ERROR-RESILIENT PACKET SWITCHED H.264 MOBILE VIDEO TELEPHONY WITH LT CODING AND REFERENCE PICTURE SELECTION

Muneeb Dawood, Raouf Hamzaoui, Shakeel Ahmad, Marwan Al-Akaidi

Department of Engineering, De Montfort University
The Gateway, LE1 9BH, Leicester,UK
muneeb.dawood@email.dmu.ac.uk, rhamzaoui@dmu.ac.uk, sahmad@dmu.ac.uk, mma@dmu.ac.uk

ABSTRACT

Packet switched video telephony over wireless networks for hand-held devices requires low-delay, low-complexity error control mechanisms to deal with packet loss. We present an efficient solution for 3G networks based on LT coding, reference picture selection, and cross-layer optimization. Experimental results on a 3G network simulator for H.264 compressed standard video sequences show that our method achieves significant peak-signal-to-noise ratio and percentage degraded video duration improvements over a state of the art technique.

1. INTRODUCTION

Wireless communications over packet networks suffer from fading, additive noise, and interference, which translate into packet loss. Since modern video encoders deliver video packets with decoding dependencies, packet loss can significantly degrade the video quality at the receiver. Many application-layer error control techniques have been proposed to combat packet loss in wireless networks [1, 2, 3]. However, most of them are not suitable for video telephony on mobile phones because they are too computationally complex or require unacceptable end-to-end delays. In this paper, we propose a solution based on forward error correction (FEC) with LT coding [4]. LT codes are well suited for real-time applications because of their low computational complexity. Our solution adapts the streaming system of [5] to the stringent limitations imposed by video telephony on mobile devices in a 3G network environment. Moreover, in contrast to [5], we exploit reference picture selection and cross-layer optimization. To the best of our knowledge, this is the first time that FEC with a rateless code, reference picture selection, and cross layer optimization are combined. Experimental results for H.264 compressed standard video sequences show that our solution provides a better end-to-end video quality than the state-of-the art technique of Zia, Diepold, and Stockhammer [6].

The remainder of the paper is organized as follows. In Section 2, we survey previous work on error control techniques for wireless video transmission. In Section 3, we describe our proposed technique. In Section 4, we compare the peak signal to noise ratio (PSNR) and the percentage degraded video duration (PDVD) [7] performance of our technique to that of [6] on a 3G network simulator [8]. Our results show improvements of up to 2.87 dB and 7.67 % in PSNR and PDVD, respectively.

2. RELATED WORK

Application-layer error control techniques for real-time video communication over wireless networks can be classified into seven groups: FEC, retransmission mechanisms (ARQ), error resilient video encoding, error concealment, intra macroblock refresh, reference picture selection, and combinations. In the following, we survey the latest research in the field.

In intra macroblock refresh techniques, some macroblocks in the frame are deliberately encoded in intra mode. Intra macroblock refresh techniques were proposed in [9], [10], [11], [12], and [13]. One drawback of these techniques is that intra coded macroblocks require more bits than inter coded macroblocks. Moreover, determining the macrobloks to be intra coded is very difficult in wireless networks.

Error tracking [14] is an intra macroblock refresh technique that requires precise identification of the spatio-temporal error propagation. This is a computationally expensive process that can be too complex for hand-held devices.

Reference picture selection techniques can be used in either the NACK or ACK mode. In the NACK mode, the receiver sends information about the damaged frame. Using this information, the encoder does not use the damaged frame as a reference frame for encoding the next frames. In the ACK mode, the decoder sends information about the correctly received frames, and the encode uses only those frames as reference frames.

In [15] and [16], reference picture selection is combined with feedback. However, the method uses rate-distortion optimization to select the best reference frame, which is too computationally complex for hand-held devices.

Zia, Diepold, and Stockhammer [6] use a combination of error tracking and reference picture selection. Reference picture selection is used in the NACK mode. When the encoder receives a NACK, it uses a rate-distortion optimized mode decision for encoding each macroblock.

