Video Multicast using Layered FEC and Scalable Compression

Wai-tian Tan, Avideh Zakhor

Department of Electrical Engineering and Computer Sciences

University of California, Berkeley, CA 94720

e-mail: {*dtan avz*}*@eecs.berkeley.edu*

Abstract

The use of scalable video with layered multicast has been shown to be an effective method to achieve rate control in heterogeneous networks. In this paper, we propose the use of layered FEC as an error control mechanism in a layered multicast framework. By organizing FEC into multiple layers, receivers can obtain different levels of protection commensurate with their respective channel conditions. Efficient network utilization is achieved as FEC streams are multicast, and only to receivers that need them. Furthermore, FEC is used without overall rate expansion by selectively dropping data layers to make room for FEC layers. Effects of bursty losses are amortized by staggering the FEC streams in time, giving rise to a trade-off between delay and quality. For rate control at the receivers, we propose an equation-based approach that computes network usage as a function of measured network characteristics. We show that equation-based rate control achieves more fair bandwidth sharing amongst competing sessions as compared to existing multicast rate control schemes such as RLM and RLC. Fairness is achieved since competing sessions sharing a path will measure similar network characteristics. Simulations and actual MBONE experiments are performed using error-resilient, scalable video compression. We find that video quality is significantly improved at the same communication rate when layered FEC is used.

1 Introduction

The deployment of IP multicast in the Internet Multicast Backbone (MBONE) has provided a convenient and efficient way of disseminating video information to multiple recipients over the Internet. Similar to the unicast Internet, the MBONE is a best-effort network which requires rate and error control mechanisms for video communications. However, the implementation of such rate and error control functions are considerably more challenging for multicast than unicast. This is because in a unicast setting, the needs of only a single receiver have to be addressed whereas in a multicast setting, it becomes necessary to simultaneously cater to the potentially conflicting needs of multiple recipients.

In the unicast Internet, flow control is typically implemented by adapting the communication rate at the source in reaction to feedback information from the receiver [1]. This source-based adaptation approach is widely used for unicast data communications, and is the basis of many unicast [2, 3, 4] and some multicast [5, 6] video transmission schemes. However, by sending the same single stream to all receivers, multicast source-based adaptation schemes cannot accommodate the heterogeneity of channel conditions and user preferences in multicast. Furthermore, by requiring feedback information from all receivers, source-based adaptation schemes also suffer from the feedback implosion problem, which occurs when the source is overwhelmed by feedback messages from the receivers in a large multicast.

One possible approach to overcome the short-comings of the source-based adaptation scheme is to transmit replicated copies of the same video corresponding to different communication rates using different multicast groups [7]. In such a way, the bandwidth heterogeneity problem is better addressed since each receiver can choose, from the available multicast groups, the single copy whose rate best matches its experienced channel conditions. However, since transmitting multiple copies of the same source consumes more network bandwidth, the actual number of copies that can be effectively sent is usually limited. Further improvements of the replicated stream approach includes

adaptively modifying the rates of each stream [7] at the cost of additional complexity due to the need for feedback, and thus the possibility of feedback implosion [8].

A more flexible alternative to the multicast flow control problem is offered by the layered multicast framework for scalable video [9, 10, 11]. Scalable video compression produces an embedded bit-stream that allows decoding at multiple rates [12, 13, 14, 4]. Under layered multicast, different layers of a scalable video are carried in different multicast groups so that receivers can individually subscribe or unsubscribe to the appropriate multicast groups to achieve flow control [11, 9, 15, 16, 17, 18]; thus, the more layers a receiver subscribes to, the higher the quality of the received video. Generally, producing an embedded bit-stream for scalable compression results in lower compression efficiency than its non-scalable counterpart at the same bit rates [19]. Nevertheless, layered multicast is often preferred to schemes based on source adaptation or stream replication due to its ability to effectively address bandwidth heterogeneity.

On the error control side, a large body of work has focused on providing partial or full reliability to all receivers using retransmissions [20, 21] or type-II hybrid-ARQ [22] schemes employing both error control codes across packets and retransmissions [23, 24, 25, 26]. Since source-based retransmission schemes are well known to be non-scalable to large groups of receivers, elaborate schemes have been employed to organize the receivers into repair trees to effect hierarchical retransmission [21, 25, 27], or local groups to facilitate local repair [20]. However, the effectiveness of such schemes is undermined by the lack of explicit topological information under IP-multicast. As a result, there is no explicit basis to determine how receivers should be organized into repair trees or groups [28]. Consequently, complex and inaccurate methods have been used to infer the network topology [20, 27]. Alternatively, manual building of repair trees [21, 29] and manual configuration of networks [25] have been proposed, with the drawback of requiring *a priori* knowledge of the set of receivers, the network topology, and requiring administrative privileges of all participating networks.

Due to high cost and complexity, multicast retransmissions are often justified only for low band-

width applications that require reliability, such as the shared white-board in the MBONE tools [20] or financial data distribution. Retransmissions are inappropriate for video multicast applications due to the higher communication rates and packet loss tolerance of video. In contrast, pure forward error correction (FEC) based approaches to error control can be readily realized under IP multicast. For instance, the Digital Fountain approach for reliable, non real-time multicast of data is based only on transmitting the original and redundant data in a "carousel" manner [30].

In this paper, we will investigate the use of an FEC based error control mechanism for video multicast which we will refer to as *layered FEC*. An overview of the basic layered FEC scheme is presented in Section 2. The optimization procedure for each receiver is discussed in Section 3. In Section 4 some possible approaches to organize FEC in a layered manner are discussed. To evaluate the layered FEC scheme in a setting where multiple concurrent sessions are present, a rate control mechanism that address inter-session fairness is needed. To this end, Section 5 discusses the use of an equation-based rate controller and evaluates its fairness properties via simulations. Simulation and experimental results for video transmissions are presented in Section 6, followed by conclusions in Section 7.

2 Overview of the Layered FEC Scheme

There are three objectives for the layered FEC scheme. First, it has to be compatible with the layered multicast model to allow effective rate control. Second, to account for the heterogeneity of packet loss rates in multicast, it needs to provide different levels of protection to different receivers. Third, it needs to cater to the different latency preferences of different users. For example, in a live multicast of a lecture, participants who want to ask questions and interact with the lecturer have a stronger preference for low latency rather than high picture quality. On the other hand, passive viewers may be willing to sacrifice latency for higher video quality. We shall classify the viewers according to their tolerance to latency and refer to the former as *low-latency viewers*, and the latter as *high-latency viewers*.

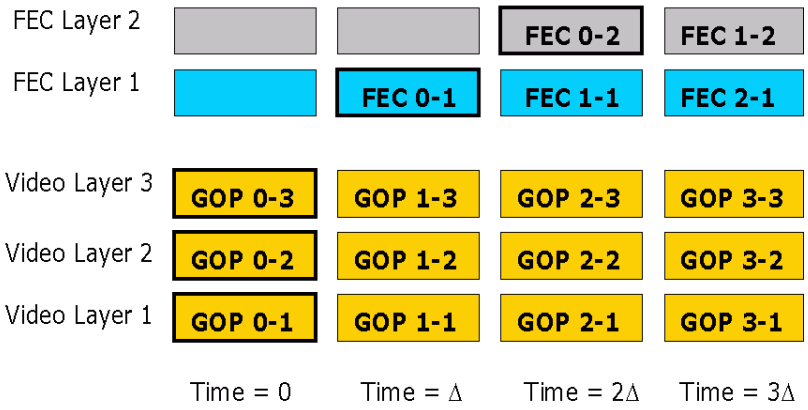


Figure 1: *Timing diagram of layered FEC scheme.*

The basic mechanism of the layered FEC scheme is illustrated in Fig.1. Video frames are blocked into groups of pictures (GOP) that are compressed together using scalable video compression techniques [4, 12, 13, 14]. In Fig. 1, GOP 0 is compressed into 3 video or data layers, GOP-0-1 to GOP-0-3, with more layers corresponding to higher picture quality. In addition, FEC layers are constructed to protect against packet losses, with more FEC layers corresponding to higher level of protection. In Fig. 1, two FEC layers, FEC-0-1 and FEC-0-2 are created for GOP 0. Each of the data and FEC layers is then transmitted in a different multicast group. In such a way, a *menu* of layers is created, and each receiver can individually choose from it the best set of layers that satisfies a prescribed rate constraint. The data layers corresponding to the same GOP are transmitted in the same time slot to provide low latency while the transmission of FEC layers are delayed with respect to the data layers as a counter measure for burstiness in packet losses. A related scheme has previously been suggested in [31] for packet audio where coarsely quantized copies of audio segments are staggered in time and transmitted using one multicast group to combat loss burstiness. There are two consequences to the delaying of FEC layers. First, low-latency viewers who desire real-time video cannot utilize the FEC layers. Second, subscribing to larger number of FEC layers results in larger latency, thereby constituting a trade-off between the level of protection and latency. Note that in some special cases, as in the scheme of [32] where special dependence structure exists between video frames, it is possible to use FEC for low-latency applications.

