FEC and Pseudo-ARQ for Receiver-driven Layered Multicast of Audio and Video

Philip A. Chou[1], Alex Mohr[2], Albert Wang[1], and Sanjeev Mehrotra[1,3]
[1]*Microsoft Corporation, One Microsoft Way, Redmond, WA 98052-6399*
[2]*Dept. of Computer Science, University of Washington, Seattle, WA 98195-2350* *
[3]*Dept. of Electrical Engineering, Stanford University, Stanford, CA 98304*
{pachou,alwang,sanjeevm}@microsoft.com, amohr@u.washington.edu

Abstract

We consider the problem of joint source/channel coding of real-time sources, such as audio and video, for the purpose of multicasting over the Internet. The sender injects into the network multiple source layers and multiple channel (parity) layers, some of which are delayed relative to the source. Each receiver subscribes to the number of source layers and the number of channel layers that optimizes the source-channel rate allocation for that receiver's available bandwidth and packet loss probability. We augment this layered FEC system with layered ARQ. Although feedback is normally problematic in broadcast situations, ARQ is simulated by having the receivers subscribe and unsubscribe to the delayed channel coding layers to receive missing information. This pseudo-ARQ scheme avoids an implosion of repeat requests at the sender, and is scalable to an unlimited number of receivers. We show gains of up to 18 dB on channels with 20% loss over systems without error control, and additional gains of up to 13 dB when FEC is augmented by pseudo-ARQ in a hybrid system. The hybrid system is controlled by an optimal policy for a Markov decision process.

1 Introduction

This paper addresses the problem of *broadcasting* audio and video efficiently and robustly over the Internet, to potentially millions of simultaneous receivers. Coding for broadcast in the Internet is difficult because of the Internet's extreme heterogeneity. The bandwidths between a sender and its various receivers can vary by three orders of magnitude, from tens of kilobits to tens of megabits per second, across different links. Packet loss probabilities can also vary by three orders of magnitude, from hundredths to tens of percents, across different routers.

Bandwidth heterogeneity in the Internet can be dealt with using layered coding. Layered coding, also known as scalable, embedded, or progressive coding, is the basis

*Supported as an intern by Microsoft Corportion and by the National Science Foundation (Grant EIA-9973531) of the Receiver-driven Layered Multicast (RLM) scheme, pioneered by McCanne[1] and others. RLM codes the audio and video signal in multiple layers, and broadcasts (actually, multicasts) each layer to a different multicast group. Each receiver estimates its available bandwidth and joins a number of these multicast groups to fill that available bandwidth. Because each receiver determines its own transmission rate, the scheme is said to be receiver-driven.

Packet loss heterogeneity in the Internet can also be dealt with using layered coding. In this paper, we study a system that uses layered channel coding as well as layered source coding to achieve both flow control and error control in a unified receiver-driven framework. The layered source coding is achieved with any of the usual techniques, while the layered channel coding is achieved using a systematic rate-compatible Reed-Solomon style erasure code. The sender multicasts all the source coding layers and all the channel coding (i.e., parity) layers to separate multicast groups. Each receiver estimates the bandwidth and packet loss probability of its channel, computes the optimal allocation of the available transmission rate between the source and channel codes (which generally results in unequal error protection for the different source layers), and joins the multicast groups for the optimal collection of source and channel layers. We then hybridized this system with a pseudo-ARQ system, in which ARQ is simulated by the sender continuously transmitting delayed parity packets to additional multicast groups. The receivers can join and leave these multicast groups to retrieve information lost in previous transmissions, up to a given delay bound. The optimal algorithm for a receiver joining and leaving multicast groups is equivalent to the optimal policy in a finite horizon Markov decision process, which contains the optimal allocation for pure FEC as a special case. To the sender, pseudo-ARQ looks like ordinary multicast, while to the receiver it looks like ordinary ARQ. Thus pseudo-ARQ uses the existing multicast protocols to avoid the problem of the repeat requests (negative acknowledgments, or NAKs) imploding upon the sender or upon other designated retransmitters. The system is completely scalable, to potentially millions of receivers, with no explicit retransmitters or additional servers.