Chung-How and Bull [17] applied FEC in conjunction with periodic reference frame selection to stop error propagation. In periodic reference frame selection, every nth frame is coded with the nth previous frame as a reference. This method lacks adaptivity as the reference frame selection is done irrespective of the transmission errors.

In [18], FEC is used with intra slice update where all macroblocks belonging to a slice are encoded in intra mode. Intra slices severely degrade the compression efficiency in case of no losses.

In [19], channel-adaptive hybrid ARQ-FEC is used, and an algorithm is proposed to determine the channel code rate and the maximum number of retransmissions for ARQ using the bit error rate as the channel parameter. One limitation of

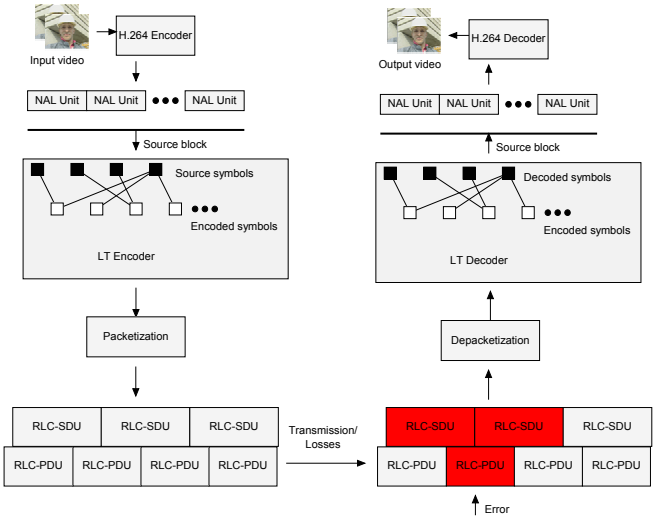

Figure 1: Proposed video transmission system.

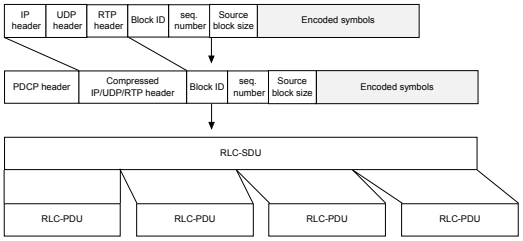

Figure 2: Packetization.

this technique is that it does not stop error propagation.

In [20], FEC and retransmission are used with accelerated retroactive decoding to stop error propagation. This technique is too computationally complex for hand-held devices.

3. PROPOSED METHOD

Figure 1 shows the block diagram of the proposed video transmission system. At the sender side, live video frames are given to the H.264 video encoder which compresses them and generates a sequence of Network Abstraction Layer (NAL) units. The NAL units corresponding to a fixed number of input frames form a source block.

LT encoding is applied to the source block to generate encoded symbols. The performance of standard LT coding is enhanced with block duplication [21]. This method increases the probability of decoding success by virtually increasing the number of source symbols.

The encoded symbols are packetized into IP/UDP/RTP packets (Figure 2). The first two bytes of the RTP payload contain the block ID of the source block to which the encoded symbols belong. The next two bytes of the RTP payload contain the sequence number of the first encoded symbol in the RTP packet. The following two bytes indicate the size of the source block.

The IP/UDP/RTP header is compressed with robust header compression [22] and a two-byte Packet Data Convergence Protocol (PDCP) header is appended to the resulting IP packet to form a Radio Link Control Service Data Unit (RLC-SDU). The RLC-SDU is mapped onto Radio Link Control Protocol Data Units (RLC-PDUs) (Figure 2). The mapping of an RLC-SDU onto RLC-PDUs is done such that if an RLC-PDU contains the last byte of an RLC-SDU and not all the bytes in the RLC-PDU have been used, then the first byte of the next RLC-SDU is concatenated to the last byte of the previous RLC-SDU in the same RLC-PDU. Thus an RLC-PDU may contain data of more than one RLC-SDU. RLC-PDU are transmitted over a 3G network. Due to bit errors in a received RLC-PDU, all IP packets that are partially or fully mapped to it are lost.