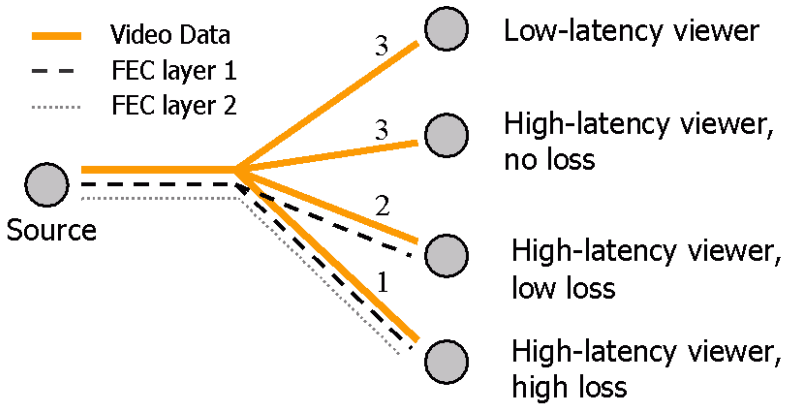


Figure 2: *Tailoring layered FEC to different receivers having the same available bandwidth. The numbers indicate the number of data layers transmitted along each link.*

Fig. 2 illustrates how the layered FEC scheme can be used to address the heterogeneity in packet loss rates and latency preferences of different receivers. Assume three data and two FEC layers of equal sizes are used and that all receivers have a bandwidth constraint of three layers. Regardless of packet loss rate, a low-latency viewer decides against the use of FEC layers to avoid the associated additional latency. In our example, it will subscribe to three data layers. A high-latency viewer, on the other hand, can take advantage of the FEC layers to obtain better video quality at the cost of higher latency. For example, while a high-latency viewer with no loss has no need for FEC and can afford to subscribe to only data layers at no extra latency, a high-latency viewer with low losses may choose to replace one data layer with an FEC layer to get better picture quality at the extra cost of one GOP delay. Similarly, a high-latency viewer with high losses can subscribe to one data and two FEC layers to compensate for the high packet losses at the extra cost of two GOP delay. We see that in all cases, the bandwidth constraints can be satisfied even when FEC is used. Furthermore, FEC packets are only transmitted to branches that lead to subscribers requiring FEC, thereby reducing overall network load. In contrast, traditional FEC schemes for multicast often compute a set of FEC packets which are multicast to all receivers regardless of whether they are needed [23, 24].

Given a menu of data and FEC layers, each receiver in a layered FEC implementation needs to

perform two basic functions. First, it needs to determine the amount of available bandwidth; this is discussed in Section 5. Second, it needs to choose from the menu the optimal set of data and FEC layers that satisfies the bandwidth constraint; this is discussed in Section 3. The related problem of actually generating the data and FEC layers is discussed in Section 4.

3 Selection of Data and FEC Layers

In this section, we consider the problem of selecting from the menu an optimal set of data and FEC layers under a prescribed packet loss rate and bandwidth constraint. Specifically, the chosen criterion for optimality is minimum distortion.

Under our layered FEC scheme, each receiver has an associated subscription level $\mathbf{s}$, which is uniquely specified by the number of subscribed data layers L, and the number of FEC packets used to protect each layer k_i, $0 \leq i < L$. An expression for distortion D can be computed as a function of the packet loss rate p and subscription $\mathbf{s}$. The exact dependence of D on p and $\mathbf{s}$ is characteristics of the specific compression technique being used. In this Section, we derive the distortion function for the error-resilient video compression technique used in our experiments.

Let $\mathbb{M}$ be the set of all possible subscriptions given by a fixed menu, and $R(\mathbf{s})$ the rate associated with subscription $\mathbf{s}$, we seek to find the optimal subscription $\mathbf{s}^* \in \mathbb{M}$ that minimizes the distortion $D(\mathbf{s}, p)$ subject to a bandwidth constraint of B:

$$\mathbf{s}^* = \underset{\mathbf{s}\in\mathbb{M},\ R(\mathbf{s})\leq B}{\arg\min}\ D(\mathbf{s}, p) \tag{1}$$

When the search space $|\mathbb{M}|$ is relatively small, $\mathbf{s}^*$ can be effectively found by exhaustive search over $\mathbb{M}$. The computational cost is then roughly the search space times the cost to evaluate $D(\mathbf{s}, p)$. Because data layer i is more important than layer j for $i < j$, in general, we can further reduce the search space by considering only the set $\mathbb{M}' \subseteq \mathbb{M}$ of subscriptions such that $k_i \geq k_j$ whenever $i < j$. For large search space, $\mathbf{s}^*$ can be found iteratively using Lagrangian type techniques [33] if some convexity assumptions are satisfied. Even though the cost of an iteration is high compared

to the cost of evaluating D, such schemes may be justifiable when the number of iterations needed is small and the search space is large.

3.1 Error-Resilient Video Compression

One important factor affecting distortion is the error-resilience properties of the video compression scheme. Generally, more error-resilient compression schemes have lower compression efficiency in loss-free environments. This means at the same communication rate, a less resilient compression scheme will have more bits available for FEC to compensate for the lack of error-resilience. While it is beyond the scope of this paper to compare various compression schemes under fixed and known loss rate and available bandwidth, it is generally agreed upon that error-resilient compression tends to work better when the channel is bursty and the loss rate is not known *a priori* [4]. There are three reasons for this. First, even with the use of FEC, video data may still be lost due to traffic burstiness, and as such, error-resilient compression can potentially result in graceful degradation. Second, it is not possible to determine the instantaneous channel conditions for the Internet. In practice, only delayed estimates of measurable quantities such as packet loss rates and round-trip times are available to video streaming applications. This can cause transient losses when network conditions change and before the application can adapt to the change. Without error-resilient compression, such losses may translate to catastrophic video quality degradations. Third, in a layered FEC framework, low-latency users do not utilize any FEC due to delay constraints, and as such, error-resilience compression offers better video quality for them under lossy conditions. The error-resilient but non-scalable scheme of [32] allows the use of FEC for low-latency applications, but is not applicable in a layered multicast environment due to the lack of scalability.

3.2 Computation of Distortion Function

In this section, we will derive the distortion function $D(\mathbf{s}, p)$ for an error-resilient and scalable video compression [4] that we use for the experiments in this paper. One characteristic of the codec is

that the bit-stream for every group of pictures that is compressed together can be separated into M components that are of similar importance and can be decoded independently. The left hand side of Fig. 3 shows the M independent, equally important components that correspond to a group of picture, and also some possible bit-rates, such as R_1 to R_4 that are available by truncating the bit-stream. One possible way to use the codec in a layered multicast setting using four layers is illustrated in the right hand side of Fig. 3. In our example, the first layer contains the bits from the bit-stream that is necessary to decode the original video at rate R_1. The bits from each component will be sent as a separate packet to give a total of M packets for the layer per GOP. The second layer contains the additional bits that is necessary to decode the original video at rate R_2. Again, one packet is generated for each component. In such a way, subscribing to one, two, three, and four layers results in total rates of R_1, R_2, R_3, and R_4 respectively.

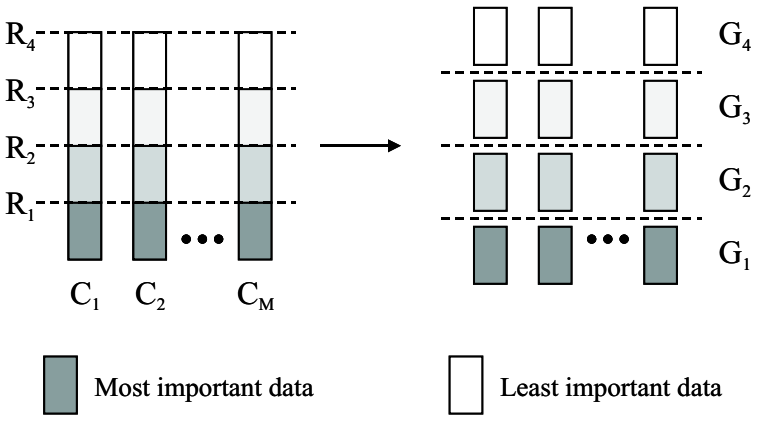
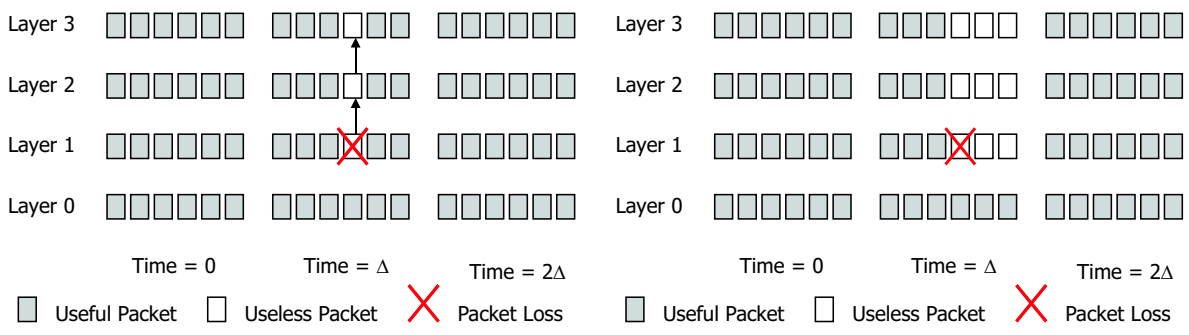


Figure 3: *Mapping scalable video into multiple multicast groups (G_1 - G_4).*

Fig. 4-(a) shows the packet loss dependence for the scheme described in Fig. 3 with four data layers, six components per GOP and a total of three GOP. Because the components can be decoded independently, a packet loss in one layer will cause error propagation into only one packet in each subsequent layers. In contrast, finely scalable but non-resilient compression schemes often fail after the first point of loss [12, 34, 14]. As a result, when a packet in one layer is lost, all subsequent packets, including some or all higher layers are rendered useless, as illustrated in Fig. 4-(b).