For channels with 20% ambient packet loss, compared to standard RLM (without error control), receiver-driven layered FEC gains up to 18 dB, and receiver-driven layered hybrid FEC/Pseudo-ARQ gains up to an additional 13 dB (for a total of up to 31 dB). As astoundingly high as these gains are, they can be had for free (i.e., with no change in network bandwidth usage) if the routers simply dropped packets in order of priority, rather than relying on explicit application-level error control mechanisms, which introduce delay and consume bandwidth that could otherwise be used for coding the source. This indicates that next generation real-time multimedia protocols could greatly benefit multimedia delivery by supporting prioritized routing.

Our proposed system can be regarded in various ways. Those familiar with RLM can regard our system as an error control extension to RLM. Those familiar with joint source-channel coding (JSCC) — in particular layered source coding with unequal error protection — can regard our system as adaptive JSCC with receiver feedback. Those familiar with hybrid FEC/ARQ can regard our system as an extension of hybrid FEC/ARQ to layered coding. Those familiar with reliable multicast techniques can regard our system as an adaptation of some of these techniques to real-time multicast.

Given the many interpretations of our work, the amount of related previous work is far too great to cover here. We cite here only the paper by Chande, Jafarkhani, and Farvardin [2], to which we owe particular debt for showing the connection between Markov decision processes, and control for joint source/channel coding with feedback. For a more complete discussion of related previous work, see [3]. In contemporaneous independent work, Tan and Zakhor developed a system very similar to ours. They first presented their system in [4]; we first presented our system in [5]. We focus more on optimal rate allocation for FEC and optimal control for hybrid FEC/ARQ. They focus more on adaptive delay for FEC, and on implementation. We believe these works to be mutually supportive.

2 System Description

2.1 Source coding and packetization

For the purposes of this paper, we assume that an embedded source encoder is available, such as 3D SPIHT [6] for video or 1D SPIHT [7] for audio. We produce a set of elementary bit streams corresponding to different layers by blocking the audio or video sequence into groups of frames (GOFs), independently encoding each GOF as an embedded bit string, packetizing each embedded bit string sequentially into fixed-length packets, and concatenating corresponding packets (from different GOFs) to form the elementary bit streams each having a bit rate of one packet per GOF.

It is commonly assumed that when an embedded bit string is sequentially packetized, each packet depends on the previous packet, in the sense that before decoding a packet (except for the first packet) it is necessary to decode the previous packet. However, such a sequential dependency structure is usually neither necessary nor advantageous. We assume, more generally, that the dependency structure can be represented by a directed acyclic graph (DAG), along with the quantity $\Delta D_l \geq 0$ at each node l of the graph, which represents the expected decrease in distortion (per group of frames) if packet l is usefully decoded. We write $l' \prec l$ if l' is an ancestor of l in the DAG.

2.2 FEC coding and packetization

Channel coding for the elementary source streams is performed as follows. Each elementary source stream is partitioned into coding blocks having K source packets per coding block. The blocksize K is constant across all elementary source streams, and in fact the block boundaries are synchronized across the elementary source streams so that K represents the overall coding delay. For example, $K = 8$ implies a coding delay of 8 seconds, if a group of frames has a duration of 1 second.

For each block of K source packets in an elementary source stream, $N - K$ parity packets are produced using a systematic (N, K) Reed-Solomon style erasure correction code. The “codelength” N is determined by the maximum amount of redundancy that will be needed by any receiver to protect the elementary source stream. (N may

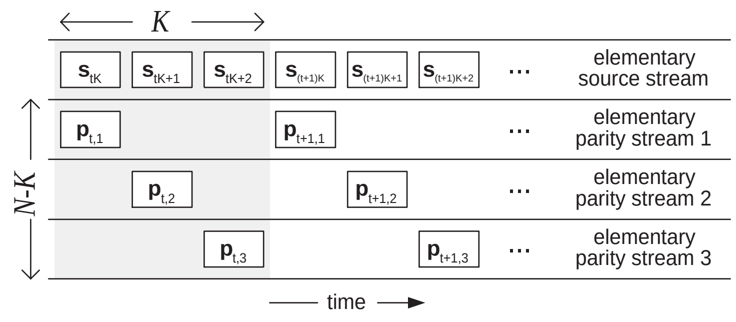

Figure 1: Generation of parity streams from each source stream.

vary across elementary source streams.) The $N-K$ parity packets are generated bytewise from the K source packets, using for each byte the generator matrix from a systematic Reed-Solomon style code over the finite Galois field $GF(2^8)$.

