Robust Layered Multiple Description Coding of Scalable Media Data for Multicast

Vladimir Stanković, Raouf Hamzaoui, Zixiang Xiong

***Abstract*— Layered multiple description codes allow robust transmission of scalable media data over packet erasure networks, while providing simple rate adaptation and bandwidth savings for shared bottleneck links. We show how to efficiently design layered multiple description codes for multicast and broadcast applications in memoryless packet erasure networks. Our approach offers a significantly better quality trade-off among clients than the best previous solution.**

I. Introduction

Robust transmission of scalable media bitstreams over packet erasure networks can be achieved with the multiple description (MD) forward error correction-based system of [1]–[3]. The system transforms a scalable information bitstream into packets (descriptions) of equal length such that information data of decreasing importance are protected with increasingly weaker maximum distance separable erasure-resilient codes.

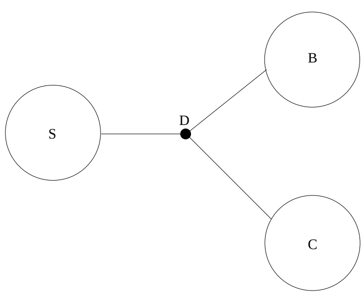

Fig. 1. A common network where server S is connected to two clients B and C over a bottleneck link.

In multicast and broadcast applications *layered codes* [4] are desirable. Indeed, assume that two clients B and C receive the same data at two different bit rates (the bit rate for B is smaller than that of C) from server S, while sharing a bottleneck link (Figure 1). Instead of generating and sending a separate bitstream of data to each client, the server can send the same bitstream over the common link. At router D, only a part of the bitstream (first quality layer) is transmitted to B, while client C receives the whole bitstream (both layers). In addition to bandwidth savings, layered coding also offers simple rate adaptation by adding/dropping layers. Finally, it allows efficient congestion control [4], [5].

V. Stanković and Z. Xiong are with the Department of Electrical Engineering, Texas A &M University, College Station, TX, USA. E-mail: stankovi@ee.tamu.edu, zx@ee.tamu.edu. R. Hamzaoui is with the Department of Computer and Information Science, University of Konstanz, Konstanz, Germany. E-mail: hamzaoui@inf.uni-konstanz.de. This paper was presented in part at the Data Compression Conference, Snowbird, UT, March 2004.

To exploit the benefits of both MD coding and layered coding, Chou, Wang, and Padmanabhan [5] proposed codes which split the multiple descriptions of the system of [1] into layers. When two layers are used, the low-bandwidth clients receive only a base layer, while the high-bandwidth clients additionally receive an enhancement layer. Unfortunately, this construction cannot offer to both clients the same quality performance as two separate, optimal, non-layered MD schemes. For example, the scheme of [5] is optimized only for the low-bandwidth clients, and thus the high-bandwidth clients potentially suffer a significant performance loss.

Our goal is to provide a better trade-off between the distortions seen by all clients in the network. To achieve this, we modify the method of [5] and define an optimal layered MD code as one that minimizes the largest performance loss experienced by any client. Such a code tends to average the quality loss among the clients, and thus ensures that none of the clients suffers a significantly higher quality degradation than the others. Finding an optimal layered MD code is a difficult combinatorial optimization problem. To save computing time, we propose two fast heuristic algorithms. Simulations show that our algorithms provide significant improvements in the quality trade-off over the results of [5].

II. Packet Erasure Protection

Suppose that a scalable compressed bitstream is to be protected and transmitted over a packet erasure channel as N packets of payload size K symbols each. The system of [1]–[3] builds K segments $S_1, \ldots, S_K$, each of which consists of $m_i \in \{1, \ldots, N\}$ information symbols, and protects each segment S_i by adding $f_i = N - m_i$ redundant symbols of an (N, m_i) systematic erasure-resilient code of maximum distance (e.g., a Reed-Solomon code). Then the ith packet ($i = 1, \ldots, N$) is formed from the ith symbol of each channel codeword. With the constraint $m_1 \leq \ldots \leq m_K$, one ensures that if at most f_i packets are lost, then the decoder can recover at least the first i segments. Here we also assume that the packet number is indicated in the header of the packet. We denote by $\mathcal{F}_N$ the set of *protections* $(f_1, \ldots, f_K)$ such that $N > f_1 \geq \ldots \geq f_K \geq 0$. We define the *neighborhood* $\mathcal{N}(F)$ of $F = (f_1, \ldots, f_K) \in \mathcal{F}_N$ as the set of protections of the form $(f_1+1, f_2, \ldots, f_K), (f_1+1, f_2+1, \ldots, f_K), \ldots, (f_1+1, f_2+1, \ldots, f_K+1)$ that are included in $\mathcal{F}_N$. Suppose that the packet erasure channel is memoryless with packet erasure rate ϵ. Let ϕ denote the operational distortion-rate function of the source coder and let X be the random variable whose value is the number of packets erased. Then, for a given protection