IP packets that are received correctly are passed to the LT decoder. If enough encoded symbols are received, LT decoding will be successful, and all NAL units associated to the source block are recovered.

If LT decoding is not successful, all NAL units associated to the source block are considered to be lost and the video decoder uses frame freeze concealment to replace all frames in the failed source block by the last successfully decoded frame. In this case, a mismatch of reference frames between the sender and receiver occurs, which results in spatio-temporal error propagation. To mitigate it, a variant of the reference picture selection technique [23] is used. The receiver sends a feedback that contains the block ID of the last successfully received block, allowing the transmitter to update the reference frame. This is illustrated in Figure 3. Here source block 2 cannot be decoded. The H.264 decoder conceals the lost frames in source block 2 by the last successfully received frame, which is frame number 2. The receiver sends feedback to the sender. The feedback reaches the encoder when it is about to encode frame number 7 which belongs to source block 4. Using the feedback information, the encoder uses frame number 2 as reference frame for encoding frame number 7. The following frames are encoded normally. Hence spatio-temporal error propagation is stopped. This method requires storing a few recently decoded frames at the sender and receiver.

To send the feedback data, we use RTP header extension [24] in the upstream video traffic (video is sent in both ways since the proposed system is for conversational applications). Eight bytes are added to the RTP payload, two of which indicate the block ID (and hence the frame number) of the frame the encoder should use as a reference to stop error propagation. The other six bytes are RTP header extension syntax. The feedback is sent in five consecutive RTP packets to make the feedback robust to transmission errors.

To further improve the performance of our system, we apply cross layer optimization (Figure 4). The idea is to map one IP packet into exactly one RLC-PDU. In this way, when an RLC-PDU is lost, only one IP packet is lost.

Since LT codes are rateless, the number of encoded symbols that are generated can be chosen such that all the encoded symbols corresponding to one source block are packed into an integral number of equally sized IP packets.

Since the feedback is sent through the upstream video, for an IP packet containing the feedback, the number of encoded symbols is adjusted such that the IP packet size is convenient for cross-layer optimization.

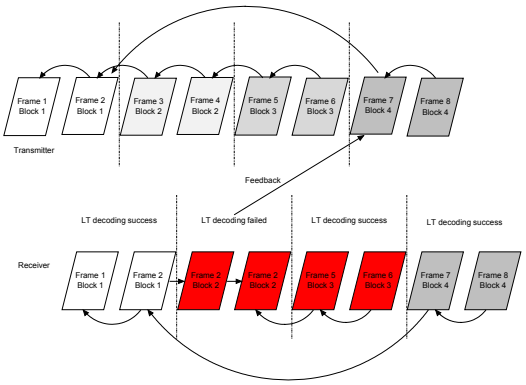

Figure 3: Reference picture selection.

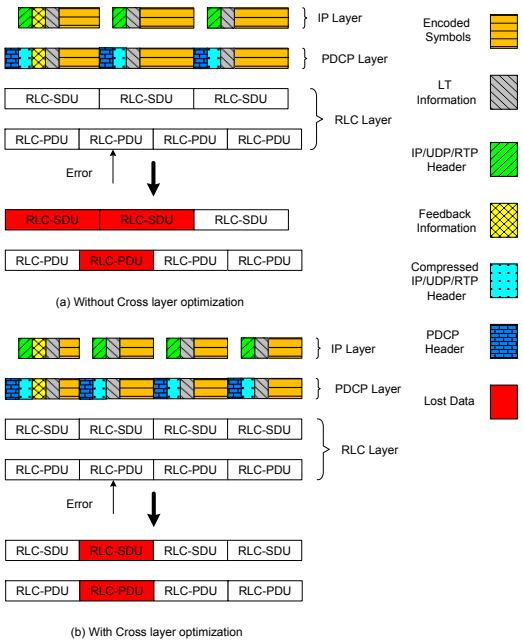

Figure 4: Cross-layer optimization.