(a): *Error-resilient scalable video.* (b): *Non-resilient scalable video.*

Figure 4: *Data dependence for error-resilient scalable video.*

Given that every group of video frames are compressed together to give M independent, equally important components as shown in Fig. 3, the total distortion for a GOP, D, can be approximated by the sum of the distortions due to the individual components $D^{(c)}$. Let $p_i^{(c)}$ be the probability that out of a total of L subscribed layers, only the first i layers of component c are decodable and with associated distortion $D_i^{(c)}$. We have,

$$\begin{aligned} D &\approx \sum_{c=1}^{M} D^{(c)} \\ &= \sum_{c=1}^{M} \sum_{i=0}^{L-1} p_i^{(c)} \cdot D_i^{(c)} \end{aligned}$$

Since all components are assumed to be of equal importance, $D_i^{(c)}$ and $p_i^{(c)}$ are independent of c, and are equal to $D_i^{(1)}$ and $p_i^{(1)}$ respectively. Hence, we get:

$$D \approx M \sum_{i=0}^{L-1} p_i^{(1)} \cdot D_i^{(1)}$$

Defining $D_i = M D_i^{(c)}$, we get:

$$D \approx \sum_{i=0}^{L-1} p_i^{(1)} \cdot D_i \tag{2}$$

Thus, D_i is the distortion over a GOP when i layers are used to decode all components, and can be approximated using the operational rate-distortion curve.

Let $q_i^{(c)}$ be the probability that the packet in layer i of component c is lost and cannot be

recovered using FEC. Then, we have,

$$p_i^{(c)} = \begin{cases} q_i^{(c)} \prod_{k=0}^{i-1} (1 - q_k^{(c)}) & \text{if } 0 \leq i < L, \\ \prod_{k=0}^{L-1} (1 - q_k^{(c)}) & \text{if } i = L. \end{cases} \tag{3}$$

Using equal packet sizes for data and FEC layers, and an FEC code with maximum error-correcting capability, all layer i data packets are recoverable if any M packets are received out of the M data and k_i FEC packets. Assume packets are lost with probability p, and that losses are independent and identically distributed across all data and FEC layers. Then, the layer i packet of component c is lost with probability p. Furthermore, the lost packet cannot be recovered only when less than a total of M packets are received from the remaining $M + k_i - 1$ data and FEC packets. Thus, we have:

$$q_i^{(c)} = p \cdot \left[\sum_{w=0}^{M-1} \binom{M + k_i - 1}{w} (1-p)^w \cdot p^{M+k_i-1-w} \right] \tag{4}$$

Equations 2 through 4 allow the computation of distortion at specific packet loss rate and subscription level of data and FEC layers. Together with Equation 1, we can then numerically determine an optimal set of data and FEC layers, through exhaustive search, for each prescribed bandwidth constraint and observed packet loss rate. This computation is used in two later Sections; in Section 4, it is used to determine how optimal FEC changes with bandwidth constraint and packet loss rate. In Section 6, the computation is used, in both the simulations and the experiments, to provide each receiver with an optimal subscription of data and FEC layers. A more elaborate analysis assuming different number of packets in different data layers can be found in [35].

4 Construction of Layered FEC

Before we look into the problem of constructing the layers menu, we need to first choose an error control code for layered FEC. An effective FEC scheme will allow recovery of lost data packets with as little redundancy packets as possible. This is achieved when the loss of any m data packets are recoverable from any m parity packets. Such properties are often referred to as maximal distance

separable (MDS) in algebraic coding theory [22]. An example of MDS code is the Reed-Solomon code, which has long been used in data communications to provide incremental redundancy [36]. Fig. 5 illustrates the use of a systematic (n, k) MDS code that takes k data packets and produces $n-k$ FEC packets, which are partitioned to form FEC layers. Let m_i denote the number of packets in the $i-$th FEC layer. Since punctured MDS codes are also MDS [22], the first FEC layer can correct up to m_1 packet losses, the maximum any error control code with m_1 redundancy packets can achieve. Similarly, the first l FEC layers achieve maximum error correcting capability by being able to correct up to $\sum_{i=1}^{l} m_i$ lost packets. Hence, there is no loss in error correcting capability by using MDS codes for layered FEC.

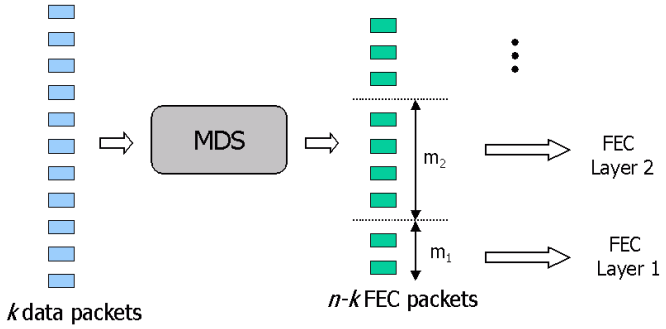
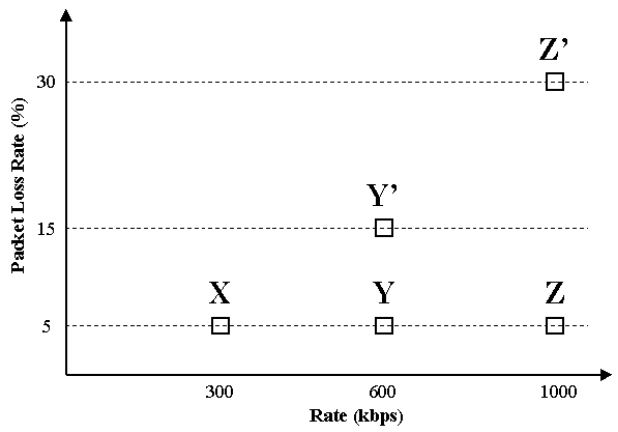


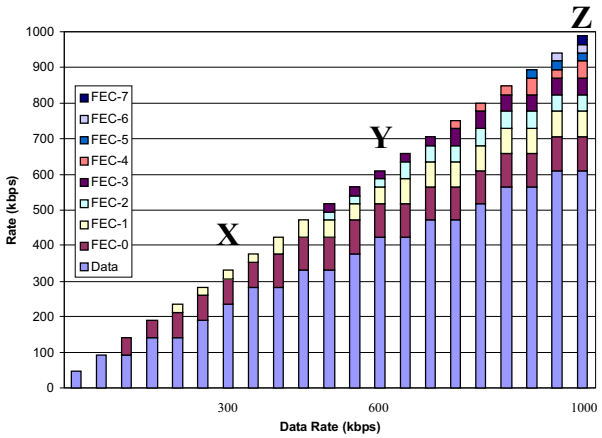
Figure 5: *Generating FEC layers using MDS codes.*

Scalable video compression algorithms typically produce embedded bit-streams. This means to decode video data layer k, all lower layers 0 to $k-1$ must also be available to decode. In such cases, we call the layers *totally ordered*. In the special case when FEC protection is provided for the first data layer only, an embedded FEC-stream spread over multiple layers can be used in a prescribed order, with layer FEC-0-1 before FEC-0-2, and so on and so forth [37]. One way to extend this approach to protect multiple video layers is to order all FEC layers corresponding to all video layers in a prescribed way, and have each receiver decide how many layers of this ordered list it would subscribe to. However, as we shall see later in this section, such total ordering could unnecessarily restrict the choices for the various receivers, which may desire different sets of FEC

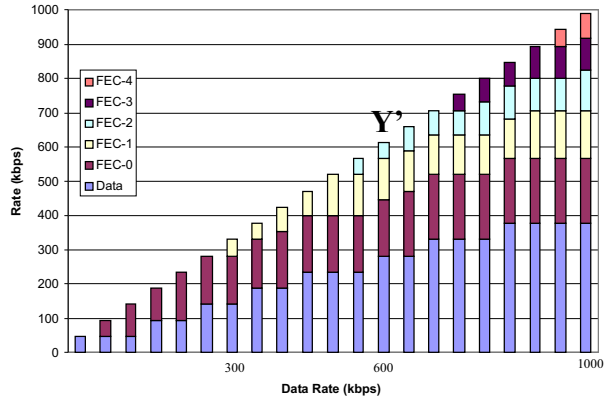
layers, depending on their available bandwidth, experienced loss rate and the data dependence of the scalable video compression used.



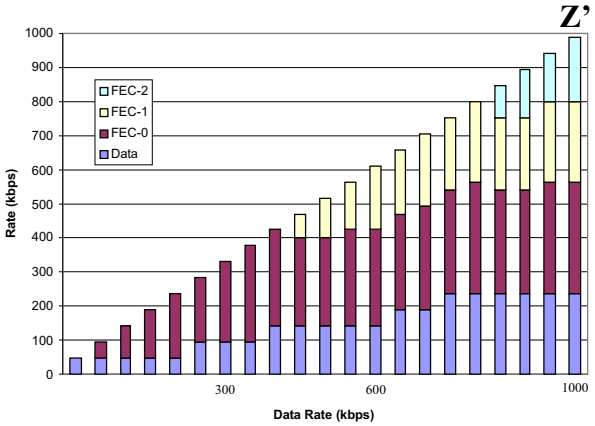
(a): *Example operating points.*



(b): *Optimal subscription for 5% loss.*



(c): *Optimal subscription for 15% loss.*



(d): *Optimal subscription for 30% loss.*

Figure 6: *Optimal FEC protection for layered video source at different available bandwidth and packet loss rates. The video layers are of constant rate of 47.3 kbps.*

