Evaluation of Error Resilience Mechanisms for 3G Conversational Video

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

Chenghao Liu, Ye-Kui Wang, Igor Curcio
Nokia Research Center (Tampere, Finland)
{ext-chenghao.liu,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 video services, with conversational video being most challenging to repair due to tighter delay constraints. 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 analyzes and compares the performance of four 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 [3]. The 3GPP standard supports the use of H.264/AVC [7] 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 transparent modes. For MTSI, the RLC typically operates in the unacknowledged mode to keep link layer delays to a minimum and enable the applications to implement their own most suitable error resilience (ER) schemes. RLC parameters (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, so that these 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 higher IP packet 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) [8, 20]. This may be augmented by transport and application layer mechanisms such as Selective Retransmission (NACK) or the use of FEC, among others. The performance of the ER mechanisms varies 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 above mechanisms have their strengths and weaknesses and none is expected to fit all operating environments. This calls for an analysis of 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. We introduce the background of the discussed ER mechanisms and the related work in this area in section 2 and explain the features and configuration of our simulation environment in section 3. Section 4 presents our simulation results and draw conclusions on the applicability of the error resilience mechanisms as a function of the characteristics of the operating environment in section 5 where we also discuss directions for future work.

2. Overview Of Error Resilience Mechanisms

The goal of error control methods is to achieve minimum end-to-end distortion under certain channel conditions, 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 [8, 16, 15, 18]. 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 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, the encoder and decoder work cooperatively and the encoder utilizes feedback information from the decoder, e.g., correctness of pictures or parts thereof, to adjust the encoding strategy adaptively. For example, feedback information can be employed in Reference Picture Selection (RPS) [15] to make the current picture select a correctly decoded older reference picture 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 algorithmically from adjacent regions of a frame and/or other frames.

Transport layer methods include retransmissions, Forward Error Correction (FEC), and interleaving [12, 9, 15]. For 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. This 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) [15, 9] and robust scheduling [4]. UEP uses FEC [9, 20] or different video encoding [17] to selectively provide better protection to more important parts of the video bitstream. 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.

For 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.

Retransmissions (NACK). NACK-based feedback may contain generic (transport) or (codec) payload-specific information [11]. 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 RPS below). Reports are collected at the receiver for a short interval and then sent to the sender. The collection period is regulated by the timing rules defined in RFC 4585 [11] 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.

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 due to RLC fragmentation.

Reference Picture Selection (RPS). 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. This method stops the temporal error propagation caused by an earlier packet loss. The mode of operation can be chosen based on the observed packet loss rate to minimize the feedback overhead.

Feedback messages can be encapsulated according to RFC 4585 [11] and ITU-T H.271 [6] 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 [11] and RFC 5104 [19].

Unequal Error Protection (UEP). Forward Error Correction (FEC) is an ER mechanism that is applicable in sce-

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

narios 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 [5]. 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 [13] is used to protect the reference pictures in the encoded video.

3. Simulation Environment

Our simulation environment is based upon ns2 [1] as the core network simulator with the extensions described in this section. The simulator needs to interface to a video codec so that adaptive encoding mechanisms can be evaluated. We extended ns2 to offer an TCP-based interface named "REAL" via which RTP video and RTCP control packets are exchanged and which also allows ns2 and the codec to synchronize their clocks. The decoder is extended to support the generation of feedback messages, the encoder to react to the received feedback messages.

Figure 1 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 [11]. 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 behavior as the 3GPP Simulator [2].

3.1. Network Setup

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

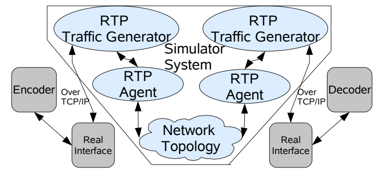

Figure 1. Simulator System Overview

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

We use an RTP packet size of 200 bytes (constant for simplicity), yielding a total packet size of 228 bytes including the UDP and IP header. 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 reassembly cannot succeed for all those packets of which fragments were contained in the respective frame. Hence, a single RLC frame (PDU) loss can result in loss of one or more media packets.

The network topology used for the simulation is depicted in figure 2. 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.

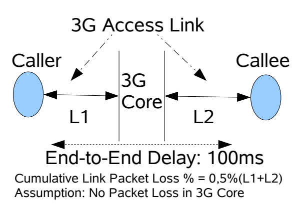

Figure 2. Network Topology

3.2. Codec Setup

We use the Nokia H.264 codec [10]. Three well-known media sequences (Foreman, Football and News) are used to study the error resilience mechanisms. The sequences 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 retransmitting 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 are created by repetition to obtain 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

This section summarizes the findings from our simulations for the four error resilience mechanisms.[2]

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 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 analyze the effectiveness of NACK-based approach for low delay scenarios. We have performed simulations using a one-way delay of 60 ms.

The tables 2 and 3 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.

[2] Further details are available in an extended version of this paper at http://www.netlab.tkk.fi/˜jo/papers/2008-er-3g-conf-ext.pdf.

Table 2. 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 3. 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

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

4.2. Slice Size Adaptation (SSA)

For conversational video 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 slices 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.

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. RTCP reporting for SSA is based on the timing rules defined in RFC 3550 (5 s ± 50 %) [14].