4. RESULTS

We compared our results to those of the state of the art technique of [6]. A 3G channel simulator [8] was used for the experiments. The simulator assumes robust header compression [22] to compress the 40-byte IP/UDP/RTP header into 3 bytes. Losses based on traces are applied to the RLC-PDUs. When an RLC-PDU is not received correctly, any IP packet that is mapped to it is discarded.

We provide experimental results for the video sequences Stunt containing 240 frames at 15 frames per second (fps) and Party containing 360 frames at 12 fps. Both video sequences are in QCIF YUV 4:2:0 format. We used the Nokia H.264 video coder with the following settings: one slice per frame, one reference frame, no rate-distortion optimization, no sub 8x8 coding modes, motion vector range equal to 8 pixels, CAVLC entropy encoding. The first frame was encoded as an I frame and the remaining frames were encoded as P frames.

As in [6], the bit rate of the radio access bearer was 128 kbps and the Transmission Time Interval (TTI) was 20 ms, giving a radio frame size of 320 bytes.

A source block corresponded to the NAL units of two video frames. This number was chosen as a compromise between LT coding efficiency which needs a large number of source symbols and frame buffering delay which increases with increasing number of frames. The size of an LT symbol was set to one bit. The Robust Soliton Distribution was used with constants $c = 0.1$ and $\delta = 0.5$. Two expanding factors [21] were used (1 and 8). An expanding factor of 1 means no source block duplication, and an expanding factor of 8 means that the source block is duplicated seven times. Block duplication increases the LT decoding efficiency by increasing the number of edges in the graph of the code. However, this also results in an increase in encoding and decoding time.

The number of encoded symbols in an IP packet was equal to 309 bytes. This is obtained by subtracting the PDCP header size (2 bytes), the compressed IP/UDP/RTP header size (3 bytes), and the LT information overhead (6 bytes) from the RLC-SDU size (320 bytes).

The target LT redundancy was 35 %. This high level of FEC redundancy was chosen to achieve good error resilience over RLC-PDU loss rates in the range $[0,5\%]$. For a transmission rate of 128 kbps, this gave an average video source rate of 89 and 90.3 kbps for the Stunt and Party video sequences, respectively.

We used the cross-layer optimization technique described in the previous section. Thus, for each source block, the number of encoded symbols in bytes was chosen to be an integral multiple of 309 bytes and as close as possible to the number giving an LT redundancy of 35%. If an RTP packet contained feedback information the number of encoded symbols in the packet was adjusted accordingly.

Both the Forward Trip Time (FTT) and Backward Trip Time (BTT) were fixed to 50 ms, which are typical values for HSPA [25].

In [6], two methods are presented; IEC1 gives the best PSNR results, while IEC2 gives the best PDVD results. The PDVD is an alternative objective measure proposed by 3GPP [7]. It indicates the percentage of the time the video was corrupted due to packet losses. Figures 5 and 6 compare our PSNR results to those of IEC1. Figures 7 and 8 compare our PDVD results to those of IEC2. When the RLC-PDU loss rate was zero, the PSNR of IEC1 was greater because the source rate was higher (no channel coding is used in the approach of [6]). However, the source coding efficiency of IEC1 and IEC2 is penalized by the use of smaller slice size (200 bytes per slice) and smaller packet size (leading to larger header overhead). As the RLC-PDU loss rate increased, our method had significantly better PSNR and overall better PDVD results. The improvement in PSNR and PDVD reached 2.45 dB and 1.28 %, respectively for Stunt and 2.87 dB and 7.67 %, respectively for Party.