$F = (f_1, \ldots, f_K) \in \mathcal{F}_N$, the expected distortion is

$$E_N(F, \epsilon) = \sum_{i=0}^{K} P_i(F, \epsilon)\phi(V_i(F)), \quad (1)$$

where $P_0(F, \epsilon) = Prob(X > f_1)$, $P_i(F, \epsilon) = Prob(f_{i+1} < X \leq f_i)$ for $i = 1, \ldots, K-1$, $P_K(F, \epsilon) = Prob(X \leq f_K)$, $V_0(F) = 0$ and for $i = 1, \ldots, K$, $V_i(F) = \sum_{k=1}^{i} m_k$. A protection that minimizes (1) over $\mathcal{F}_N$ will be denoted by $F_N^{(\epsilon)}$. It can be computed in $O(N^2K^2)$ time [6] or closely approximated in $O(NK)$ time with the local search algorithm of [7].

III. Design of Layered Multiple Description Codes

We consider the situation where many clients simultaneously request the same data from a server, while sharing a bottleneck link. A layered multiple description (LMD) protection scheme splits multiple descriptions into *layers*, successive packets of the same payload size, and sends to Client i the first i layers. Thus, if we assume that the packet payload size is K symbols and that the i-th layer consists of N_i packets, then $K(N_1 + \cdots + N_i)$ symbols will be sent to Client i. For clarity, we assume in the following that we have only two layers. The first layer (*base layer*) is sent to the low-bandwidth client (LC), while both the base layer and the *enhancement layer* are sent to the high-bandwidth client (HC). The base layer is protected with $F_1 = (f_1^1, \ldots, f_K^1) \in \mathcal{F}_{N_1}$. Thus, this layer contains the first $\sum_{i=1}^{K}(N_1 - f_i^1)$ information symbols. The enhancement layer consists of q, $0 \leq q \leq N_2$, successive packets of parity symbols used to strengthen the protection of the base layer followed by $N_2 - q$ packets, which are protected with $F_2 = (f_1^2, \ldots, f_K^2) \in \mathcal{F}_{N_2-q}$. In this way, the enhancement layer contains the next $\sum_{i=1}^{K}(N_2 - q - f_i^2)$ information symbols. Note that the HC ignores the last $N_2 - q$ packets of the enhancement layer if it is not able to successfully decode all information symbols of the base layer. In the following, we say that $L = (f_1^1, \ldots, f_K^1, q, f_1^2, \ldots, f_K^2)$ is an (N_1, N_2)-packet LMD protection. Table I shows an example where $N_1 = 3, N_2 = 4, K = 4, (f_1^1, f_2^1, f_3^1, f_4^1) = (2, 1, 1, 0)$, $q = 2$, and $(f_1^2, f_2^2, f_3^2, f_4^2) = (1, 1, 1, 0)$.

Packet 1	1	2	4	6
Packet 2	x	3	5	7
Packet 3	x	x	x	8
Packet 4	x	x	x	x
Packet 5	x	x	x	x
Packet 6	9	10	11	12
Packet 7	x	x	x	13

TABLE I

The first three packets are sent to both the LC and the HC. The remaining four packets are sent to the HC only. Numbers denote information symbols, "x" denotes a parity symbol.

Given an (N_1, N_2)-packet LMD protection $L = (F_1, q, F_2)$, it is easy to show that the expected distortion for the LC is

$$E_{N_1}(L, \epsilon_1) = E_{N_1}(F_1, \epsilon_1) = \sum_{i=0}^{K} P_i(F_1, \epsilon_1)\phi(V_i(F_1)), \quad (2)$$

and the expected distortion for the HC is

$$E_{N_1+N_2}(L, \epsilon_2) = \sum_{i=0}^{K-1} P_i(F_1 + q, \epsilon_2)\phi(V_i(F_1 + q)) + P_K(F_1 + q, \epsilon_2)E_{N_2-q}(F_2, \epsilon_2, V_K(F_1 + q)), \quad (3)$$