Each of the parity packets so generated is placed in its own multicast stream, so that now each elementary source stream (at rate 1 packet per GOF) is accompanied by $N-K$ elementary parity streams, each at rate $1/K$ packets per GOF, as illustrated in Figure 1. Thus each source layer is now represented by one elementary source stream and a number of elementary parity streams.

2.3 Optimal rate allocation

A receiver now has many streams to which it can subscribe: it can subscribe to any collection of elementary source streams and any collection of elementary parity streams associated with those source streams. In this section, we solve the problem of finding the collection of elementary source and parity streams to which the receiver should subscribe in order to minimize its expected reconstruction error, given that the receiver's channel from the sender has a fixed packet loss probability and maximum transmission rate.

We shall assume that all packets transmitted within the time period of a K-GOF code block are lost independently with some probability ϵ. It is the responsibility of the receiver to update its estimate of ϵ for each code block, as well as to update its estimate of the available transmission rate.

Let L be the number of source layers, K be the number of source packets per code block per layer, and $N-K$ be the number of parity packets per code block per layer. Let N_l be the number of source plus parity packets in layer l to which the receiver subscribes. Then $\rho_l = N_l/K$ is the rate, in number of packets per GOF, transmitted to the receiver in layer l.

The vector $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_L)$ is called the *rate allocation*. Any given rate allocation induces a total rate and an expected distortion. For a given rate allocation, the total rate, in terms of transmitted packets per GOF, is given by

$$R(\boldsymbol{\rho}) = \sum_l \rho_l. \tag{1}$$

This is just the total number of packets in a block to which the receiver subscribes divided by K.

The expected distortion for a given rate allocation is somewhat more complicated to express. For each source packet $k = 1, \ldots, K$ in each level l to which the receiver subscribes let $I_k^{(l)}$ be the indicator random variable that is 1 if and only if the receiver can recover the source packet after channel decoding. Then the probability that the receiver can recover the source packet after channel decoding is given by $EI_k^{(l)}$. Actually this quantity does not depend on k, because by symmetry, the random variables $I_k^{(l)}$, $k = 1, \ldots, K$, are exchangeable. Hence

$$EI_k^{(l)} = \frac{1}{K} E\left[\sum_k I_k^{(l)}\right] = \frac{1}{K} N(N_l, K) = P\{I_k^{(l)} = 1\},$$

where $N(N_l, K)$ is the expected number of source packets that can be recovered after channel decoding with a (N_l, K) code. If the source and channel packet losses are independent with probability ϵ, then when $N_l \geq K$,

$$N(N_l, K) = \sum_{s=0}^{K} \binom{K}{s} \epsilon^{K-s}(1-\epsilon)^s \sum_{c=0}^{N_l-K} \binom{N_l - K}{c} \epsilon^{N_l-K-c}(1-\epsilon)^c \times \begin{cases} K & \text{if } s+c \geq K \\ k & \text{if } s+c < K \end{cases}.$$

When $N_l = 0$ (i.e., when the receiver does not subscribe to layer l), then $N(N_l, K) = 0$. Now, the kth source packet in layer l can be fully decoded if and only if it and all of its ancestors in the dependency graph can be recovered after channel decoding, i.e., if and only if the product $\prod_{l' \preceq l} I_k^{(l')}$ equals 1. Since the $I_k^{(l)}$ are independent, $E \prod_{l' \preceq l} I_k^{(l')} = \prod_{l' \preceq l} EI_k^{(l')}$ and the expected reduction in distortion by subscribing to layer l at rate ρ_l is $\prod_{l' \preceq l} EI_k^{(l')} \times \Delta D_l$. Hence the expected distortion (per GOF) for the given rate allocation $\boldsymbol{\rho}$ is

$$D(\boldsymbol{\rho}) = D_0 - \sum_l \left(\prod_{l' \preceq l} (1 - \epsilon(\rho_{l'})) \right) \Delta D_l, \tag{2}$$

where $\epsilon(\rho_l) = 1 - EI_k^{(l)} = (1 - N(K\rho_l, K)/K) = P\{I_k^{(l)} = 0\}$ is the residual probability of packet loss after channel decoding.