The frame rate of the Stunt video sequence was 15 fps, giving a frame buffering time of 66.6 ms. Since the RTT was 100 ms, the transmission time of the source block was 120 ms (7 RLC-PDU frames, in average, sent with a TTI of 20 ms), and the time required to declare an LT decoding failure was 30 ms in average, the feedback typically reached the sender after it had encoded six frames. Thus an LT decoding failure affected six frames before the encoder used the feedback information to stop error propagation. In the rare event where

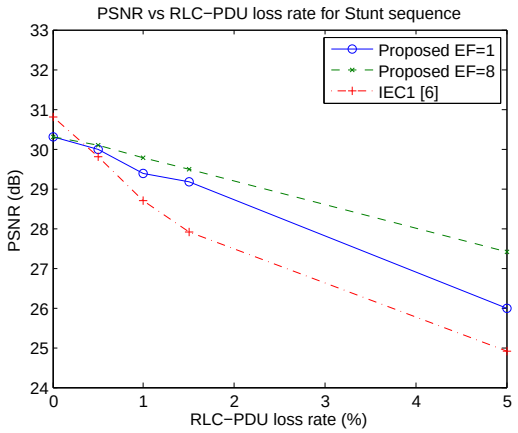

Figure 5: Average Y-PSNR as a function of the RLC-PDU loss rate for the Stunt sequence.

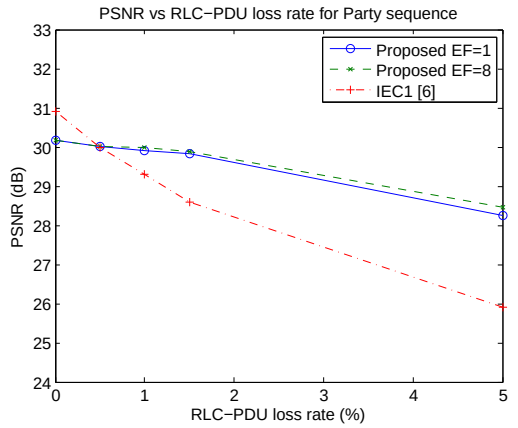

Figure 6: Average Y-PSNR as a function of the RLC-PDU loss rate for the Party sequence.

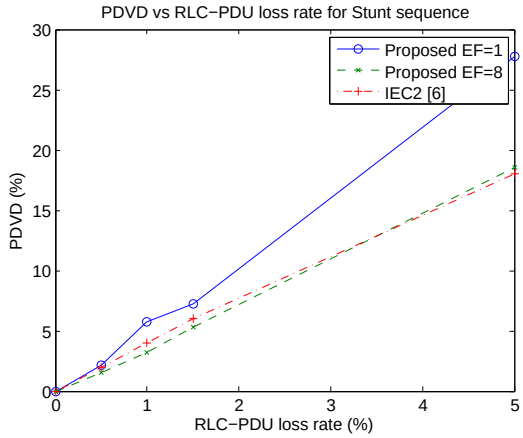

Figure 7: PDVD as a function of the RLC-PDU loss rate for the Stunt sequence.

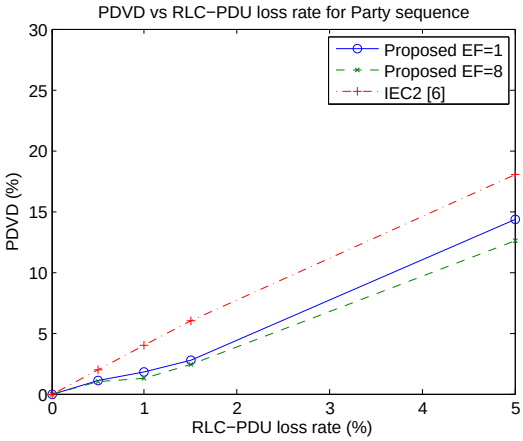

Figure 8: PDVD as a function of the RLC-PDU loss rate for the Party sequence.

the first three RTP packets containing the feedback were lost, the error was propagated to seven frames.