In Section 3, we discussed how to optimally choose a set of data and FEC layers given a bandwidth constraint and packet loss rate. Fig. 6 shows the optimal sets for receivers operating at different operating points, as characterized by different values of p and B in the p-B plane. Fig. 6-(a) shows the loss-bandwidth operating points of various receivers and Figs. 6-(b), (c), and (d) show the resulting optimal subscription levels for each operating point. The aggregate bit-rate of all the video layers combined is labelled "Data" in Fig. 6, while the bit-rate for FEC protection of video

layer i is labelled "FEC-i". We see that for receivers operating at point X, with $B = 300$ *kbps* and $p = 5\%$, the optimal protection is to use roughly 70 and 23.5 *kbps* to protect the first and second data layers respectively, and leave the other data layers unprotected. Moving along the 5% loss line in Fig. 6-(a) to points Y and Z, with bandwidth of 600 *kbps* and 1000 *kbps* respectively, we see from Fig. 6-(b) that the optimal number of layers to protect the first data layer does not increase significantly when the total bandwidth is increased from 300 *kbps* to 600 and 1000 *kbps*. This is because the first data layer is already sufficiently protected, and additional bandwidth is used to protect other data layers instead. Thus, to satisfy the FEC needs of receivers X, Y and Z using FEC layers that are added in a fixed order, it is desirable that the first few FEC layers provide enough FEC for receiver X, with additional FEC layers extending FEC protection to more data layers. Moving along the path $X \rightarrow Y' \rightarrow Z'$ however, with loss rates of 5, 15, and 30% respectively and bandwidth of 300, 600 and 1000 *kbps* respectively, we see from Figs. 6-(b) to (d) that the desired protection level for both data layers 1 and 2 increases significantly from X to Y', and from Y' to Z'. Thus, to efficiently satisfy the FEC needs of receivers Y' and Z', after the FEC need of X is met, it is desirable that additional FEC layers provide more protection to the first few data layers rather than extending protection to other data layers. Thus, it is clear that the FEC requirements of receivers Y and Z are different from those of Y' and Z'. This can also be seen by comparing the composition of FEC layers in Figs. 6-(b) to (d). This implies that there exists no one total ordering scheme for the FEC layers that results in optimal subscription levels for different receivers with different loss/bandwidth characteristics.

A more flexible alternative is to employ a partial ordering as follows: Each FEC layer can be used to protect only one data layer. While no explicit join order exists for different FEC layers that protect different data layers, receivers are required to join in sequence the FEC layers that protect the same data layer. In such a way, FEC layers are partitioned into sets, each of which protects a different data layer. A receiver then need only decide for each set how many FEC layers

to subscribe to, depending on its total available bandwidth and experienced loss rate. In such a way, each receiver can individually decide how much FEC to protect each data layer without the unnecessary constraints of a single ordering. While this approach is more flexible, it does require more computation compared to a total-ordering approach. This is because finding the optimal combination of data and FEC layers is a linear search when total-ordering is used, but a combinatorial search when partial-ordering is used. We show experimental results corresponding to partial-ordering in Section 6. Thus, our goal in system design is to provide sufficient number of FEC layers for each data layer in order to enable heterogeneous population of receivers to optimally choose the appropriate subscription level from the available "menu".

Using more and thinner FEC layers generally yields finer grain protection. This however, consumes a large number of multicast addresses and induces considerable overhead costs. In practice, the number of multicast addresses is limited. To better utilize the multicast address space, one can strategically use the FEC layers to cover a typical region in the p-B plane. In that case, a receiver outside the typical region may not find the best possible FEC from the menu, but can nevertheless still optimize within the menu. The tradeoff between the number of multicast addresses used and the granularity of available protection requires a cost/benefit analysis, and is beyond the scope of this paper. Alternative schemes based on collecting receiver interests to adaptively change the source [6, 38] require additional protocols to provide closed-loop feedback to the sender, thereby limiting scalability to large group of receivers. However, given channel conditions of all receivers, an optimal menu that minimizes the average distortion of all receivers can be found using iterative techniques [39].

5 Equation-based Rate Control

A central part of a layered multicast system is the rate control algorithm at the receivers. This is traditionally realized by dropping a layer when congestion is detected and probing for more band-

width when losses are absent [11, 16]. One early example is the receiver-driven layered multicast (RLM) of [11]. Another example is the RLC scheme of [16] which achieves TCP-friendliness by using exponentially distributed layer sizes, and by carefully choosing the time instants at which layers can be added. Since the act of probing for bandwidth may cause congestion, leading other receivers to inadvertently reduce their subscription levels, the difficult tasks of coordinating probes [16] and communicating their results are necessary [11]. While such coordination techniques may be effective for a single multicast session, there are no existing mechanisms to coordinate probes across different sessions, leading to potentially unfair allocation of bandwidth. A more recently proposed scheme, equation-based rate control (ERC), does not involve probing [17, 4]. Instead, the available bandwidth is computed directly at the receivers based on measured quantities such as packet loss rate.

One characteristic of ERC is its ability to allocate bandwidth fairly among competing sessions [40]. For example, two simultaneous video multicast sessions which share one source-destination pair but with different starting times, may initially have different rates. However, since the two sessions will measure the same channel conditions, and therefore compute the same rates under ERC, they will eventually converge towards the same target rate for the shared source-destination pair. This is achieved by adding layers when the current layer is below the target rate and dropping them otherwise. In contrast, layer adding/dropping techniques based on simple thresholding, as employed by RLM, do not have a common target rate for competing sessions. As a result, two sessions sharing the same bottle-neck but with different initial rates will likely add and drop layers synchronously rather than converging towards fair sharing [41]. In effect, ERC achieves fairness by exploiting packet loss rate as an implicit side-information to compute a fair target rate for competing sessions.

Existing ERC implementations are typically based on the TCP-friendly equation [42] which gives

the throughput T of a TCP connection under packet loss rate p and round trip time rtt,

$$T = \frac{1.22 \cdot MTU}{rtt \cdot \sqrt{p}} \tag{5}$$

where MTU, the maximum transport unit, is a constant often taken to be 500 bytes. While Equation 5 and its variants may be used directly to provide TCP-friendliness [17, 43], its use in a multicast context has several drawbacks. First, it requires the inherently non-scalable rtt measurement [17], which has partly been addressed by [43] recently. Second, as shown in Fig. 7, a dependence on rtt would cause two receivers, R_1 and R_2, connected to the same source but with different rtt to have different rates even if they share the same bottleneck, resulting in under-utilization of resources.

Generally speaking, our proposed layered FEC scheme can be used with different implementations of equation-based rate control algorithm, be it dependent on round-trip time and thus TCP-friendly, or independent of round-trip time to avoid network under-utilization. In this paper however, we choose to employ an equation-based scheme that that does not depend on rtt but replaces it by a constant in Equation. 5. The choice of the constant dictates the aggressiveness of ERC in consuming bandwidth under different packet loss rates. While actual rtt would vary between hundreds of milliseconds for inter-continental traffic to under 10 milliseconds for local traffic, we choose for our experiments in Section 6 an intermediate value of 80 milliseconds, which yields reasonable communication rate as a function of packet loss rate for the experiments.

Another desirable feature in multicast rate control algorithms for multimedia is to adapt slowly in time to changing network conditions. There are two reasons for this. First, while scalable video is well suited to time varying channels, rapid variations in communication rates results in visually unpleasant video. Second, even after a multicast group is unsubscribed, packets in that group will still be delivered until pruning of the multicast tree is completed. Such leave-latency can last several seconds [41], making it impractical to adapt on a time-scale shorter than a few seconds. For instance, when losses are bursty, direct application of Equation 5 using instantaneous estimates of

p may cause the estimated rate to fluctuate rapidly. Such short term fluctuations can be suitably reduced by smoothing algorithms. In our ERC implementation, an averaging time-window is used at the receivers to estimate the instantaneous packet loss rate $\hat{p}$, which is averaged over all data and FEC layers. Generally, a smaller window is able to track changes in network conditions faster than a larger window, but at the expense of larger overhead. Since a multicast rate control algorithm should not react faster than a few seconds, the higher complexity of a sub-second window is not justified. To measure $\hat{p}$ in our experiments in this paper, we therefore choose a three seconds window to provide sufficient responsiveness to changing network conditions. The p used in rate control is then obtained by smoothing $\hat{p}$ using first order auto-regression: $p \rightarrow \lambda p + (1 - \lambda)\hat{p}$. The parameter λ controls the tradeoff between responsiveness and smoothness of output rates, and a value of 0.9 is found experimentally to yield a reasonable tradeoff for video communications.

To evaluate the performance of ERC compared to existing implementations of RLM and RLC when multiple multicast sessions are present, we perform simulations using the network simulator (ns) and the dumbbell topology in Fig. 8. In the contemporary and independent work of [43], the TCP-friendliness of RLC is examined when only a single multicast session is present. Our simulation differs in that we are interested in the interaction and fairness between independent and competing multicast sessions. Constant layer size of 47 *kbps* is used for ERC and RLM to result in

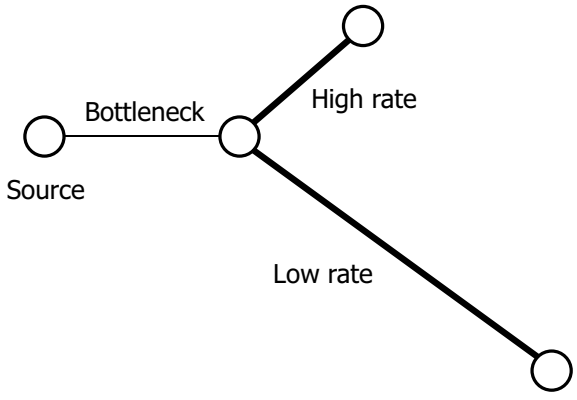


Figure 7: *Under-utilization caused by dependence on rtt.*

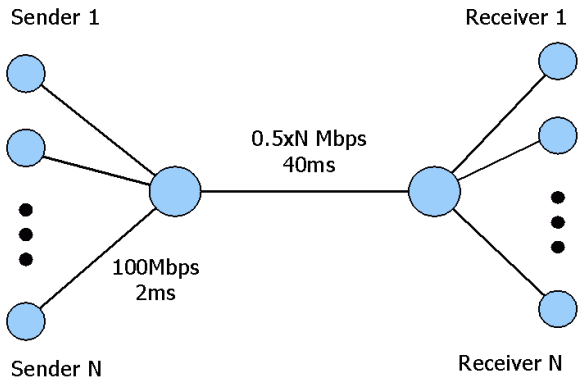


Figure 8: *"Dumbbell" topology.*

fine rate granularity without using an excessive number of multicast addresses. Since RLC requires exponential layer sizes, a base layer size of 47 *kbps* is used, with each additional layer doubling the total bandwidth. The simulation consists of 4 sessions between 4 different pairs of senders and receivers in Fig. 8. The throughputs of the 4 sessions as a function of time as they coexist in the dumbbell topology are shown in Fig. 9. Of the 4 sessions, two are ERC sessions with one receiver, and two are TCP sessions. The 4 protocol instances share a bottleneck bandwidth of 2 *Mbps* with the two ERC instances starting at time 0 and 300 seconds respectively and the TCP instances at 100 and 200 seconds respectively. We see that the ERC instances finally settle to similar eventual rates despite the fact that at $t = 300$s they have different rates. While Fig. 9 shows fair sharing of bandwidth by the 4 protocol instances, fair sharing of bandwidth between TCP and ERC instances does not necessarily happen in general, since unlike TCP, our ERC implementation does not depend on round trip time. As mentioned earlier, this lack of dependence on *rtt* is done deliberately in order to achieve scalability and avoid under-utilization of network resources. Regardless of fair sharing with TCP, we have shown experimentally that different ERC instances share the available bandwidth fairly amongst themselves.