With expressions (1) and (2) for the rate and expected distortion for any given rate allocation now in hand, we are able to optimize the rate allocation to minimize the expected distortion subject to a rate constraint. By restricting ourselves to solutions on the lower convex hull of the set of rate-distortion pairs $\{(R(\boldsymbol{\rho}), D(\boldsymbol{\rho})\}$, we can solve the problem by finding the rate allocation $\boldsymbol{\rho}$ that minimizes the Lagrangian

$$J(\boldsymbol{\rho}) = D(\boldsymbol{\rho}) + \lambda R(\boldsymbol{\rho}) = D_0 + \sum_l \left[\left(- \prod_{l' \preceq l} (1 - \epsilon(\rho_{l'})) \right) \Delta D_l + \lambda \rho_l \right]. \tag{3}$$

The solution to this problem is completely characterized by the set of distortion increments ΔD_l (which are determined by the source, source code, and source packetization), and the residual loss probability function $\epsilon(\rho)$ (which is determined by the channel, channel code, and channel packetization). This simplifies the problem of determining the quantities needed for the optimization. However, the minimization itself is complicated by the fact that the expression for the distortion cannot be split into a sum of terms each depending on only a single ρ_l, as is usually the case in bit allocation problems. By approximating the Lagrangian, the optimal rate allocation was solved (approximately) by [8]. Quite recently, Chande and Farvardin showed how to solve the problem exactly, using dynamic programming [9]. Here, we solve the problem exactly using an iterative algorithm. The advantage to the iterative algorithm is that it generalizes to the feedback case, which we consider in Section 2.5.

Our iterative approach is the following. Consider the distortion-rate function $D(\boldsymbol{\rho})$ in (2). Begin with an initial rate allocation $\boldsymbol{\rho}^{(t)}$, $t = 0$. Linearize $D(\boldsymbol{\rho})$ around $\boldsymbol{\rho}^{(t)}$ in the variables $\epsilon_l = \epsilon(\rho_l)$, $l = 1, \ldots, L$, using the first two terms in the Taylor series expansion of (2) around $\epsilon_l^{(t)} = \epsilon(\rho_l^{(t)})$:

$$D(\boldsymbol{\rho}) = D(\boldsymbol{\rho}^{(t)}) + \sum_l S_l^{(t)} \epsilon_l + o\left(||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^{(t)}||^2\right), \tag{4}$$

where $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_L)$, $\boldsymbol{\epsilon}^{(t)} = (\epsilon_1^{(t)}, \ldots, \epsilon_L^{(t)})$, and $S_l^{(t)}$ is the partial derivative of (2) with respect to ϵ_l evaluated at $\boldsymbol{\epsilon}^{(t)}$. It is straightforward to show that

$$S_l^{(t)} \triangleq \frac{\partial}{\partial \epsilon_l} \left[D_0 - \sum_{l'} \left(\prod_{l'' \preceq l'} (1 - \epsilon_{l''}) \right) \Delta D_{l'} \right]_{\boldsymbol{\epsilon}^{(t)}} = \sum_{l' \succeq l} \prod_{\substack{l'' \preceq l' \\ l'' \neq l}} (1 - \epsilon_{l''}^{(t)}) \Delta D_{l'}.$$

The factors S_l can be regarded as the sensitivity to losing a packet in layer l, i.e., the amount by which the distortion will increase when a packet in layer l is lost, given the current amount of erasure protection on the other layers. Now we use the first two terms of (4) in (3) and minimize the approximate Lagrangian

$$\tilde{J}(\boldsymbol{\rho}) = D(\boldsymbol{\rho}^{(t)}) + \sum_l \left[S_l^{(t)} \epsilon(\rho_l) + \lambda \rho_l \right].$$

This quantity is simple to minimize since each term in the sum depends only on one ρ_l. In fact, the solution to this problem is to set each ρ_l equal to the rate at which the line at slope $-\lambda$ supports the graph of $S_l^{(t)} \epsilon(\rho)$. This produces a new rate allocation $\boldsymbol{\rho}^{(t+1)}$, and the process is repeated. Assuming the process converges with $\boldsymbol{\rho}^{(t)} \to \boldsymbol{\rho}^*$ (which we have always observed in practice) the resulting solution $\boldsymbol{\rho}^*$ is optimal, because (4) is exact at $\boldsymbol{\rho} = \boldsymbol{\rho}^{(t)}$. By adjusting λ, the overall rate constraint can be met.