Our simulations show that, in the lossy scenario, a slice size of 400 bytes performs badly when compared to one of

Table 4. PSNR as function of slice size and loss scenario

	200 bytes	400 bytes	Adaptive
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

200 bytes and the adaptive mechanisms (the latter can reach slice sizes as low as 150 bytes). But in the no loss scenario, slice size 200 performs badly when compared to 400 and adaptive (the 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 4.

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. Header compression is not applicable end-to-end and typically not supported in wireless SIP endpoints nor in WLAN installations. This introduces significant overhead when small slice sizes are used (17 % for a slice size of 200 bytes). Also, WLAN performance suffers from small packet sizes. Finally, the public Internet introduces different loss patterns and congestion losses may need to be distinguished from error losses. These mixed scenarios are for further study.

4.3. Reference Picture Selection (RPS)

For RPS, RTCP-encapsulated feedback messages are generated by the receiver to indicate, in case of losses, to the sender which pictures have been decoded correctly/incorrectly. (Regular) RTCP reports are sent every 250 ms and consume about 2 % of the media bandwidth.

Table 5 shows the average PSNR of the RPS compared to an unprotected sequence (serving as a reference) in an error free scenario.[3] RPS achieves better PSNR because the encoder uses the best matching latest reference frame in displaying order as reference frame.

To evaluate the error resilience performance of RPS at frame level, we choose the 0.5% link loss scenario. PSNR and the error propagation period serve as performance measures; the latter indicates the number of consecutive frames impacted by an erroneous frame, from the first one referencing it until (and excluding) the first picture using the new (correct) reference at the decoder. Table 6 shows the mean PSNR. 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. We observe that, at a one way delay of 100 ms and a frame rate of 15 fps, error propagation is stopped by RPS in about four to seven pictures.[4] This means correction takes place in about 240 to 420 ms, which shows the effectiveness of the mechanism.

Table 5. 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

Table 6. 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

4.4. FEC Protection for Reference Frames

The use of FEC to protect reference pictures for the sequences Foreman, Football and News consumes close to 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 5.

Table 6 shows the PSNR with an RLC frame loss rate of 0.5 %. The performance the of 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 7 shows that, by using FEC, 21–24% of the lost packets can be recovered.

From the above results, it is clearly visible that subsequence-based UEP does not perform as good as the unprotected case, i.e., the additional overhead exceeds the gain of FEC protection for the evaluated scenario. However, UEP can still be applicable for lossy environments with high link delay where FEC-based recovery may effectively the only repair option.

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

[4] For details, refer again to the extended version of this paper.

Table 7. 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

$$A = \%\_of\_packets\_lost(including\_FEC\_packets)$$
$$B = \%\_of\_packets\_lost\_after\_FEC\_correction$$
$$C = \frac{A-B}{A} \times 100$$

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 3.

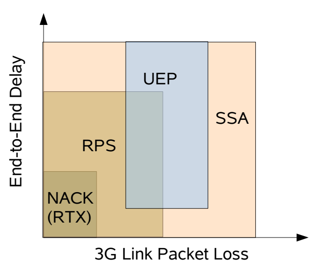

Figure 3. Applicability of ER Mechanisms

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-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 heterogeneous scenarios to ultimately derive adaptive error resilience mechanisms suitable for the entire wired and wireless Internet.

References

[1] Homepage of the Network Simulator (ns2) and the Network Animator (nam). http://nsnam.isi.edu/nsnam.

[2] 3GPP. Software Simulator for MBMS Streaming over UTRAN and GERAN. Proposal S4-050560, 3rd Generation Parnership Proejct (3GPP), Sept. 2005.

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

[4] P. Chou and Z. Miao. Rate-distortion optimized streaming of packetized media, 2001.

[5] 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.

[6] 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.

[7] ITU-T Rec. H.264. Advanced video coding for generic audiovisual services.

[8] S. Kumar, L. Xu, M. K. Mandal, and S. Panchanathan. Error resiliency schemes in H.264/AVC standard. *Journal of Visual Comm. and Image Representation*, 17:425–450, 2006.

[9] A. Li. RTP Payload Format for Generic Forward Error Correction. RFC 5109, Dec. 2007.

[10] Nokia. H.264/AVC codec donation - Licensing Statement. Information S5-050673, 3rd Generation Parnership Proejct (3GPP), Sept. 2005.

[11] 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, 2006.

[12] C. Perkins and O. Hodson. Options for Repair of Streaming Media. RFC 2354, June 1998.

[13] J. Rosenberg and H. Schulzrinne. An RTP Payload Format for Generic Forward Error Correction. RFC 2733, 1999.

[14] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 3550, July 2003.

[15] Y. Wang, S. Wenger, J. Wen, and A. Katsaggelos. Review of error resilient coding techniques for real-time video communications. *IEEE Signal Processing Magazine*, 17(4), 2000.

[16] Y. Wang and Q. Zhu. Error control and concealment for video communication: A review. *Proceedings of the IEEE*, 86(5):974–997, 1998.

[17] 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.

[18] S. Wenger. H.264/AVC over IP. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(7):645–656, 2003.

[19] S. Wenger, U. Chandra, M. Westerlund, and B. Burman. Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF). RFC 5104, Feb. 2008.

[20] S. Wenger and J. Ott. A joint source/channel coding approach for low-delay video. January 2008.