where ϵ_1 and ϵ_2 are the packet erasure rates in the connections between the server and the LC and the server and the HC, respectively. Here we use the notations $F_1 + q = (f_1^1 + q, \ldots, f_K^1 + q) \in \mathcal{F}_{N_1+q}$ and $E_N(F, \epsilon, t) = \sum_{i=0}^{K} P_i(F, \epsilon)\phi(t + V_i(F))$, for $F \in \mathcal{F}_N$ and $t \geq 0$. Note that if F_1 is optimal for the LC, the HC will have a performance loss compared to the case where $F_{N_1+N_2}^{(\epsilon_2)}$ is used. Similarly, if $F_{N_1+N_2}^{(\epsilon_2)}$ is used for the HC, then $F_1 \neq F_{N_1}^{(\epsilon_1)}$, and the LC suffers a performance loss. Thus, with an LMD protection, both clients cannot simultaneously obtain the smallest possible distortion (as with two optimal non-layered MD protections).

It is shown in [5] that a naive approach to solve the problem by optimizing the protection for only one client usually leads to very high distortions for the non-optimized client. A better approach called LMD coding by unequal erasure protection [5] (and referred to as the *q-method* in the following) uses the LMD protection $L = (F_{N_1}^{(\epsilon_1)}, q, F_2)$ where F_2 and q are chosen to minimize (3) subject to $F_1 = F_{N_1}^{(\epsilon_1)}$. In this way, the LC always has an optimal performance, while the HC suffers a performance loss. For example, for the Foreman video sequence encoded with MPEG-4 FGS, the expected distortion for the HC was 1.4 dB worse than the minimum possible [5].

To reduce such a large performance loss, we propose to minimize the maximum performance loss for the two clients, that is, we look for an (N_1, N_2)-packet LMD protection that minimizes the cost function

$$\max\{|E_{N_1}(L, \epsilon_1) - E_{N_1}(F_{N_1}^{(\epsilon_1)}, \epsilon_1)|, |E_{N_1+N_2}(L, \epsilon_2) - E_{N_1+N_2}(F_{N_1+N_2}^{(\epsilon_2)}, \epsilon_2)|\}. \quad (4)$$

Since the number of candidates is $\binom{N_1+K-1}{K}\sum_{q=0}^{N_2}\binom{N_2-q+K-1}{K}$, minimizing (4) with brute force is not feasible. In the following, we propose two heuristic iterative improvement algorithms that compute an approximate solution in reasonable time.

Algorithm 1: Input: $K, N_1, N_2, \epsilon_1, \epsilon_2, \phi$. Output: an (N_1, N_2)-packet LMD protection L^*.

1. **Initialization:** Compute $D_1^* = \arg\min_{S \in \mathcal{F}_{N_1}} E_{N_1}(S, \epsilon_1)$ and $D_2^* = \arg\min_{S \in \mathcal{F}_{N_1+N_2}} E_{N_1+N_2}(S, \epsilon_2)$. Set $F_1 = D_1^*$. Set $F_2 = \arg\min_{S \in \mathcal{F}_{N_2}} E_{N_2}(S, \epsilon_2, V_K(F_1))$. Set $L^* = (F_1, 0, F_2)$, $q = 0$, and $\min = \max\{|E_{N_1}(L^*, \epsilon_1) - E_{N_1}(D_1^*, \epsilon_1)|, |E_{N_1+N_2}(L^*, \epsilon_2) - E_{N_1+N_2}(D_2^*, \epsilon_2)|\}$.
2. **Refinement:** If $\mathcal{N}(F_1) = \emptyset$, go to 4. Otherwise, let $S_1 = (s_1, \ldots, s_K) = \arg\min_{S \in \mathcal{N}(F_1)} E_{N_1}(S, \epsilon_1)$. Set $S_2 = \arg\min_{S \in \mathcal{F}_{N_2-q}} E_{N_2-q}(S, \epsilon_2, V_K(S_1))$. Set $L' = (S_1, q, S_2)$.
3. Compute $\Delta = \max\{|E_{N_1}(L', \epsilon_1) - E_{N_1}(D_1^*, \epsilon_1)|, |E_{N_1+N_2}(L', \epsilon_2) - E_{N_1+N_2}(D_2^*, \epsilon_2)|\}$. If $\Delta < \min$, set $\min = \Delta$, $L^* = L'$, $F_1 = S_1$. If $s_1 < N_1$ go to 2.
4. Set $q = q + 1$. If $q > N_2$, output L^* and stop.
5. Set $F_2 = \arg\min_{S \in \mathcal{F}_{N_2-q}} E_{N_2-q}(S, \epsilon_2, V_K(D_1^*))$. Set $L' = (D_1^*, q, F_2)$. Compute $\Delta = \max\{|E_{N_1}(L', \epsilon_1) -$

$E_{N_1}(D_1^*, \epsilon_1)|, |E_{N_1+N_2}(L', \epsilon_2) - E_{N_1+N_2}(D_2^*, \epsilon_2)|\}$. If $\Delta < \min$, set $\min = \Delta$ and $L^* = L'$.

