A NOVEL PACKET LOSS RECOVERY TECHNIQUE FOR MULTIMEDIA COMMUNICATION

Wenqing Jiang

C-Cube Microsystems, Inc.
1778 McCarthy Blvd.
Milpitas, CA 95035
Email: wjiang@c-cube.com

Antonio Ortega

Integrated Media Systems Center
Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089
Email: ortega@sipi.usc.edu

ABSTRACT

In this paper a novel loss recovery technique is proposed for multimedia communications over lossy packet networks. The proposed technique uses a combination of recent results on multiple description coding and erasure recovery codes in channel coding. The uniqueness of the proposed technique lies in its ability to recover not only the data carried in lost packets, but also the decoding state for successive packets. Experimental results on image and speech coding show that the proposed technique has excellent coding performance compared to some of the best results published and it can also significantly reduce the error propagation in successive packets due to packet losses.

1. INTRODUCTION

With the rapid growth of the Internet, recent years have seen a flurry of research activities in error protection and control for multimedia communications (for a good review see [1, 2]). The single most important driving force behind these research works is the fact that the best-effort service model, as currently being implemented by most Internet service providers, does not guarantee timely lossless packet delivery. Indeed recent studies on Internet packet dynamics have shown that end-to-end packet loss and delay occur quite often especially during "busy" working hours [3, 4] and packet losses, if not dealt with appropriately, can cause very annoying quality variations in the received signal hence degrading the quality of multimedia communications.

A majority of research works on error control and correction have thus far limited themselves to correct only bit errors in the corrupted packets or to recover only the lost (or overly delayed) packets [1, 2, 5, 6]. While these approaches work well for memoryless source codecs, e.g. a pulse code modulation (PCM) coder, in which data is independently coded and thus packets can be independently decoded, problems arise when source codecs with memory are used.

Source codecs with memory usually operate using the knowledge learnt from past encoded data and adapt the coding on the fly. They can also be characterized as a special type of *state* machines where the *state* is defined as the knowledge the codec learned and used for the encoding of new incoming data. The output of a source codec with memory therefore depends on both the incoming data and the codec state (see Fig. 1 for an illustration of the state-dependent decoding process). One example of such codecs is the differential pulse code modulation (DPCM) scheme in which a prediction is adaptively computed for the data to be encoded and only the prediction residue is encoded and transmitted. Another example is adaptive quantizer based codecs in which quantization stepsizes or codebooks to be used are updated constantly using the statistics of past encoded data [7]. In both cases, the state information, i.e., predictions in DPCM codecs or codebooks in adaptive quantization codecs, is not transmitted but with the assumption that the decoder will be able to derive it on its own using the past decoded data.

Such a precondition for correct decoding, though always true for error-free transmissions, can not be guaranteed for communications over lossy packet networks. As a result packet losses will not only increase distortion level in the corresponding encoded signal segment but also disrupt the decoding state for successive packets. Such a decoder malfunction in case of channel errors, often resulting in *error propagation* in the received signal, has been studied recently in the context of packet communication for DPCM codecs [8], the CELP-based speech codec G.729 [9] and motion compensated hybrid video codecs [10, 11]. In some cases, it is found that distortion caused by state loss is more annoying than that due to data loss since the former propagates in time and has lasting negative effect on human perceptions [9, 10]. A good error control scheme therefore has to be able to recover both the lost data and the lost decoding state in order to minimize the signal quality drop in the presence of packet losses.

An often used technique to prevent error propgation is to refresh the decoding state periodically, e.g., inserting Intra-frames (I-frame) at certain intervals in most hybrid motion-compensated video codecs [10]. The expense for doing so, however, can sometimes become significant; in low bit rate applications bits used for encoding I-frames can be an order higher than that needed by predictive-frames (P-frame) or bidirectional predictive frames (B-frame) and high loss rate channels often necessitates a high rate of I-frame insertion for reasonable signal quality receptions.

