder chooses a correct reference picture. Many of the high PSNR spikes are reached after the encoder chooses a correctly received frame as reference.

To observe how quickly RPS stops error propagation, the error propagation period for each incorrectly received picture are presented in figure 6 (d-f). The width of each spike in the figure, is proportional to the number of pictures that reference an incorrect picture. The width of the spike is also coded in the y-axis, indicating the intensity of the error propagation period. The error propagation period counts the error propagated pictures from first picture that uses the incorrect picture as reference until the first picture is received using the new (correct) reference at the decoder. Then, the error propagation period is set to zero. When the one way delay is 100 ms and the frame rate is 15 frames per second, error propagation is stopped by RPS in about four to seven pictures.

Table 6. Maximum Achievable PSNR

	Foreman	Football	News
UnProtected	33.30	28.80	37.70
RPS	35.54	29.48	40.15
UEP	31.12	27.42	36.12

4.4. Use of FEC to Protect Reference Frames

The use of FEC to protect reference pictures for the sequences Foreman, Football and News consumes close to

³UEP, being another way to limit error propagation, is included as another data point which will be discussed in the next subsection.

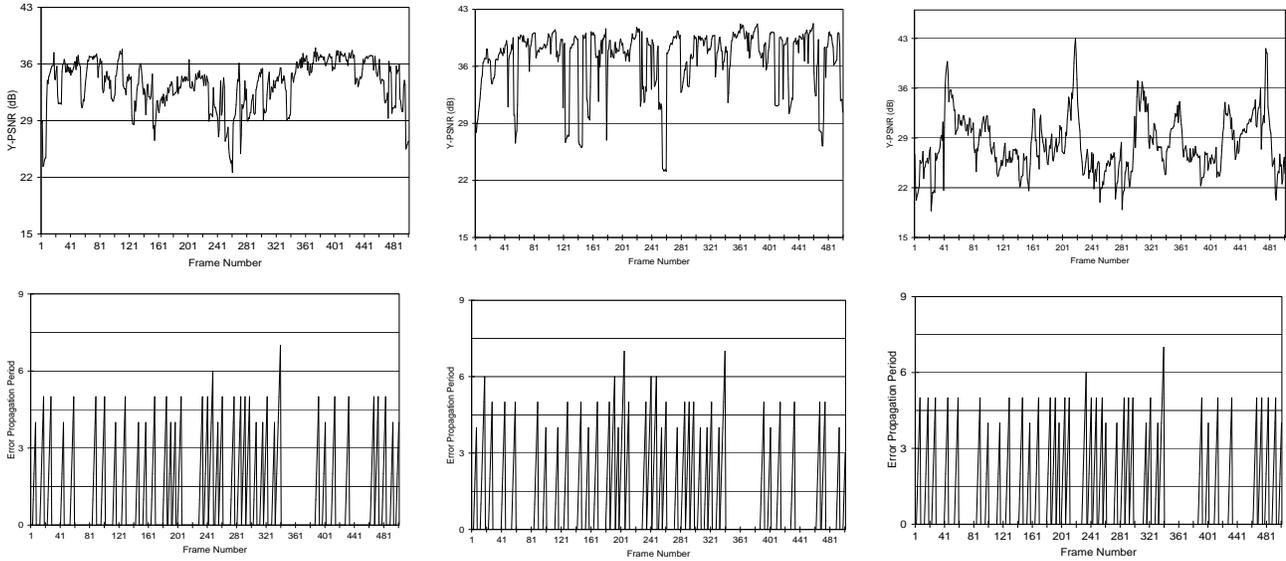


Figure 6. Row 1: PSNR Variation due to RPS for Foreman (a), News (b), and Football (c) sequences. Row 2: Error Propagation Period after using RPS for Foreman (d), News (e) and Football (f).

Table 7. PSNR values after Simulation

	PSNR After Simulation		
	Foreman	Football	News
UnProtected	32.15	28.03	35.39
RPS	33.68	28.05	37.37
UEP	28.32	26.86	34.47
	% Drop in PSNR		
	Foreman	Football	News
UnProtected	3.47	2.66	6.14
RPS	5.23	4.85	6.92
UEP	9.00	2.05	4.57

Table 8. Corrective Performance due to FEC

	A	B	C
Foreman	1.99	1.56	21.60
Football	2.02	1.58	22.04
News	1.79	1.36	23.81

23% overhead. This overhead needs to be compensated by reducing the media bit rate in the encoder. This significantly contributes to the reduced maximum obtainable PSNR for UEP, observed in table 6.