The resulting rate allocation $\rho_1^*, \ldots, \rho_L^*$ in general will not be equal across layers, because the packet loss sensitivities $S_1, \ldots, S_L$ in general are not equal across layers. Thus the layers are provided with unequal amounts of protection.

2.4 Pseudo-ARQ

Even with unequal error protection, FEC does not achieve the capacity of the packet erasure channel, except in the limit of large blocksizes. ARQ, on the other hand,

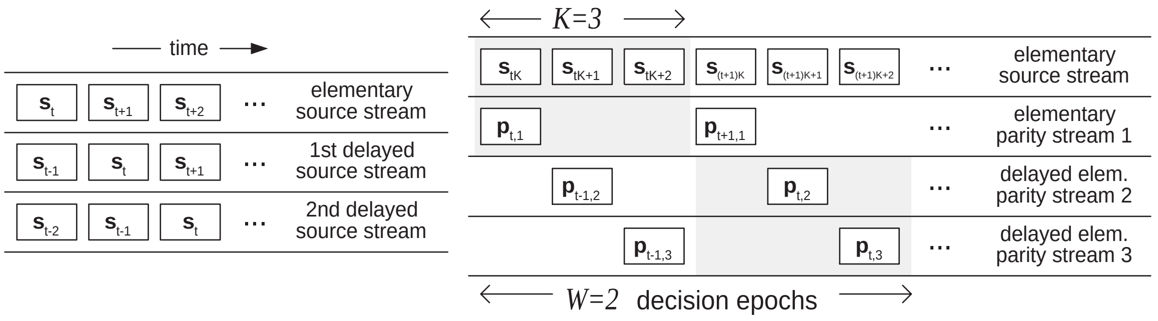
time
elementary source stream
1st delayed source stream
2nd delayed source stream
K=3
elementary source stream
elementary parity stream 1
delayed elem. parity stream 2
delayed elem. parity stream 3
W=2 decision epochs

Figure 2: (a) Pseudo-ARQ and (b) hybrid FEC/Pseudo-ARQ for a single source stream.

makes optimal use of the forward channel by transmitting only as many redundant packets as lost packets. In addition, it adapts automatically to the packet loss probability. For these reasons, ARQ is used extensively for data transmission and even for real-time media transmission, such as video-on-demand. But in the broadcast case, ARQ is usually regarded as impractical because of the NAK implosion problem. However, we observe that for broadcast of real-time media, the delay is bounded, so that the number of repeat-requests has to be limited to at most a few. In addition we observe that for very large numbers of receivers, each packet will be lost by at least one receiver. With these observations, it makes sense for the sender to repeatedly transmit all of the packets all of the time, up to the delay bound.

This leads to the idea of Pseudo-ARQ, illustrated in Figure 2a, in which the sender transmits delayed versions of the source to different multicast groups. If a receiver loses a packet, it can obtain a repeated packet by joining (and soon leaving) the multicast group to which the delayed version of the source is transmitted. If it loses the repeated packet, then it can obtain a re-repeated packet by joining and leaving the multicast group to which a further delayed version of the source is transmitted, and so on, until the packet is received or the delay bound is reached. The number of delayed versions multicast by the sender is therefore proportional to the delay bound.

Because the transmission rate for multiple copies of the source can be excessively high, we actually advocate a hybrid of FEC and Pseudo-ARQ techniques, in which the sender delays the parity packets rather than the source packets themselves, so that the "repeated" information is actually parity information. Figure 2b illustrates hybrid FEC/Pseudo-ARQ for one source layer, blocked as before for $K = 3$ source packets, each block producing $N - K = 3$ parity packets, two of which are delayed by one code block into a second "epoch" of parity packets for the block. The coding delay for this system is the blocklength (3) times the number of epochs (2), or 6 in this case. Hybridizing FEC with Pseudo-ARQ reduces network traffic compared to pure Pseudo-ARQ because a single parity packet can be used to respond to a NAK for any packet in the block. Thus network traffic is reduced, and scalability is enhanced, in the hybrid system.

2.5 Optimal error control

In hybrid FEC/Pseudo-ARQ, the control process at each receiver for joining and leaving the various multicast groups (as a function of which packets have been received) is stochastic, depending on the packet loss process. Indeed, this stochastic control process can be regarded as a finite horizon Markov decision process.