In this paper we provide a different approach using a combination of recent results on multiple description coding (MDC) and error correction codes. The explicit redundancy-based MDC techniques for error protection have been shown to yield very competitive performances over lossy packet channels [12, 13, 14, 15] and are also capable of erasure recovery for predictive codecs [8]. Error correcting codes, specifically the Reed-Solomon erasure codes, can also significantly improve the error robustness of encoded bitstreams [6, 16]. We combine merits from these two worlds and propose a new scheme which can approximately recover not only the lost packet data but also the lost decoding state.

The novelty of our schemes lies in its competitive coding performance for packet erasure recovery and its applicability to a wide range of state-dependent source codecs used in practice, e.g. ADPCM based speech codecs (G.721/G.722/G.723, etc.) and hybrid motion compensated video codecs (H.263, H.263+, and MPEG4, etc.). An earlier work of the same philosophy has also shown that it can be easily integrated into an adaptive error-control system according to time-varying channel characteristics [17].

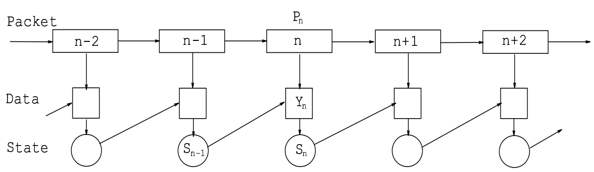

Figure 1: State dependent packet decoding at the receiver.

2. THE PROPOSED SCHEME

2.1. State Dependent Packet Decoding

In Fig. 1 we show a schematic plot of the state dependent packet decoding procedure. Each received packet $\mathbf{P}_n$, when decoded, not only contributes its data part $\mathbf{Y}_n$ for the reconstruction of the encoded signal, but also helps to recover the decoder state $\mathbf{S}_n$ for the correct decoding of next packet. If packet $\mathbf{P}_n$ is lost on the way to the receiver, the decoder will not be able to recover $\mathbf{Y}_n$ nor $\mathbf{S}_n$, whose loss will result in incorrect decoding for multiple successive packets.

2.2. Encoding and Packetization

The proposed technique to combat packet losses is illustrated in Fig. 2 and the encoding and packetization algorithm is defined as follows.

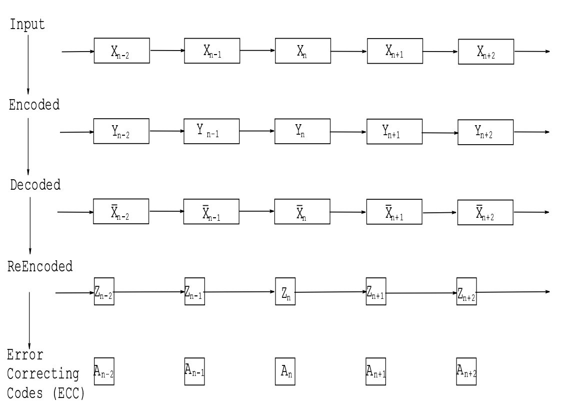

Figure 2: Illustration of encoding and packetization.

Algorithm 2.2 for Encoding and Packetization

Step 1: Split input data sequence $\mathbf{X}$ into small segments based on the packet size and coding rate.
Assume $\mathbf{X}= \{\cdots \mathbf{X}_{n-2}\ \mathbf{X}_{n-1}\ \mathbf{X}_n\ \mathbf{X}_{n+1}\ \mathbf{X}_{n+2} \cdots\}$ with $\mathbf{X}_n$ being one segment of the input $\mathbf{X}$, e.g., one speech frame in a speech coding system or one video frame/field in a video coding system. In a still image coding system, $\mathbf{X}_n$ can also be one polyphase component [12].

Step 2: Encode each segment at a high coding rate R using codec Q_1.
Let $\mathbf{Y}$ be the encoded bitstream. In case of packet loss-free transmission, $\mathbf{Y}$ will be used to reconstruct the encoded signal $\mathbf{X}$.

Step 3: Generate error correcting codes for each segment at a low coding rate ρ.

(a) Decode bitstream $\mathbf{Y}$ and reconstruct from $\mathbf{Y}$ the input as $\bar{\mathbf{X}}$.