Figs. 10 and 11 shows the corresponding results for RLM and RLC respectively. We see in Fig. 10 that the 2 RLM instances fail to share bandwidth fairly. In particular, the first RLM instance, starting at time 0, hogs most of the bandwidth, and since packet loss rate is lower than the drop threshold, it refuses to relinquish bandwidth to later TCP and RLM instances. In Fig. 11, the relatively large layer size of RLC near 500 *kbps* induces vigorous reaction and oscillation when layers are added. In particular, the first RLC instance starting at time 0 consistently adds large layers, causing congestion, to which the second instance, starting at $t = 300$s would react by dropping layers. This can be attributed to the fact that even though one RLC session can prevent oscillation amongst its receivers through synchronization, when there are multiple sessions, the lack of inter-session synchronization may give rise to oscillations.

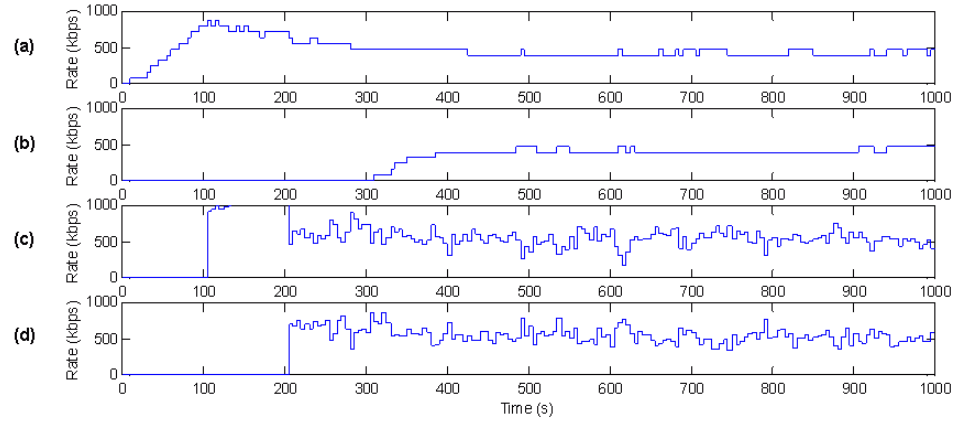


Figure 9: *Sharing of bandwidth by (a) ERC flow starting at time 0, (b) ERC flow starting at time 300s, (c) TCP flow starting at time 100s, and (d) TCP flow starting at time 200s.*

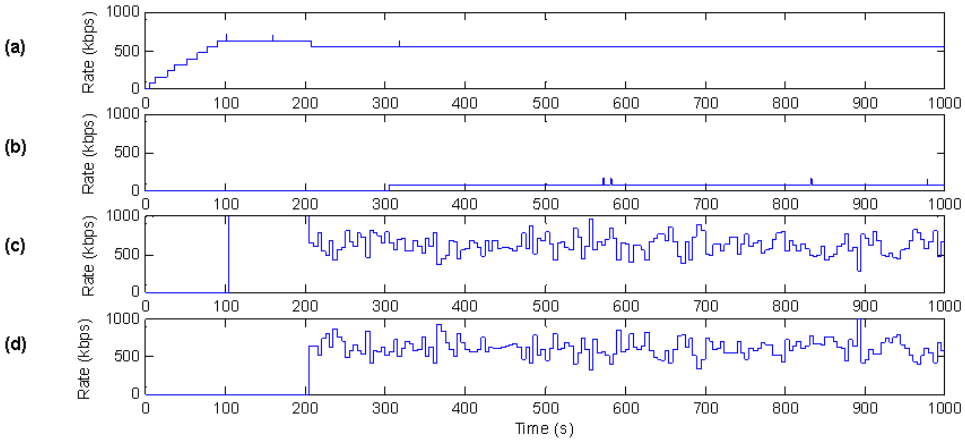


Figure 10: *Sharing of bandwidth by (a) RLM flow starting at time 0, (b) RLM flow starting at time 300s, (c) TCP flow starting at time 100s, and (d) TCP flow starting at time 200s.*

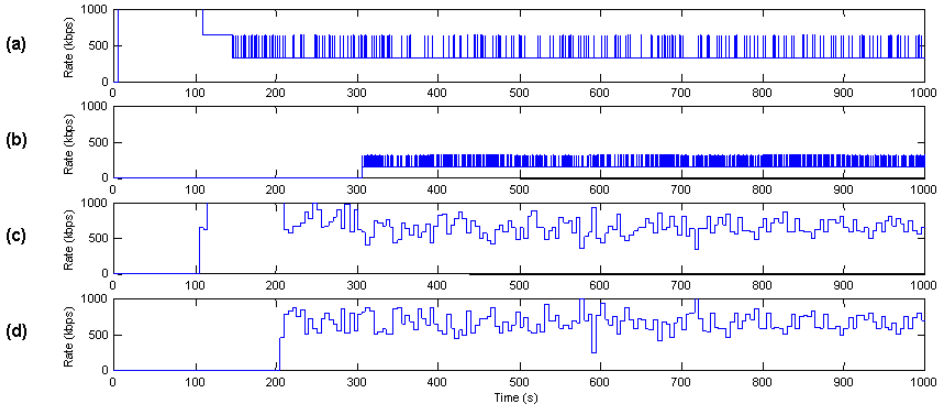


Figure 11: *Sharing of bandwidth by (a) RLC flow starting at time 0, (b) RLC flow starting at time 300s, (c) TCP flow starting at time 100s, and (d) TCP flow starting at time 200s.*

To further quantify the fairness properties of the various protocols, a fairness index can be used [16]. A fairness index $\mathcal{F}$ of a set of N rates X_i can be defined to be a function between 0 and 1, that is 1 when all the rates are identical, and close to 0 when a large fraction of the sum of the rates $\sum_i X_i$ is concentrated in one rate, say X_j. One such function is:

$$\mathcal{F} = 1 - \frac{1}{2(N-1)} \sum_{i=1}^{N} \left| \frac{X_i}{\mu} - 1 \right| \tag{6}$$

where μ is the average rate. Thus, $\mathcal{F}(1,3,5,7,9) = 0.7$ and $\mathcal{F}(50,51,52) = 0.990$, matching the intuition behind fairness index.

Fig. 12 summarizes the fairness indices for ERC, RLM, and RLC on the dumbbell topology of Fig. 8 using two layer size organization methods. Under the *linear* organization, layers are of equal size so that the aggregate bandwidth is a linear function of the number of layers. On the other hand, under the *exponential* organization, the aggregate bandwidth doubles with each additional layer. The bottleneck bandwidth is adjusted proportionally to the number of flow to result in 500 *kbps* per flow. The results are obtained by running simulations of 1000 seconds duration and measuring the average rate of each flow using the last 300 seconds to exclude transient effects. Protocol instances are started at random times between 0 and 500 seconds, and reported results are averaged over 3 independent runs. We see that ERC using both linear and exponential layer

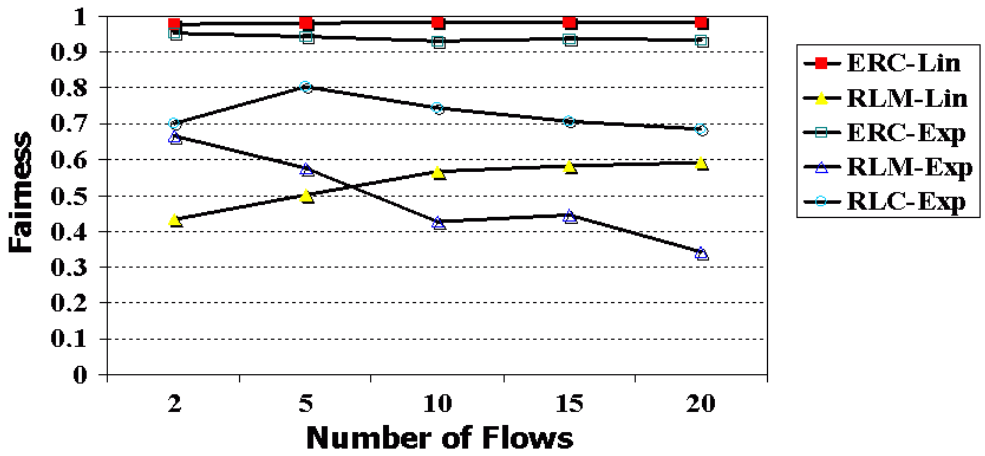


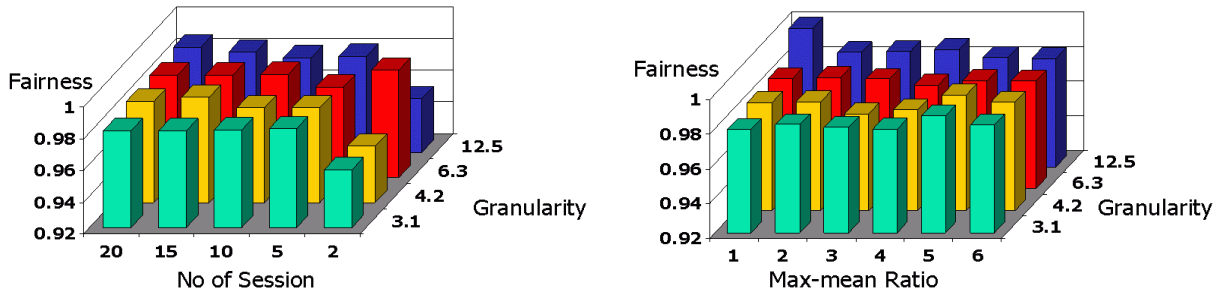
Figure 12: *Fairness comparisons of ERC, RLM and RLC using linear and exponential layer sizes.*

sizes score significantly higher fairness index than both RLM and RLC.