A Markov decision process with finite horizon W is a W-step finite-state stochastic process in which a decision can be made at each step, to influence the transition probabilities, in order to maximize an expected reward or minimize an expected cost. A sequence of decision rules is called a *policy*, and the optimal policy, in our context, is the one that minimizes the expected cost $D + \lambda R$ of traversing the state space.

Let s be the number of source packets received, let c be the number of parity packets received, and let t be the number of total (source plus parity) packets transmitted (not all of which are received). These quantities define the state of the Markov decision process for a given code block of a given source layer l. Let w be the decision epoch. Initially, $s = c = t = w = 0$, and the expected rate and (change in) distortion for the code block are each zero. At the beginning of each decision epoch $w = 0, \ldots, W-1$, the receiver may request the server to transmit a packets. In the multicast scenario, the receiver accomplishes this by subscribing to the appropriate number of elementary streams during the decision epoch. The value of a may take on any value in an alphabet A_w, possibly depending on the epoch. A decision rule d_w is a deterministic mapping $d_w(s,c,t)$ from states at epoch w into A_w; a policy π is a sequence of decision rules $d_0, d_1, \ldots, d_{W-1}$ [10]. We seek the optimal policy π_l for each layer l that minimizes

$$J(\boldsymbol{\pi}) = D(\boldsymbol{\pi}) + \lambda R(\boldsymbol{\pi}) = D_0 + \sum_l \left[\left(- \prod_{l' \preceq l} (1 - \epsilon(\pi_{l'})) \right) \Delta D_l + \lambda R(\pi_l) \right]. \qquad (5)$$

Here, $\epsilon(\pi_l)$ is the probability that a particular packet is not recovered in the code block for layer l (after up to W retransmissions) under error control policy π_l, and $R(\pi_l)$ is the expected number of packets transmitted for the code block under the policy (normalized by the blocklength K). As in Section 2.3, (5) can be solved by starting with an initial "policy allocation" $\boldsymbol{\pi}^{(t)}$, $t = 0$, and iteratively minimizing the Taylor series approximation

$$\tilde{J}(\boldsymbol{\pi}) = D(\boldsymbol{\pi}^{(t)}) + \sum_l \left[S_l^{(t)} \epsilon(\pi_l) + \lambda R(\pi_l) \right], \qquad (6)$$

where $S_l = S_l^{(t)}$ is the sensitivity of losing a packet in layer l under the policy allocation $\boldsymbol{\pi}^{(t)}$. Clearly, (6) can be minimized componentwise, by finding the policy π_l that minimizes the term $S_l \epsilon(\pi_l) + \lambda R(\pi_l)$. This can be solved with a dynamic programming algorithm. For details, see [3].

3 Results

We now present the results of our analyses. For the analyses in this paper, we model the source as having an operational distortion-rate function $D(R) = A 2^{-2R}$ for some

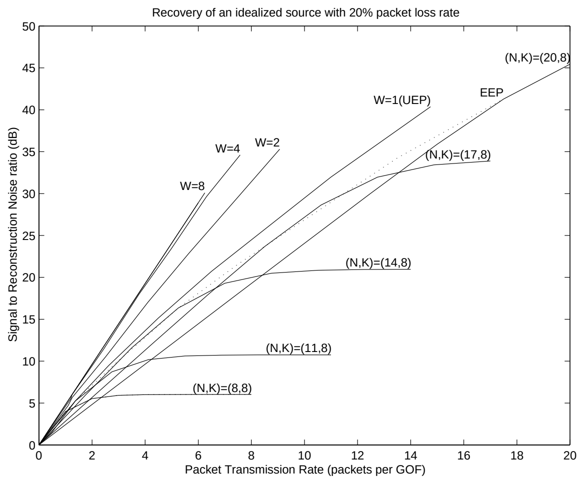

Figure 3: Analytical results.

constant A and sequential packetization dependencies; we model the channel as having independent 20% packet loss; and we model the network as having no delay, jitter, resequencing, or join and leave latencies. We concentrate on a single receiver, and do no analysis of aggregrate receiver behavior. Thus in this paper we do no actual source coding, channel coding, or network transmission.