Table 7 shows the PSNR with an RLC frame loss rate of 0.5%. The performance of the unprotected case and RPS is better than when using UEP. However, one interesting observation is that the quality drop in PSNR is less in UEP for two out of three sequences. Table 8 shows that, by using FEC, close to 21–24% of the lost packets can be recovered.

$$A = \% \text{ of packets lost (including FEC packets)}$$

$$B = \% \text{ of packets lost after FEC correction}$$

$$C = \frac{A-B}{A} \times 100$$

From the above results, it is clearly visible that UEP does not even perform as good as the unprotected case, for the operating scenario under evaluation. But, UEP is still ap-

plicable for environments that observe high packet loss and high link delay. The paper on the use of subsequences and FEC for streaming media also supports this argument [19].

5. Conclusion

In this paper, we have evaluated four error resilience mechanisms for MTSI conversational video services in a specific call scenario. Our results presented above show the applicability of the ER mechanisms over 3G Links. The applicability of the ER mechanisms discussed here, can be depicted as a function of observed packet loss and end-to-end delay. For the chosen call scenario, the applicability of the discussed error resilience mechanisms is shown in figure 7.

Summarizing the evaluation, NACK is applicable only for low end-to-end delay scenarios. Even in the low end-to-end delay environment, at high observed packet losses, NACK is not applicable as it leads to spending more bits in retransmission. SSA becomes applicable at high packet loss and in heterogeneous operating environment. RPS performs well in our chosen environment. Its effectiveness is somewhat proportional to both the packet loss and end-to-

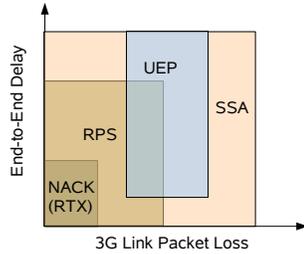


Figure 7. Applicability of ER Mechanisms

end delay. The higher the packet loss and end-to-end delay, the more time it takes to stop the temporal error propagation. UEP is not effective for the operating environment chosen here for evaluation. But we expect its relevance to increase in high delay scenarios, where the cost of repair using interactive mechanisms is high.

The analysis also shows that there can be more than one error resilience scheme applicable at a particular operating environment. We also observe the need for adaptive error resilience system that can choose from a set of ER mechanisms depending on the observed error conditions. As briefly pointed out above, future work will need to take into account heterogenous scenarios to ultimately derive adaptive error resilience mechanisms suitable for the entire wired and wireless Internet.

References

- [1] 3GPP. Software Simulator for MBMS Streaming over UTRAN and GERAN. Proposal S4-050560, 3rd Generation Partnership Project (3GPP), Sept. 2005.
- [2] 3GPP TS 26.114. IP Multimedia Subsystem (IMS): Multimedia Telephony; Media handling and interaction.
- [3] P. Chou and Z. Miao. Rate-distortion optimized streaming of packetized media, 2001.
- [4] M. H. D. Tian and M. Gabbouj. Sub-sequence video coding for improved temporal scalability. In *Proc. of IEEE Int. Symposium on Circuits and Systems*, May 2005.
- [5] I.-T. R. H.271. Video back channel messages for conveyance of status information and requests from a video receiver to a video sender, May 2006.
- [6] ITU-T Rec. H.264. Advanced video coding for generic audiovisual services.
- [7] S. Kumar, L. Xu, M. K. Mandal, and S. Panchanathan. Error resiliency schemes in H.264/AVC standard. *Journal of Visual Communication Image Representation*, 17:425–450, April 2006.
- [8] A. Li. RTP Payload Format for Generic Forward Error Correction. RFC 5109 (Proposed Standard), Dec. 2007.
- [9] Nokia. H.264/AVC codec donation - Licensing Statement. Information S5-050673, 3rd Generation Partnership Project (3GPP), Sept. 2005.
- [10] J. Ott, S. Wenger, N. Sato, C. Burmeister, and J. Rey. Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF). RFC 4585 (Proposed Standard), July 2006.
- [11] C. Perkins and O. Hodson. Options for Repair of Streaming Media. RFC 2354 (Informational), June 1998.
- [12] J. Rosenberg and H. Schulzrinne. An RTP Payload Format for Generic Forward Error Correction. RFC 2733 (Proposed Standard), Dec. 1999. Obsoleted by RFC 5109.
- [13] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 3550 (Standard), July 2003.
- [14] Y. Wang, S. Wenger, J. Wen, and A. Katsaggelos. Review of error resilient coding techniques for real-time video communications, 2000.
- [15] Y. Wang and Q. Zhu. Error control and concealment for video communication: A review, 1998.
- [16] Y.-K. Wang, M. Hannuksela, and M. Gabbouj. Error resilient video coding using unequally protected key pictures. *2003 International Workshop on Very Low Bitrate Video (VLBV 2003)*, pages 290–297, Sept. 2003. Madrid, Spain.
- [17] S. Wenger. H.264/AVC over IP. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(7):645–656, 2003.
- [18] S. Wenger, U. Chandra, M. Westerlund, and B. Burman. Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF). RFC 5104 (Proposed Standard), Feb. 2008.
- [19] S. Wenger and J. Ott. A joint source/channel coding approach for low-delay video. *submitted for publication*, January 2008.