(b) Re-encode $\bar{\mathbf{X}}$ at a lower bit rate ρ using codec Q_2. Let $\mathbf{Z}$ be the the newly encoded bistream. In case that any $\mathbf{Y}_n$ is lost in the transmission, $\mathbf{Z}_n$ will be used for the reconstruction of corresponding input segment $\mathbf{X}_n$.

(c) Generate error correcting codes $\mathbf{A}$ for $\mathbf{Z}$.

Step 4: Pack the encoded bitstream $\mathbf{Y}$ and error correcting codes $\mathbf{A}$ into multiple packets $\mathbf{P}$.

As one can see, Step 1 and 2 are essentially common practices in typical packetization procedures. The uniqueness of our scheme lies in the generation of the error correcting codes in Step 3 and the packetization in Step 4, which we now provide design details.

The error-correcting codes we use belong to a special type of block codes, the Reed-Solomon (RS) erasure correction codes. For each K data packets, $N - K$ parity packets are generated using a systematic (N, K) *shortened* RS codes [18]. The flexibility of the choice of N can be used to control the maximum amount of redundancy in the system design, however, a fixed $N = 2K$ is chosen in this paper for simplicity.

In Step 3, error correcting codes $\mathbf{A}$ is generated using $\mathbf{Z}$ but $\mathbf{Z}$ is not transmitted to the receiver as shown in Step 4. In other words some symbols, i.e. $\mathbf{Z}$, used in the process of generating the RS codes are not transmitted but are absolutely necessary for error recovery in the presence of packet losses. To do so, rather than directly encoding the original input $\mathbf{X}$ at a redundant rate to obtain $\mathbf{Z}$, as have been practiced in existing similar works [14, 15, 8, 12, 17], we propose to first decode $\mathbf{Y}$ into $\bar{\mathbf{X}}$ and then re-encode $\bar{\mathbf{X}}$ to generate $\mathbf{Z}$. Such a change guarantees that the receiver can recover $\mathbf{Z}$ using $\mathbf{Y}$ without sending $\mathbf{Z}$ using extra bits. This constitutes a major difference from previous designs and can provide significant coding gains over similar existing works [14, 15, 8, 12, 17]. As will be shown later, it also helps packet loss recovery even if the low bit rate codec Q_2 is also of state-dependent nature, in which case previously proposed techniques will fail [14, 15, 8].

The process for generating the error-correcting codes goes as follows. Every K consecutive packets from $\mathbf{Z}$, i.e. $\{\mathbf{Z}_m, m = n, n+1, \cdots, n+K-1\}$, are used to generate another K parity packets, i.e., $\{\mathbf{A}_m, m = n, n+1, \cdots, n+K-1\}$. To combat packet losses especially burst packet losses of length K, $\mathbf{A}_n$ is packed with K units/packets phase shift relative to data $\mathbf{Y}_n$. One example is to pack $\mathbf{A}_n$ with $\mathbf{Y}_{n+K}$ in packet $\mathbf{P}_{n+K}$, $\mathbf{A}_{n+1}$ with $\mathbf{Y}_{n+K+1}$ in packet $\mathbf{P}_{n+K+1}$ and so on. Such a packetization strategy guarantees that any K received packets can be used to reconstruct the original K packets from $\mathbf{Z}$. Packetization examples for one and two packets losses are shown in Fig. 3 and Fig. 4. As one can see that, only $\mathbf{Y}$, the encoded bitstream at rate R, and $\mathbf{A}$, the erasure recovery codes, are actually transmitted.

2.3. Packet Loss Recovery

2.3.1. Recovery of Lost Data

Algorithm 2.3.1 for Data Recovery

Step 1: Assume K packets are lost, i.e., $\mathbf{P}_n^K\{\mathbf{P}_m, m = n, n+1, \cdots, n+K-1\}$ are lost. Collect next K received packets and extract erasure recovery codes $\mathbf{A}_n^K = \{\mathbf{A}_m, m = n, n+1, \cdots, n+K-1\}$.