The frame rate of the Party video sequence was 12 fps, giving a frame buffering time of 83.3 ms. Here the feedback reached the sender after it had encoded five frames. In the rare event where the first three RTP packets containing the feedback were lost, the error was propagated to six frames.

For Stunt, the average source block size was 12,064 symbols. Using LT coding with expanding factor 8 gave improvement in PSNR and PDVD over LT coding without duplication for nearly all RLC-PDU loss rates. The improvement in PSNR at RLC-PDU loss rate 5% was largest because for this loss rate, the received overhead is smallest, making block duplication more useful.

For Party, the average source block size was 14,960 symbols. Because the number of source symbols was large, LT coding with block duplication gave only a small improvement in PSNR and PDVD over LT coding without duplication.

Finally, we note that because the source block size was larger for Party than for Stunt, LT coding was more efficient for the first sequence, leading to better PSNR and PDVD performance.

4.1 End to end delay analysis

Table 1 shows the end to end delay components of our system on a PC with Core 2 Duo 1.83 Ghz and 1 GB RAM. For Stunt, the total end to end delay was always below 400 ms, which is the maximum delay specified by 3GPP for conversational applications [26]. For Party, the end to end delay was above 400 ms. The end to end delay can easily be reduced by decreasing the LT decoding time, which is the main contributor to the overall end to end delay.

A more efficient implementation of LT decoding is given in [27]. For a source block size of 1,000,000, the authors report a decoding time of 37 s on a Pentium 4 1.7 GHz, 512 MB RAM as compared to 327 s on our more powerful machine.

This analysis shows that an implementation of our system on hand-held devices with an end to end delay below 400 ms can be achieved.

5. CONCLUSIONS

We presented an error resilience technique for real-time video communication over 3G networks. The method uses LT coding, reference picture selection to stop error propagation, and IP packet size optimization to minimize the effects

Delay Component	Stunt (EF=1)	Stunt (EF=8)	Party (EF=1)	Party (EF=8)
Frame buffering	66.6	66.6	83.3	83.3
H.264 encoding	4.6	4.6	4.6	4.6
LT encoding	0	0	0	0
Transmission delay	120	120	140	140
Propagation delay	50	50	50	50
LT decoding	120	139.2	186	223
H.264 decoding	5.2	5.2	5.2	5.2
Total delay	366.4	385.6	469.1	506.1

Table 1: End to end delays in ms for the Stunt and Party video sequences for expanding factors (EF) 1 and 8.

of error propagation from the RLC layer to the IP layer. Experimental results on a 3G packet loss simulator show that our technique can achieve better PSNR and PDVD results than those of [6].

Currently, our LT coder adds a fixed coding redundancy of about 35% at all loss rates. We expect better results by making the channel coding rate adaptive. This will be the topic of future research.

Since our system targets video telephony, it is important to further reduce the end to end delay. This can be done by a more efficient implementation of the LT decoder or by using Raptor codes instead of LT codes.

REFERENCES

[1] Y. Wang and Q. Zhu, Error control and concealment for video communication: A review, Proc. IEEE, vol. 86, pp. 974–997, May 1998.

[2] Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos, Error resilient video coding techniques, IEEE Signal Processing Magazine, pp. 61–82, July 2000.

[3] T. Stockhammer and W. Zia, "Error-resilient coding and decoding strategies for video communication", in: Multimedia over IP and Wireless Networks, M. van der Schaar and P. A. Chou (eds.), Academic Press, 2007.

[4] M. Luby, "LT-Codes", Proc. 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002.

[5] S. Ahmad, R. Hamzaoui, and M. Al-Akaidi, "Robust live unicast video streaming with rateless codes", in Proc. 16th Int. Packet Video Workshop, Lausanne, Nov. 2007.

[6] W. Zia, K. Diepold, and T. Stockhammer, "Complexity constrained robust video transmission for hand-held devices," in *Proc. IEEE ICIP 2007*, Vol. 4, Sept. - Oct. 2007, pp. 261–264.