Figure 3 shows the signal-to-noise ratio of the end-to-end reconstruction error as a function of the total packet transmission rate in packets per GOF. With original RLM, which has no redundancy ($(N, K) = (8, 8)$), as the receiver subscribes to more layers its performance saturates, because with high probability there is a loss within the first few layers, rendering the subsequent layers useless. With a fixed amount of FEC determined by the sender ($(N, K) = (11, 8)$, $(14, 8)$, $(17, 8)$, and $(20, 8)$), performance still eventually saturates for a similar reason, although at a higher level. Perhaps more importantly, there is also likely to be a penalty due to channel mismatch if the amount of FEC is sender-driven, particularly if the packet loss rate is underestimated. Receiver-driven equal error protection can achieve the convex hull (dotted line) of the sender-driven equal error protection schemes, with no channel mismatch. This is because for each transmission rate and packet loss rate, the receiver can choose the optimal amount of redundancy to be applied equally to all layers to which it subscribes. Of course, receiver-driven unequal error protection ($W = 1$, $K = 8$) can perform still better, by up to 3 dB. However, when this is combined with Pseudo-ARQ ($(W, K) = (2, 4)$, $(4, 2)$, and $(8, 1)$), even when the delay is held constant, performance quickly approaches the optimal performance (the operational distortion-rate function evaluated at the channel capacity) as the number of epochs increases. Indeed, the performance of Pseudo-ARQ for $W = 8$ is indistinguishable from $D((1 - \epsilon)R)$, where $(1 - \epsilon)R$ is the information-theoretic capacity (in packets per GOF) of the channel with transmission rate R and independent packet loss probability ϵ.

4 Conclusion

In summary, we have presented a FEC and hybrid FEC/Pseudo-ARQ error control architecture for use in receiver-driven layered multicast systems. Pseudo-ARQ uses only existing multicast mechanisms to process the repeat requests, and hence is scalable to an unlimited number of receivers without incurring any repeat request implosions. We also presented algorithms for optimizing each receiver's reconstruction quality at a given transmission rate and packet loss probability. For the hybrid FEC/Pseudo-ARQ scheme, this optimization involves finding the optimal policy for a finite horizon Markov decision process. Analytical results show that on a channel with 20% packet loss, receiver-optimized FEC with moderate delay can gain up to 18 dB over systems without explicit error control, and receiver-optimized Pseudo-ARQ can gain up to an additional 13 dB.

References

[1] S. R. McCanne. *Scalable Compression and Transmission of Internet Multicast Video.* PhD thesis, The University of California, Berkeley, CA, December 1996.

[2] V. Chande, H. Jafarkhani, and N. Farvardin. Joint source-channel coding of images for channels with feedback. In *Proc. Information Theory Workshop*, San Diego, CA, February 1998. IEEE.

[3] P. A. Chou, A. E. Mohr, A. Wang, and S. Mehrotra. FEC and pseudo-ARQ for receiver-driven layered multicast of audio and video. Technical Report MSR-TR-99-86, Microsoft Research, Redmond, WA, November 1999.

[4] W.-T. Tan and A. Zakhor. Multicast transmission of scalable video using receiver-driven hierarchical FEC. In *Proc. Packet Video Workshop*, New York, April 1999.

[5] P. A. Chou, A. E. Mohr, A. Wang, and S. Mehrotra. FEC and pseudo-ARQ for receiver-driven layered multicast. In *Proc. Communication Theory Workshop*, Aptos, CA, May 1999. IEEE.

[6] B.-J. Kim and W. A. Pearlman. An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees (SPIHT). In *Proc. Data Compression Conference*, pages 251–260, Snowbird, UT, March 1997. IEEE Computer Society.

[7] Z. Lu and W. A. Pearlman. An efficient, low-complexity audio coder delivering multiple levels of quality for interactive applications. In *Proc. Workshop on Multimedia Signal Processing*, pages 529–534, Redondo Beach, CA, December 1998. IEEE.

[8] J. Lu, A. Nosratinia, and B. Aazhang. Progressive source-channel coding of images over bursty channels. In *Proc. Int'l Conf. Image Processing*, Chicago, IL, October 1998. IEEE.

[9] V. Chande and N. Farvardin. Progressive transmission of images over bit error and packet erasure channels. 1999. Submitted.

[10] M. L. Puterman. *Markov Decision Processes.* Wiley, 1994.