Step 2: Decode erasure codes $\mathbf{A}_n^K$ to get $\mathbf{Z}_n^K = \{\mathbf{Z}_m, m = n, n+1, \cdots, n+K-1\}$.

Step 3: Denote reconstructed data from previous K packets, i.e., $\mathbf{P}_{n-K}^K\{\mathbf{P}_m, m = n-K, n-K+1, \cdots, n-1\}$, as $\hat{\mathbf{X}}_{n-K}^K = \{\hat{\mathbf{X}}_m, m = n-K, n-K+1, \cdots, n-1\}$. Re-encode $\hat{\mathbf{X}}_{n-K}^K$ using Q_2 at bit rate ρ to get $\mathbf{Z}_{n-K}^K = \{\mathbf{Z}_m, m = n-K, n-K+1, \cdots, n-1\}$.

Step 4: Decode $\mathbf{Z}_n^K$ with the help of $\mathbf{Z}_{n-K}^K$ if necessary. Recover lost $\mathbf{Y}_n^M = \{\mathbf{Y}_m, m = n, n+1, \cdots, n+K-1\}$ using the newly decoded data.

In Fig. 3 we show the packetization for recovering one lost packet, in which parity code $\mathbf{A}_n$ is piggy-backed in packet $\mathbf{P}_{n+1}$ with one unit delay relative to the primary encoded data $\mathbf{Y}_n$. Note also that parity code $\mathbf{A}_n$ is a function of only $\mathbf{Z}_n$ in this case.

Assuming packet $\mathbf{P}_n$ is lost and packets $\mathbf{P}_{n-1}$ and $\mathbf{P}_{n+1}$ are received correctly. Using Algorithm 2.3.1, the recovery process is straightforward. First, $\mathbf{A}_n$ is extracted from packet in $\mathbf{P}_{n+1}$. By channel decoding, $\mathbf{Z})_\mathbf{n}$ can be reconstructed, which, when decoded, will provide a a coarse quantized version of $\mathbf{X}_n$.

The decoding process of $\mathbf{Z}_n$ depends on the property of the the low bit rate codec Q_2. If Q_2 generates independent bit stream, then each $\mathbf{Z}_n$ can be directly decoded. In such a scenario, Step 3 can be skipped, i.e., there is no need to recover $\mathbf{Z}_{n-1}$. However, if Q_2 decoding is also state dependent, one has to recover first the decoding state for $\mathbf{Z}_n$. In this case, Step 3 is followed to reconstruct $\mathbf{Z}_{n-1}$ first by re-encoding (using Q_2 at rate ρ) $\hat{\mathbf{X}}_{n-1}$. After decoding $\mathbf{Z}_{n-1}$ using Q_2, one finally is able to correctly decode $\mathbf{Z}_n$ to get the low bit rate recovery data for packet $\mathbf{P}_n$.

An example of packetization scheme to protect from two consecutive packet losses is shown in Fig. 4. Note in this case, $\{\mathbf{A}_n, \mathbf{A}_{n+1}\}$ are generated from $\{\mathbf{Z}_n, \mathbf{Z}_{n+1}\}$ using a (4,2) RS erasure code. Their packetization are delayed two units. Details of data recovery is exactly the same as explained before and is omitted here for lack of space.

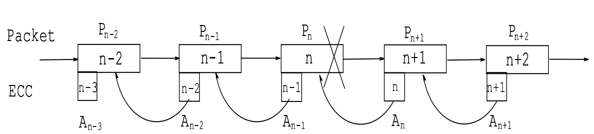

Figure 3: Recovery of single packet loss.

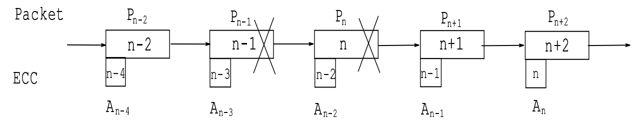

Figure 4: Recovery of two packets loss.