6. Set $F_1 = D_1^*$ and go to 2.

The optimal protections D_1^*, D_2^*, F_2, S_2 (in Steps 1, 2, and 5) can be computed with the optimal algorithm of [6]. To reduce the execution time, one can use instead the suboptimal but faster algorithms of [2], [3], [7]. In all our simulations, we used the local search algorithm of [7].

Algorithm 1 starts by computing an optimal protection for the base layer and an optimal protection for the enhancement layer. The resulting LMD protection L^* is that of the q-method with $q = 0$. In the refinement phase, we update the solution as long as we can decrease the cost function (4). This step worsens the protection of the base layer, but improves the performance for the HC. Then, motivated by the observation that an increase of the number of sent packets requires a stronger protection [8], we use q parity packets from the enhancement layer to strengthen the protection of the base layer and repeat the search. Note that the solution computed by the algorithm cannot be worse than the one found with the q-method.

Compared to the q-method, Algorithm 1 reduces the performance loss for the HC. However, this is penalized by the appearance of a small performance loss for the LC. Typically, the performance loss for the HC will be much larger than the one for the LC. The reason is that in Step 6 we set the temporal solution for the LC, F_1, to D_1^*, which is optimal for the LC. To improve the performance trade-off among the two clients, we propose the following variant, which we call Algorithm 2. It is identical to Algorithm 1 with the exception of two modifications. In Step 6, we do not set F_1 to D_1^*; instead, we set F_1 to $F_1 + q$. Also the solution to the refinement in Step 2 is done for packet erasure rate ϵ_2, that is, we set $S_1 = \arg\min_{S \in \mathcal{N}(F_1)} E_{N_1+q}(S, \epsilon_2)$ and $L' = (S_1 + (-q), q, S_2)$. In this way, Algorithm 2 tries to avoid getting stuck at a solution whose base-layer part is too close to D_1^*.

When the local search algorithm of [7] is used, the worst-case complexity of both algorithms is $O(N_2^2 N_1)$.

IV. Results

This section provides a comparison between the q-method of Chou *et al.* [5], Algorithm 1, and Algorithm 2. An exponential model was used to model the packet loss rate in the channel [2], [7]. In all experiments, the number of packets in the base layer was fixed to $N_1 = 128$, and the number of packets in the enhancement layer, N_2, was varied from 10 to 125. The scalable information bitstream was generated with the SPIHT algorithm [9] for images and 3D-SPIHT [10] for video. The packet payload size was equal to $K = 48$ bytes for images and 200 bytes for video. Instead of minimizing the expected distortion, we maximized the expected peak signal to noise ratio (PSNR). This was achieved in a straightforward manner by adapting the cost functions and the algorithms accordingly.

Figure 2 shows results for the standard grey-scale 8 bits per pixel 512×512 Lenna. The mean packet loss rate in the link between the server and the LC was $\epsilon_1 = 0.05$ and that of the link between the server and the HC was $\epsilon_2 = 0.2$. For the q-method, the loss experienced by the HC was as high as 1.35 dB. (We recall that the q-method always provides optimal protection for the LC.) In contrast, with Algorithm 2, the highest loss was 0.65 dB for the LC and 0.69 dB for the HC. Algorithm 1 gave a smaller loss for the LC ($<$ 0.05 dB), but the loss for the HC was up to 1 dB.