Finally, to further examine the fairness property of ERC, we compute the fairness index as a function of the number of flows, granularity, and max-mean ratio. Granularity is defined as the number of constant size layers that can be supported by a flow's fair share of bandwidth, and is obtained by varying the layer sizes while fixing the bandwidth so that each flow has 500 *kbps* on the average. Max-mean ratio is the ratio of the maximum video source rate available at the sender to the average fair rate, and signifies the worst-case bandwidth hogging a session can achieve. For example, when max-mean ratio is close to 1, the amount of bandwidth a session can hog is close to its fair share, thereby guaranteeing fair allocation. On the other hand, when max-mean ratio is much larger than one, a session can potentially hog much more bandwidth than its fair share, leading to potentially unfair allocations. It is varied by changing the maximum video source rate available at the sender. Five independent runs are performed and the results are averaged and shown in Fig. 13. We observe that high level of fairness, above 0.95 in all cases and close to 0.98 in most cases, is achieved across the parameter space. From Fig. 13-(a), we see that fairness index is generally lower when the number of sessions is two. This is because for very small number of sessions, the resulting change in loss rate due to a receiver adding or dropping a layer is large, causing large variations in measured average packet loss rate, and thus transmission rate. In Fig. 13-(b), we see that fairness is largely independent of the max-mean ratio and granularity when 10 sessions are used. This is because with the multiplexing of at least a moderate number of sessions, relatively consistent measurements of loss can be maintained, giving rise to fair sharing of bandwidth.

6 Video Transmission Results: Simulations and Experiments

In this section, we will describe simulations and actual MBONE experiments using the layered FEC with scalable video scheme. Scalable compression technique in [4] and equation based rate control



(a): *Using flows of maximum rate 1.5 Mbps.* (b): *Using 10 sessions.*

Figure 13: *Fairness of ERC under different conditions.*

of Section 5 are used. The purpose of the simulations is to evaluate the scheme under varying and controlled network loads. Actual MBONE experiments, on the other hand, provide realistic testing conditions. A 25 seconds clip from the movie "Raider of the Ark" of size 320×224 at 12 fps is used as source, which is compressed using a GOP of 4 for low latency, and is repeatedly transmitted.

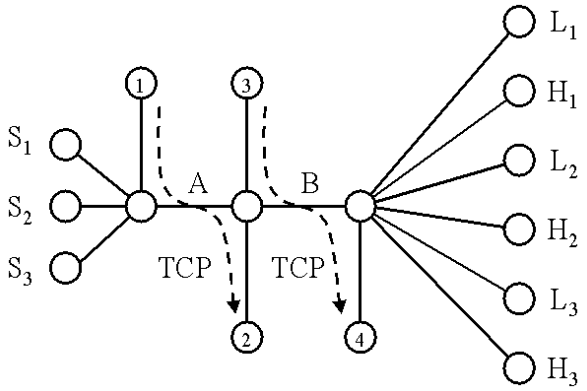


Figure 14: *Modified "dumpbell" topology.*

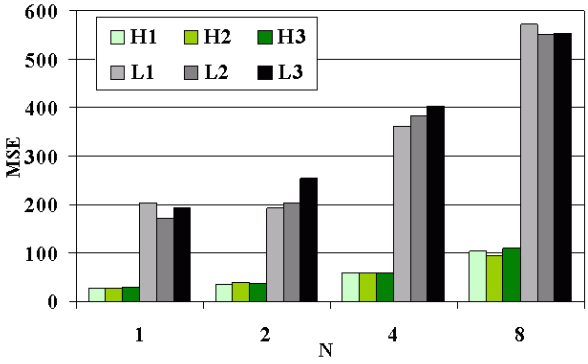


Figure 15: *Average MSE experienced by low-latency (L_i) and high-latency (H_i) viewers under different number of TCP cross traffic (N).*

Fig. 14 shows the modified dumbbell topology used in the *ns* simulation. Three independent sources S_1, S_2 and S_3 are started at random times between 0 and 50 seconds. All three sources use the same menu of data and FEC layers, which contains 10 layers of data at 100 *kbps* each, and 10 FEC layers of 50 *kbps* each. Five of the FEC layers are used to protect the first data layer, while 3, 1 and 1 FEC layers are used to protect the second, third and fourth data layers respectively. This is roughly optimized for packet loss rates in the range of 5 to 15%. Each source S_i has two

receivers, L_i, a low-latency viewer who will not use any FEC in order to achieve low latency, and H_i, a high-latency viewer who is willing to trade delay for better video quality by taking advantage of the proposed layered FEC method.

To serve as cross traffic, N instances of TCP from node 1 to node 2, and another N instances from node 3 to node 4 are each started at a random time between 0 and 50 seconds. The links A and B, shared by the layered FEC sources and TCP, each has a bandwidth of 3 $Mbps$. This is barely enough to carry the 10 data layers of 100 $kbps$ for the 3 sources. By setting N to 1, 2, 4, and 8, average loss rates of 4, 5, 8, and 12% are sustained respectively in the shared link, yielding available bandwidths of 620, 550, 420 and 340 $kbps$.

Fig. 15 shows the average distortion, measured in mean squared error (MSE), of the low and high latency viewers, for different number, N, of TCP cross-traffic. As we expect, the average distortion experienced by the high-latency viewers is significantly lower than that of the low-latency viewers under the same conditions. By using layered FEC, the MSE is reduced by approximately a factor of 6 at the cost of an additional delay of at most 10 GOPs or 3.3 seconds. Due to the adaptability of layered FEC to different channel conditions, we see that the average distortion increases only slightly as the number of cross-traffic is increased when layered FEC is used. Fig. 16 shows the distortion traces as a function of time observed by the low-latency and high-latency viewers L_1 and H_1. We see that the use of layered FEC not only yields a lower distortion, but also smaller variation in distortion over time. The large variation in distortion for the low-latency viewers is caused by traffic burstiness. Even though the average loss rate may be as moderate as 4%, the instantaneous loss rates during a GOP time or 1/3 second can sometimes reach 30%. The problem is aggravated by the small number of packets per layer per GOP. Supporting reasonable packet sizes and layer sizes, such as 400 bytes and 100 $kbps$ respectively results in a small number of packets per layer per GOP, such as 10. This means that a single packet loss in a data layer corresponds to 10% loss of data in the GOP, regardless of how low the average packet loss rate is. By sufficiently

protecting only the more important data layers, lower average distortion and smaller variation in distortion can be achieved by layered FEC. Fig. 16 shows that variation in distortion remains low as N increases.

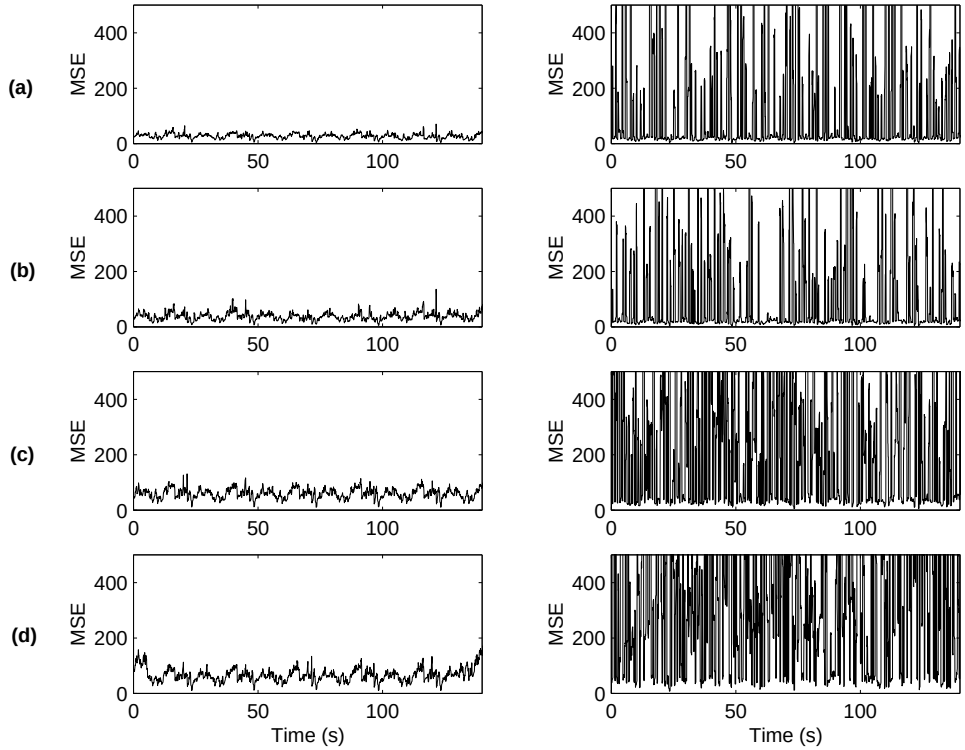


Figure 16: *Distortion traces for receivers using layered "staggered" FEC (left) and not using layered FEC (right). The numbers of TCP cross traffic used are (a) 1, (b) 2, (c) 4, and (d) 8 respectively.*