2.3.2. Recovery of Lost Decoding State

The basic idea is inspired by the work by Singh and Ortega in their work on erasure recovery for predictive codecs [8], in which the coarsely quantized data is used to invalidate unlikely sequence decoding paths and the one with the minimum error is chosen as the most likely one. We generalize the idea for any state-dependent codecs (i.e. source codecs with memory) and define the algorithm for decoding state recovery due to packet erasures as follows.

Algorithm 2.3.2 for State Recovery

Step 1: Assume packet $\mathbf{P}_n$ is lost and the decoding state for $\mathbf{P}_{n+1}$ needs to be restored. Collect next $M+1$ successively received packets and extract erasure recovery codes $\mathbf{A}_{n+1}^M = \{\mathbf{A}_m, m = n+1, n+2, \cdots, n+M\}$ and encoded bit streams $\mathbf{Y}_{n+1}^M = \{\mathbf{Y}_m, m = n+1, n+2, \cdots, n+M\}$. Initialize algorithm distortion $D^{(0)}$, loop control variable ϵ, and decoding state $S^{(0)}$.

Step 2: Decode erasure codes $\mathbf{A}_{n+1}^M$ to get $\mathbf{Z}_{n+1}^M = \{\mathbf{Z}_m, m = n+1, n+2, \cdots, n+M\}$. Decode (using codec Q_2) $\mathbf{Z}_{n+1}^M$ to obtain the low bit rate reconstruction of the corresponding input signal. For simplicity, $\mathbf{Z}_{n+1}^M$ is also used to denote this low bit rate reconstruction.

Step 3: For a given decoding state $S^{(k)}$, decode $\mathbf{Y}_{n+1}^M$ to reconstruct the corresponding original input sequence as $\hat{\mathbf{X}}_{n+1}^M = \{\mathbf{X}_m, m = n+1, n+2, \cdots, n+M\}$.

Step 4: Using Q_2, re-encode $\hat{\mathbf{X}}_{n+1}^M$ at rate ρ to obtain $\hat{\mathbf{Z}}^M = \{\hat{\mathbf{Z}}_m, m = n+1, n+2, \cdots, n+M\}$.

Step 5: Compute the distance $D^{(k)} = |\mathbf{Z}_{n+1}^M - \hat{\mathbf{Z}}_{n+1}^M|^2$. If $|D^{(k)} - D^{(k-1)}|/D^{(k)} \leq \epsilon$, stop. The current state $S^{(k)}$ is then the optimal decoding state S^*. Otherwise choose a new state $S^{(k+1)}$ and go back to **Step 3**.

As one can see, the state recovery is formulated as an optimization algorithm over the state space $\mathbf{S}$, i.e. the set of all possible initial states for decoding packet $\mathbf{P}_{n+1}$. The optimal solution S^* is such a decoding state from which the distortion between the received data $\mathbf{Z}$ and the re-encoded data $\hat{\mathbf{Z}}$ is minimized.

3. EXPERIMENTAL RESULTS

The first experiment on still image transmission is used to show the coding performances of the proposed scheme. The basic system framework is the same as that previously presented in [12]. The input image is first wavelet transformed and the wavelet coefficients are polyphase transformed into 16 polyphase components, each of which is then coded independently at 0.4bps using the SPIHT codec [19]. The encoded bitstreams are packed into different packets (which constitute the $\mathbf{Y}$ part in algorithm 2.2). Next

$\mathbf{Y}$ is decoded into $\hat{\mathbf{X}}$ which is then re-encoded at 0.1bps to generate $\mathbf{Z}$. Finally erasure codes $\mathbf{A}$ is generated from $\mathbf{Z}$ using a (32,16) RS erasure code.