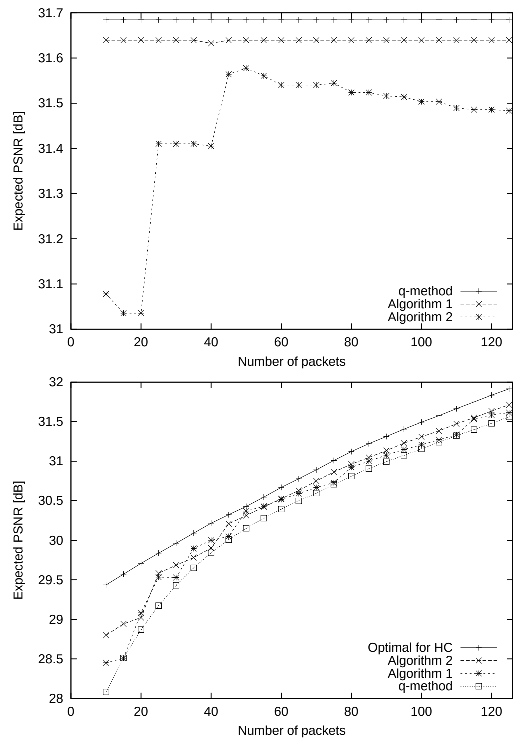

Fig. 2. Expected PSNR for the LC (top) and the HC (bottom) as a function of the number of packets in the second layer. The number of packets in the first layer is fixed to 128. The mean packet loss rate is 0.05 and 0.2 for the LC and the HC, respectively. Results are for the SPIHT bitstream of Lenna.

If we compare the average loss of the LC and the HC, Algorithm 1 always provided the best result, which was up to 0.35 dB better than that of the q-method.

Figure 3 compares the results for the 512×512 Peppers image. Results for the standard 176 $\times 144$ QCIF Foreman video sequence are presented in Figure 4. Table II shows the largest expected loss caused by each algorithm.

We obtained similar results for other packet loss rates ϵ_1, ϵ_2 with $0.01 \leq \epsilon_1 < \epsilon_2$ or $\epsilon_1 < \epsilon_2 < 0.1$. However, when $\epsilon_1 = \epsilon_2$, the q-method usually had a small performance loss, and thus our algorithms were able to improve the solution only slightly. Also, when $\epsilon_1 > \epsilon_2$, our algorithms were not able to significantly improve the q-method. On the other hand, when $\epsilon_1 < 0.01$ and $\epsilon_2 > 0.1$, all three methods experienced a large performance loss. For example, for the Lenna image, when we assumed an error-free link between the server and the LC, and heavily corrupted the link between the server and the HC (the packet erasure rate was 0.2), then for $N_1 = 128$ and $N_2 = 10$, the HC suffered a performance loss (compared to the optimal value of 29.44 dB) as high as 9.11 dB, 8.39 dB, and 1.60 dB with the q-method, Algorithm 1, and Algorithm

Algorithm	Lenna	Peppers	Goldhill	Foreman
q-method	1.35	1.78	1.52	1.49
Algorithm 1	1.06	1.46	1.22	1.32
Algorithm 2	0.69	0.69	0.52	0.70

TABLE II
LARGEST EXPECTED PSNR LOSS IN dB. THE NUMBER OF PACKETS IS FIXED TO 128 FOR THE LC AND VARIED BETWEEN 10 AND 125 FOR THE HC. THE MEAN PACKET LOSS RATE IS 0.05 AND 0.2 FOR THE LC AND HC, RESPECTIVELY.

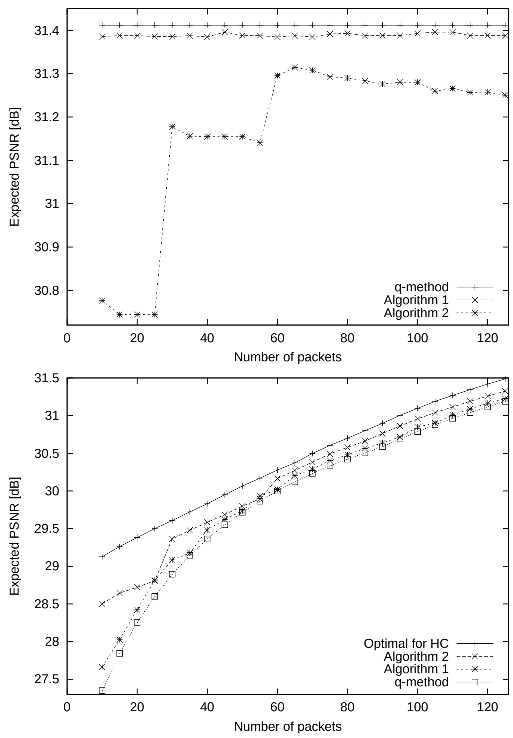

Fig. 3. Expected PSNR for the LC (top) and the HC (bottom) as a function of the number of packets in the second layer. The number of packets in the first layer is fixed to 128. Mean packet loss rates are 0.05 and 0.2 for the LC and the HC, respectively. Results are for the SPIHT bitstream of Peppers.