To illustrate the importance of delaying transmission of FEC layers, the above simulation is repeated with the modification that all FEC layers for a GOP are transmitted in the same time slot as the data layers. The distortion traces for viewer H_1 is shown in Fig. 17 for different number, N, of TCP cross-traffic. We see that while results similar to Fig. 16 are obtained when N is 1 or 2, occasional "peaks" of high MSE appear for $N = 3$ and 4 when transmission of FEC layers are not delayed. Such peaks are a result of bursty losses, which induce heavy packet drops to both data and FEC layers of a GOP if they are transmitted concurrently. In this case, since data and FEC layers are of size 100 and 50 *kbps* respectively, even if only one FEC layer is used to protect the

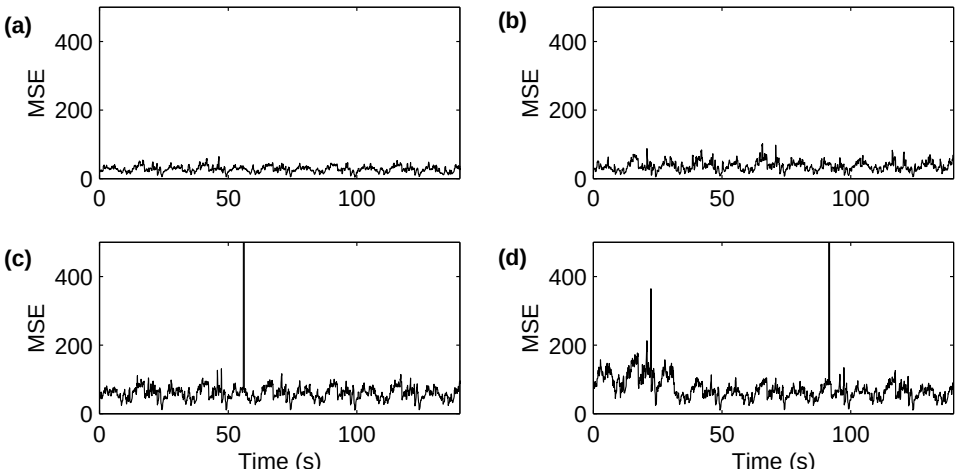


Figure 17: *Distortion traces when all FEC layers are transmitted in the same time slot as data layers. The numbers of TCP cross traffic used are (a) 1, (b) 2, (c) 4, and (d) 8 respectively.*

first data layer, bursty losses of over 33% are needed to cause distortion peaks. Such large bursts are less likely when N is small, resulting in low distortion traces in Fig. 17 for $N = 1$ and 2.

We next describe our MBONE experiments using 3 hosts located at Berkeley, Indiana and Sweden. A menu consisting of 6 data layers of 100 *kbps* each and 6 FEC layers of 50 *kbps* each is used. Of the 6 FEC layers, 4 and 2 are used to protect the first and second data layers respectively. This is because the first data layer always needs more protection than the second data layer and so forth. Furthermore, from Fig. 6, FEC is needed for data layer 3 only at relatively high available rates of 600 *kbps* and above, which is unlikely in our experiment. The video source is located at Indiana, and at each of the locations Berkeley and Sweden, one low-latency and one high-latency viewers is simultaneously present.

While low packet loss is observed for video flowing from Indiana to Berkeley, relatively bursty losses appear in the video received at Sweden. Figs. 18 (a) and (b) show the distortion or MSE trace in the absence and presence of layered FEC for transmission from Indiana to Sweden. The packet loss rate seen by the first data layer after and before correction by FEC are shown in Figs. 18 (c) and (d) respectively. We see that the use of layered FEC helps to significantly reduce the effects of bursty losses, which are manifested in "peaks" of high MSE. However, the use of FEC reduces the

bandwidth available for data and when there is no packet loss, the resulting distortion is slightly higher with layered FEC. While the burst near $t = 43$s causes more packet losses than the burst near time 80 seconds, the resulting distortion is higher in the latter case when layered FEC is used. This is because the prolonged period of loss-free transmission between time 50-80 seconds causes the high-latency receiver to abandon FEC protection, thereby increasing susceptibility to sudden losses. The average MSE of the traces are 31.8 and 82.7 with and without layered FEC respectively.

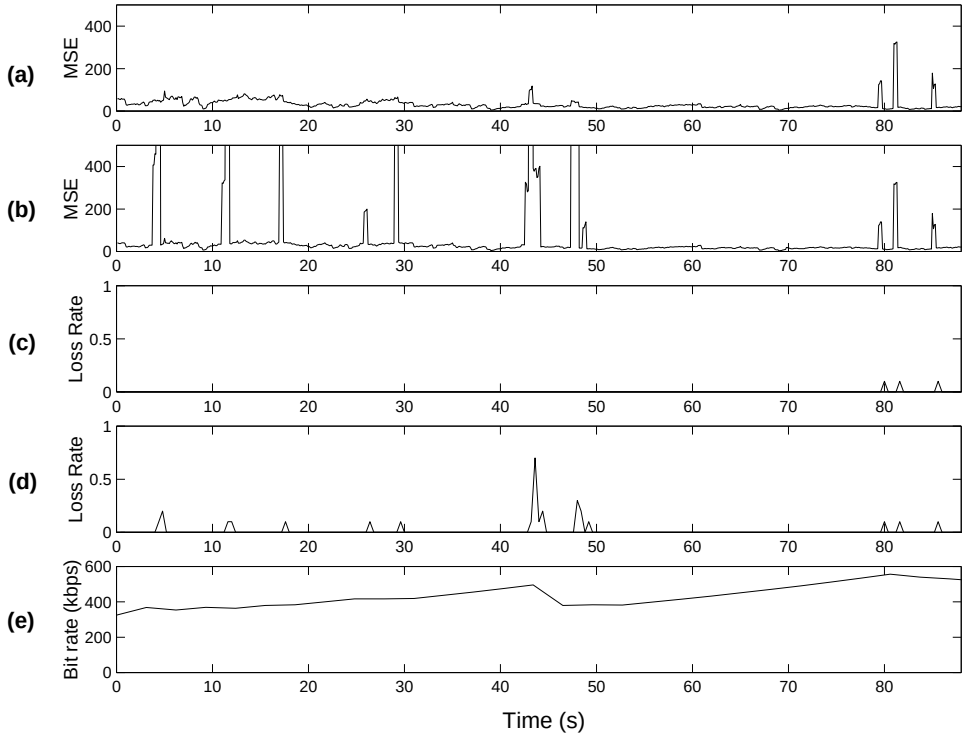


Figure 18: *MBONE transmission from Indiana to Sweden (10 am, 12/14/99). (a) MSE when layered FEC is used, (b) MSE when no FEC is used, (c) effective packet loss rate in first data layer after correction by FEC, (d) packet loss rate for first data layer, and (e) available bandwidth calculated by ERC rate controller.*

The same MBONE experiment is repeated at a later time to verify the effectiveness of our technique. Again, no significant loss is observed between Indiana and Berkeley, while heavy and bursty losses are observed between Indiana and Sweden. Figs. 19 (a) and (b) show the MSE trace in the presence and absence of layered FEC for the Indiana-Sweden communication. Results are similar to Fig. 18 in that the use of layered FEC not only results in a lower average MSE of 120

compared to 450 when FEC is not used, but also causes smaller variation in distortion. Fig. 19 (c) shows the packet loss rate seen by the first data layer. Despite bursty losses that often exceed 50%, all first layer data packets are recovered through the use of the layered FEC technique.

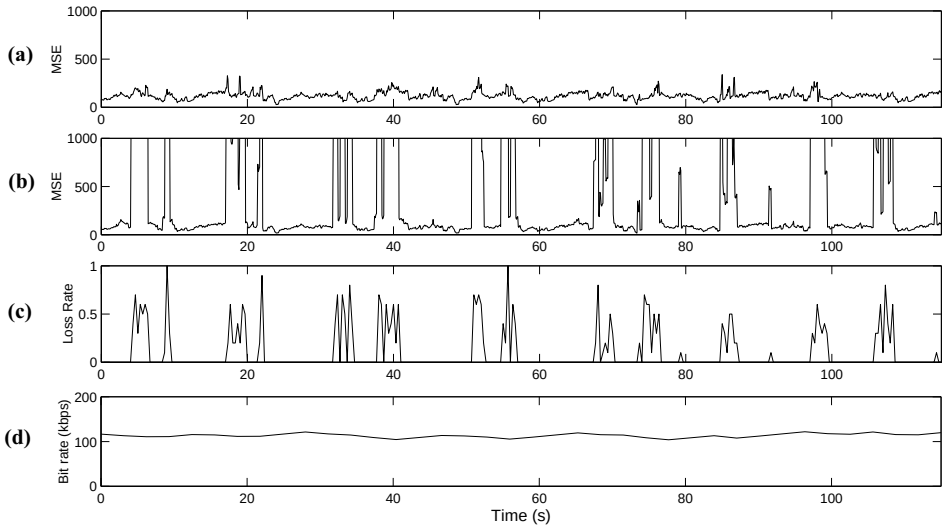


Figure 19: *MBONE transmission from Indiana to Sweden (2 pm, 15/5/00). (a) MSE when layered FEC is used, (b) MSE when no FEC is used, (c) packet loss rate for first data layer, and (d) available bandwidth calculated by ERC rate controller.*

7 Summary and Conclusions

We have proposed the use of layered FEC as an error control method that allows each receiver to individually trade-off latency for better reception quality. The scheme can be combined with layered multicast to simultaneously provide flow and error control, and is efficient in that FEC is provided only for the more important layers of the video data, and that there is no loss in error correcting capability by imposing a layered FEC structure. Simulation and MBONE experiments are performed to demonstrate the utility of scheme.