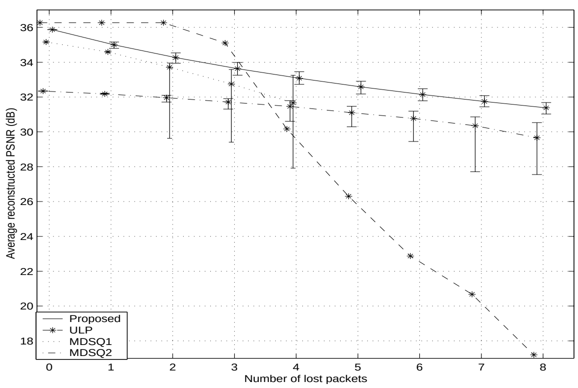

Figure 5: Performance comparisons for Lena 512x512 graylevel image coded at total bitrate 0.5bps with redundancy 20% under different packet loss assumptions (up to 50%). ULP: unequal loss protection[6]. MSDQ1 and MSDQ2: multiple description scalar quantizer based wavelet image coding[5]

Since no decoding dependency exists between consecutive packets in this experiment, erasure codes $\mathbf{Z}$ are packed together with its $\mathbf{Y}$ counterpart without delay, i.e., $\mathbf{P}_n = \{\mathbf{Y}_n\ \mathbf{Z}_n\}$ for $n = 0, 1, \cdots, 15$. As a result total 16 packets are generated and at least 8 packets have to be received to recover all polyphase components (either at 0.4bps or 0.1bps). Fig. 5 gives the reconstructed mean peak signal-to-noise ratios for the Lena image under different packet loss assumptions. The best and the worst PSNRs are also shown in vertical bars. As one can see, the performance of the proposed scheme is very competitive even compared to some of the best coding results published to date.

The second experiment on speech coding is used to demonstrate the state recovery capability of the proposed scheme. The coding algorithms are modified using source codes from RAT 3.0 [20], whose strategy for packet loss recovery is described in part in RFC 2198 [21]. The primary coding used is the Intel/DVI4 ADPCM algorithm which encodes each linear 16-bit sample into a 4-bit symbol. The coding state S constitutes the predicted value $pred$ and the index ind into the quantization stepsize table. The redundant coding is a simplified LPC algorithm which generates 10 prediction coefficients, one period estimation and one gain estimator for each frame.

The speech used is the sentence *draw the outer line first then fill the interior* by a female speaker at sample rate $16KHz$ and 16-bit per sample. There are total 180 packets, each of which consists of 320 speech samples quantized at 4bps using the Intel/DVI4 algorithm. The redundant LPC data is generated on dequantized speech signal and packed with one packet delay (for one packet loss, erasure code can be simply a copy of the data itself). Assuming packet 60 is lost, Figure 6 provides a comparison of peak signal-to-noise ratios (PSNR) of each speech frame before and after the application of algorithm 2.3.2. An exhaustive search is performed to find the decoding state and two packets are used in the optimization process, i.e. $M = 2$. It can be seen that that state recovery significantly reduces reconstruction error for packets immediately after the lost ones thus avoiding further error propagation in the packet sequence.

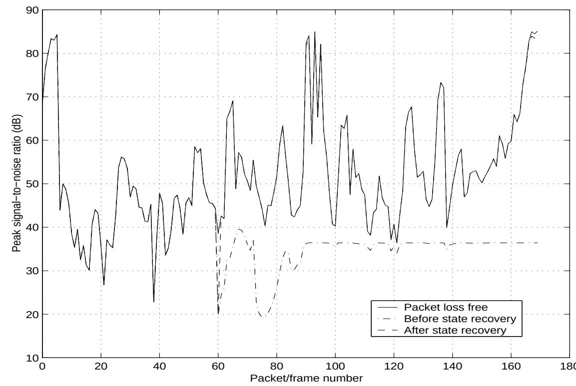

Figure 6: Comparison of frame PSNRs before and after decoding state recovery when packet no.60 is lost. Solid: ADPCM at 4bps; Dash-dotted: before state recovery, PSNRs of successive packets drift away. Dashed: after state recovery, PSNRs of successive packets catches up quickly.