2, respectively, while the performance loss experienced by the LC (the optimal value was 32.88 dB) was 0 dB, 0.02 dB, and 1.62 dB, with the three methods, respectively. Thus, in this situation, only Algorithm 2 provided a solution that was acceptable to all clients.

Algorithms 1 and 2 are fast. For non-optimized implementations, the CPU time averaged over all N_2 values was 0.2 s on a PC having an AMD Athlon XP 2400 MHz processor.

V. CONCLUSION

We proposed two fast algorithms for constructing two-layer multiple description codes for multicast and broadcast applications. Our codes provided a better quality trade-off among the clients than the best previous solution of [5]. With our second algorithm and except for some extreme cases, the quality loss for all clients was less than 0.7 dB. This answers the open question of [5] as to whether two-layer multiple description codes with less than 1 dB penalty could be designed. Future work will include an efficient extension of our algorithms to more than two layers.

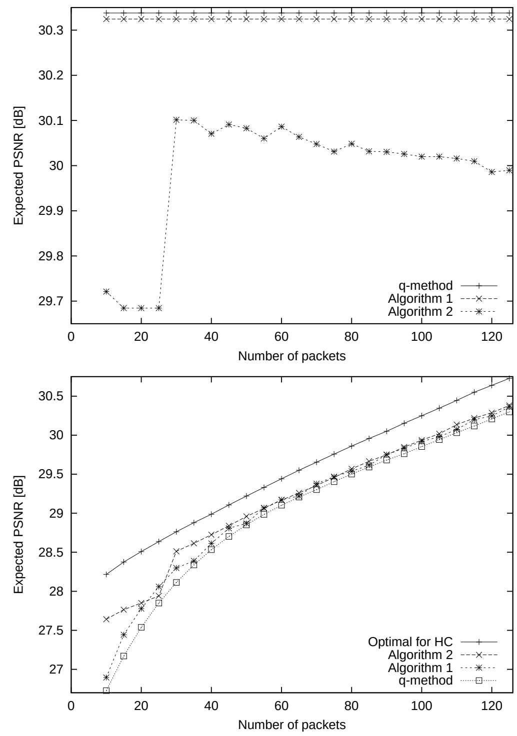

Fig. 4. Expected Y-PSNR for the LC (top) and the HC (bottom) as a function of the number of packets in the second layer. The number of packets in the first layer is fixed to 128. Mean packet loss rates are 0.05 and 0.2 for the LC and the HC, respectively. Results are for the 3D-SPIHT bitstream of the first group of frames (of size 16) of the QCIF Foreman sequence.

REFERENCES

[1] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan, “Priority encoding transmission,” *IEEE Trans. Inform. Theory*, vol. 42, pp. 1737–1744, Nov. 1996.

[2] A.E. Mohr, R.E. Ladner, and E.A. Riskin, “Approximately optimal assignment for unequal loss protection,” *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 367–370, Vancouver, Sept. 2000.

[3] R. Puri and K. Ramchandran, “Multiple description coding using forward error correction codes,” *Proc. 33rd Asilomar Conf. Signals and Systems*, vol. 1, pp. 342–346, Pacific Grove, CA, Oct. 1999.

[4] P.A. Chou, A. Mohr, A. Wang, and S. Mehrota, “Error control for receiver-driven layered multicast of audio and video,” *IEEE Trans. Multimedia*, vol. 3, pp. 108–122, March 2001.

[5] P. A. Chou, H. J. Wang, and V. N. Padmanabhan, “Layered multiple description coding,” *Proc. PV 13th Int. Packet Video Workshop*, Nantes, France, April 2003.

[6] S. Dumitrescu, X. Wu, and Z. Wang, “Globally optimal uneven error-protected packetization of scalable code streams,” *IEEE Trans. Multimedia*, vol. 6, pp. 230–239, April 2004.

[7] V. Stanković, R. Hamzaoui, and Z. Xiong, “Real-time error protection of embedded codes for packet erasure and fading channels,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, pp. 1064–1072, Aug. 2004.

[8] V. Stanković and Z. Xiong, “Packet erasure protection for multicasting,” *Proc. Data Compression Conference*, pp. 142–151, Snowbird, UT, March 2004.

[9] A. Said and W. A. Pearlman, “A new fast and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 243–250, June 1996.

[10] B. Kim, Z. Xiong, and W.A. Pearlman, “Low bit-rate scalable video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT),” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 692–695, Dec. 2000.