References

[1] V. Jacobson, "Congestion avoidance and control," in *Proc. ACM Sigcomm '88*, Stanford, CA., Aug. 1988, pp. 314–29.

[2] H. Kanakia, P. Misha, and A. Reibman, "An adaptive congestion control scheme for real time packet video transport," *ACM/IEEE Trans. Networking*, vol. 3, no. 6, pp. 671–82, Dec. 1995.

[3] T. Turletti and C. Huitema, "Video-conferencing on the Internet," *ACM/IEEE Trans. Networking*, vol. 4, no. 3, pp. 340–51, June 1996.

[4] W. Tan and A. Zakhor, "Real-time Internet video using error resilient scalable compression and TCP-friendly transport protocol," *IEEE Trans. Multimedia*, vol. 1, no. 2, pp. 172–86, June 1999.

[5] J. Bolot, T. Turletti, and I. Wakeman, "Scalable feedback control for multicast video distribution in the Internet," in *Proc. ACM Sigcomm '94*, London, England, Sept. 1994, pp. 58–67.

[6] E. Amir, S. McCanne, and R. Katz, "Receiver-driven bandwidth adaptation for light-weight sessions," in *Proc. ACM Intl. Multimedia Conf. '97*, Seattle, WA., Nov. 1997, pp. 415–26.

[7] S. Cheung, M. Ammar, and X. Li, "On the use of Destination Set Grouping to improve fairness in multicast video distribution," in *Proc. IEEE Infocom '96*, San Francisco, CA., Mar. 1996, pp. 553–60.

[8] X. Li, M. Ammar, and S. Paul, "Video multicast over the Internet," *IEEE Network*, vol. 13, no. 2, pp. 46–60, Apr. 1999.

[9] N. Shacham, "Multipoint communication by hierarchically encoded data," in *Proc. IEEE Infocom '92*, Florence, Italy, May 1992, vol. 3, pp. 2107–14.

[10] S. Deering, "Internet multicast routing: State of the art and open research issues," in *Multimedia Integrated Conferencing for Europe*, Oct. 1993.

[11] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," in *Proc. ACM Sigcomm '96*, Palo Alto, CA., Aug. 1996, pp. 117–30.

[12] D. Taubman and A. Zakhor, "Multirate 3-d subband coding of video," *IEEE Trans. Image Processing*, vol. 3, no. 5, pp. 572–88, Sept. 1994.

[13] S. McCanne, M. Vetterli, and V. Jacobson, "Low-complexity video coding for receiver-driven layered multicast," *IEEE J. Selected Area Comm.*, vol. 15, no. 6, pp. 983–1001, Aug. 1997.

[14] B. Kim and W. Pearlman, "Low-delay embedded 3-d wavelet color video coding with SPIHT," in *Proc. SPIE Visual Comm. and Image Processing '98*, San Jose, CA., Jan. 1998, vol. 3309, pp. 955–64.

[15] D. Taubman and A. Zakhor, "A common framework for rate and distortion based scaling of highly scalable compressed video," *IEEE Trans. Circuits Sys. Video Tech.*, vol. 6, no. 4, pp. 329–54, Aug. 1996.

[16] L. Vicisano, L. Rizzo, and J. Crowcroft, "TCP-like congestion control for layered multicast data transfer," in *Proc. IEEE Infocom '98*, San Francisco, CA., Mar. 1998, vol. 3, pp. 996–1003.

[17] T. Turletti, S. Parisis, and J. Bolot, "Experiments with a layered transmission scheme over the Internet," Tech. Rep., INRIA, http://www.inria.fr/RRRT/RR-3296.html, Nov. 1997.

[18] X. Li, S. Paul, and M. Ammar, "Layered video multicast with retransmissions (LVMR): Evaluation of hierarchical rate control," in *Proc. IEEE Infocom '98*, San Francisco, CA., Mar. 1998, vol. 3, pp. 1062–72.

[19] G. Cote, B. Erol, M. Gallant, and F. Kossentini, "H.263+ video coding at low bit rates," *IEEE Trans. Circuits Sys. Video Tech.*, vol. 8, no. 7, pp. 849–66, Nov. 1998.

[20] S. Floyd, V. Jacobson, C. Liu, S. McCanne, and L. Zhang, "A reliable multicast framework for light-weight sessions and application level framing," *ACM/IEEE Trans. Networking*, vol. 5, no. 6, pp. 784–803, Dec. 1997.

[21] S. Paul, K. Sabnani, J. Lin, and S. Bhattacharyya, "Reliable multicast transport protocol (RMTP)," *IEEE J. Selected Area Comm.*, vol. 15, no. 3, pp. 407–21, Apr. 1997.

[22] S. Wicker, *Error Control Systems for Digital Communication and Storage*, Prentice-Hall, 1995.

[23] J. Nonnenmacher, E. Biersack, and D. Towsley, "Parity-based loss recovery for reliable multicast transmission," *ACM/IEEE Trans. Networking*, vol. 6, no. 4, pp. 349–61, Aug. 1998.

[24] S. Pejhan, M. Schwartz, and D. Anastassiou, "Error control using retransmission schemes in multicast transport protocols for real-time media," *ACM/IEEE Trans. Networking*, vol. 4, no. 3, pp. 413–27, June 1996.

[25] R. Kermode, "Scoped hybrid automatic repeat request with forward error correction (SHARQFEC)," in *Proc. ACM Sigcomm '98*, Vancouver, Canada, Sept. 1998, pp. 278–89.

[26] J. Rosenberg and H. Schulzrinne, "An RTP payload format for generic forward error correction," *RFC2733*, Dec. 1999.

[27] X. Xu, A. Myers, H. Zhang, and R. Yavatkar, "Resilient multicast support for continuous-media applications," in *Proc. IEEE NOSSDAV '97*, New York, NY., May 1997, pp. 183–94.

[28] J. Nonnenmacher, L. Martin, J. Matthias, E. Biersack, and G. Carle, "How bad is reliable multicast without local recovery," in *Proc. IEEE Infocom '98*, San Francisco, CA., Mar. 1998, vol. 3, pp. 972–9.

[29] X. Li, S. Paul, P. Pancha, and M. Ammar, "Layered video multicast with retransmission (LVRM): Evaluation of error control scheme," in *Proc. IEEE NOSSDAV '97*, New York, NY., May 1997, pp. 161–72.

[30] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," in *Proc. ACM Sigcomm '98*, Vancouver, Canada, Sept. 1998, pp. 56–67.

[31] J. Bolot and A. Garcia, "The case for FEC-based error control for packet audio in the Internet," To appear, *ACM/Springer Multimedia Systems*, 1999.

[32] I. Rhee and S. Joshi, "Error recovery for interactive video transmission over the Internet," *IEEE J. Selected Area Comm.*, vol. 18, no. 6, pp. 1033–49, June 2000.

[33] P. Chou, A. Mohr, A. Wang, and S. Mehrotra, "FEC and pseudo-ARQ for receiver-driven layered multicast of audio and video," Submitted, *IEEE Data Compression Conference*, Mar. 2000.

[34] A. Mohr, E. Riskin, and R. Ladner, "Graceful degradation over packet erasure channels through forward error correction," in *IEEE Data Compression Conference '99*, Snowbird, UT, Mar. 1999.

[35] T. Zhang and Y. Xu, "Unequal packet loss protection for layered video transmission," *IEEE Trans. Broadcasting*, vol. 45, no. 2, pp. 243–52, June 1999.

[36] D. Mandelbaum, "An adaptive-feedback coding scheme using incremental redundancy," *IEEE Trans. Info. Theory*, vol. 20, no. 3, pp. 388–9, May 1974.

[37] W. Tan and A. Zakhor, "Multicast transmission of scalable video using receiver-driven hierarchical FEC," in *Packet Video Workshop '99*, New York, NY., Apr. 1999.

[38] B. Vickers, M. Lee, and T. Suda, "Feedback control mechanisms for real-time multipoint video services," *IEEE J. Selected Area Comm.*, vol. 15, no. 3, pp. 512–30, Apr. 1997.

[39] P. Chou and K. Ramchandran, "Clustering source/channel rate allocations for receiver-driven multicast with error control under a limited number of streams," Submitted, *IEEE Intl. Conf. Multimedia Expo.*, July 2000.

[40] X. Li, S. Paul, and M. Ammar, "Multi-session rate control for layered video multicast," in *Proc. SPIE Multimedia Computing and Networking*, San Jose, CA., Jan. 1999, vol. 3654, pp. 175–89.

[41] R. Gopalakrishnan, J. Griffioen, G Hjálmtýsson, C. Sreenan, and S. Wen, "A simple loss differentiaion approach to layered multicast," To appear, *Proc. IEEE Infocom 2000*.

[42] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behaviour of the TCP congestion avoidance algorithm," *ACM Computer Comm. Review*, vol. 27, no. 3, pp. 67–82, July 1997.

[43] D. Sisalem and A. Wolisz, "Mlda: TCP-friendly congestion control framework for heterogeneous multicast environments," in *IEEE Intl. Workshop on QoS*, Pittsburgh, PA., June 2000, pp. 65–74.