[7] 3GPP, Video codec performance, Technical Report 3GPP TR 26.902, 3G Partnership Project.

[8] 3GPP Channel Simulator http://www.3gpp.org/ftp/specs/archive/26_series/26.902/.

[9] G. Cote and F. Kossentini, "Optimal mode selection and synchronization for robust video communications over error-prone networks," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 952–965, June 2000.

[10] Q. Chen et al., "Attention-based adaptive intra refresh for error-prone video transmission," *IEEE Communications Magazine*, vol. 45, pp. 52–60, Jan. 2007.

[11] X. Wang et al., "Robust GOB intra refresh scheme for H. 264/AVC video over UMTS," in *Proc. 6th IEE International Conference on 3G and Beyond*, London, UK, Nov. 2005, pp. 1–4.

[12] D. H. Yoon, H. Pang, and S. Ji, "Spiral intra macroblock refresh with motion vector restriction for low bit-rate video telephony over a 3G network," *IEEE Trans. Consumer Elect.*, vol. 50, pp. 1038–1043, Nov. 2004.

[13] Y. J. Liang, K. El-Maleh, and S. Manjunath, "Upfront intra-refresh decision for low-complexity wireless video telephony," in *Proc. ISCAS 2006*, Island of Kos, Greece, May 2006.

[14] W. Tu and E. Steinbach, "Proxy-based error tracking for H.264 based real-time video transmission in mobile environments," in *Proc. IEEE ICME '04*, June 2004, pp. 1367–1370.

[15] T. Wiegand et al., "Error-resilient video transmission using long-term memory motion-compensated prediction," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 1050–1062, June 2000.

[16] Y. J. Liang, M. Flierl, and B. Girod, "Low-latency video transmission over lossy packet networks using rate-distortion optimized reference picture selection," in *Proc. IEEE ICIP 2002*, Rochester, USA, Sep. 2002.

[17] J. T. H. Chung-How and D.R. Bull, "Robust H.263+ video for real-time Internet applications," in *Proc. IEEE ICIP 2000*, Vancouver, Canada, Sept. 2000, pp. 544–547.

[18] Q. Qu, Y. Pei, and J. W. Modestino, "Robust H.264 video coding and transmission over bursty packet-loss wireless networks," in *Proc. IEEE VTC 2003*, Orlando, Florida, Oct. 2003, pp. 3395–3399.

[19] J. Wen, Q. Dai, and Y. Jin, "Channel-adaptive hybrid ARQ/FEC for robust video transmission over 3G," in *Proc. IEEE ICME 2005*, Amsterdam, The Netherlands, July 2005.

[20] I. Rhee and S. Joshi, "Error recovery for interactive video transmission over the Internet," *IEEE Journal on Selected Areas in Commun.*, vol. 18, pp. 1033–1049, June 2000.

[21] S. Ahmad, R. Hamzaoui, and M. Al-Akaidi, "Unequal error protection using fountain codes and block duplication," in *Proc. MESM 2008*, Amman, Jordan, Aug. 2008.

[22] C. Bormann et al., Robust Header Compression, IETF RFC 3095, July 2001.

[23] B. Girod and N. Farber, "Feedback-based error control for mobile video transmission," in *Proc. IEEE*, Oct. 1999, pp. 1707–1723.

[24] D. Singer and H. Desineni, A General Mechanism for RTP Header Extensions, IETF RFC 5285 , July 2008.

[25] H. Holma and A. Toskala (eds.), HSDPA/HSUPA for UMTS High Speed Radio Access for Mobile Communications, Wiley, 2007.

[26] 3GPP, Group Services and System Aspects: Service aspects; Services and service capabilities, Technical Specification 3GPP TS 22.105, 3G Partnership Project.

[27] P. Cataldi, M.P. Shatarski, M. Grangetto, and E. Magli, "Implementation and performance evaluation of LT and Raptor codes for multimedia applications", in *Proc. IIH-MSP'06*, Dec. 2006