4. CONCLUSIONS

In this paper we have proposed a novel packet loss recovery technique for time-constrained multimedia communications. Detail algorithms for encoding, packetization, data and state loss recovery in the presence of packet losses are also provided. The main advantage of the propose technique is its competitive coding performance, its simplicity in system implementation and its applicability to a wide range of multimedia codecs. There are however several issues remain open for further researches, e.g., the optimality of redundancy rate allocation (i.e., ρ w.r.t R given total rate $R_0 = R + \rho$ and channel statistics), the optimality of packet sequence length M used in the state recovery algorithm, and how to accurately characterize the quality drop due to lost decoding state for multimedia communications.

The authors would like to thank Alex Mohr and Sergio Servetto for providing their coding results used in Fig. 5.

5. REFERENCES

[1] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet-loss recovery techniques for streaming audio," *IEEE Network Magazine*, Sept./Oct. 1998.

[2] C. Perkins and O. Hodson, *Options for Repair of Streaming Media*, June 1998, RFC2354.

[3] V. Paxon, "End-to-end internet packet dynamics," in *Proceedings of SIGCOMM*, 1997.

[4] J. Bolot, "End-to-end packet delay and loss behavior in the internet," in *Proc. of SIGCOMM'93*, Sept. 1993, pp. 289–298.

[5] S. D. Servetto, K. Ramchandran, V. A. Vaishampayan, and K. Nahrstedt., "Multiple description wavelet based image coding," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 813–826, 2000.

[6] A. E. Mohr, E. A. Riskin, and R. E. Ladner, "Generalized multiple description coding through unequal loss protection," in *Proc. Int. Conf. on Image Proc. (ICIP)*, Kobe, Japan, Oct. 1999.

[7] A. Ortega and M. Vetterli, "Adaptive scalar quantization without side information," *IEEE Transactions on Image Processing*, vol. 6, no. 5, pp. 665–676, may 1997.

[8] R. Singh and A. Ortega, "Erasure recovery in predictive coding environments using multiple description," in *Proc. of IEEE Workshop on Multimedia Signal Processing*, Copenhagen, Denmark, Sept. 1999.

[9] J. D. Rosenberg, "G.729 error recovery for internet telephony," http://www.cs.columbia.edu/ jdrosen, 1996.

[10] K. Stuhlmuller, N. Farber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE Journal on Selected Areas in Communications. Special Issue on Error-Resilient Image and Video Transmission*, vol. 18, no. 6, pp. 1012–1032, June 2000.

[11] J. Apostolopoulos, "Error-resilient video compression through the use of multiple states," in *Proc. Intl. Conf. on Image Proc.*, 2000, vol. III, pp. 352–355.

[12] W. Jiang and A. Ortega, "Multiple description coding via polyphase transform and selective quantization," in *Proc. of Visual Communication and Image Processing*, San Jose, CA, Jan. 1999.

[13] W. Jiang and A. Ortega, "Multiple description speech coding for robust communication over lossy packet networks," in *Proc. of Intl. Conf. on Multimedia Engineering*, New York, NY, July 2000.

[14] V. Hardman M. A. Sasse, M. Handley, and A. Watson, "Reliable audio for use over the Internet," in *Proc. INET*, 1995.

[15] J.-C. Bolot, S. Fosse-Parisis, and D. Towsley, "Adaptive FEC-based error control for Internet telephony," in *Proc. IEEE INFOCOMM'99*, 1999, vol. 3, pp. 1453–1460.

[16] L. Rizzo, "Effective erasure codes for reliable computer communication protocols," *ACM Computer Communication Review*, vol. 27, no. 2, pp. 24–36, Apr. 1997.

[17] P. Sagetong and A. Ortega, "Optimal bit allocation for channel adaptive multiple description coding," in *Proc. of Electronic Imaging*, Jan. 2000.

[18] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*, Prentice Hall, 1983.

[19] A. Said and W. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits and Sys. for Video Tech.*, vol. 6, no. 5, pp. 243–250, June 1996.

[20] "Robust audio tool," http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/.

[21] C. Perkins, I. Kouvelas, O. Hodson, V. Hardman, M. Handley, J. C. Bolot, A. Vega-Garcia, and S. Fosse-Parisis, *RTP Payload for Redundant Audio Data*, Sept. 1997.