Fine-Grained Layered Multicast

John Byers, Michael Luby, Michael Mitzenmacher

Abstract— **Traditional approaches to receiver-driven layered multicast have advocated the benefits of cumulative layering, which can enable coarse-grained congestion control that complies with TCP-friendliness equations over large time scales. In this paper, we quantify the costs and benefits of using *non-cumulative* layering and present a new, scalable multicast congestion control scheme which provides a fine-grained approximation to the behavior of TCP additive increase / multiplicative decrease (AIMD). In contrast to the conventional wisdom, we demonstrate that fine-grained rate adjustment can be achieved with only modest increases in the number of layers and aggregate bandwidth consumption, while using only a small constant number of control messages to perform either additive increase or multiplicative decrease.**

Keywords—**Reliable multicast, congestion control, TCP-friendliness, Fibonacci sequences, non-cumulative layering.**

I. INTRODUCTION

One of the significant challenges associated with multicast delivery to large audiences is providing a scalable congestion control mechanism that is compliant with TCP yet addresses heterogeneity in end-to-end bandwidth across receivers. Recently, the technique of *layered* multicast, which employs multiple multicast groups to transmit content at different rates, has been employed as a strategy capable of accommodating diverse sets of receivers. A novel instantiation of this approach, receiver-driven layered multicast, was advocated by McCanne, Jacobson and Vetterli [12] as a mechanism for addressing receiver heterogeneity in the context of packet video transmission. Their approach enforces *cumulative layering*, which imposes an ordering on the multicast layers and requires clients to subscribe and unsubscribe to layers in sequential order. In the context of appropriately encoded packet video transmissions, subscription to each additional layer in a cumulative organization provides improvements in either frame rate or picture quality. The complexity of the encoding process and a desire to keep the number of layers manageable motivates the following natural and widely-used rate allocation scheme:

The multicast group associated with the base layer transmits at a rate B_0 and all other layers i transmit at rate $B_0 * 2^{i-1}$.

In such an allocation, subscribing to an additional layer doubles a receiver's effective reception rate; similarly, leaving a layer halves the reception rate. While congestion control in the context of a cumulative layered organization is possible, it is necessarily coarse-grained. This is in contrast to TCP, which employs additive increase / multiplicative decrease (AIMD) to achieve fine-grained congestion control. However, researchers have demonstrated that if the frequency of join and leave attempts is carefully orchestrated across cumulative layers, it is possible to achieve long-term rates that closely approximate the functional relationship between throughput and loss rates that TCP achieves. This relationship, which is called TCP-friendliness, is an increasingly commonly used metric for parameterizing and evaluating congestion control schemes [18], [4], [6], [2], [1].

A cumulative, layered organization has also recently been proposed for reliable multicast of bulk data [18], [3]. In reliable multicast, the key challenge is to minimize the number of redundant packets that arrive at any receiver, even as receivers dynamically and asynchronously perform join and leave operations. Early work in this area addressed these problems by combining judicious use of Reed-Solomon forward error correction techniques together with careful organization of packet transmissions across layers [18]. Subsequent work described a digital fountain model [3] which motivated the use of and employed new forward error correcting codes which can efficiently generate a virtually unbounded number of encoding packets [10]. This unbounded encoding eliminated the need for complex packet scheduling algorithms. Moreover, such a coding strategy can be combined with any layered multicast organization, since subscription to an additional layer simply delivers encoded data more quickly. Furthermore, in contrast to the problem of video transmission, for bulk data transmission there is no longer the requirement that the set of subscription layers be cumulative; each layer has utility independent of any other layer. This motivates consideration of non-cumulative approaches for subscribing to sessions in a layered multicast. Returning to the basic layering described above, it is clear that by using non-cumulative layering, a receiver can subscribe to a set of layers which yields an aggregate rate of jB_0, for any positive integer j between 1 and 2^i, where i is the number of layers. This ability to fine-tune the rate implies that AIMD congestion control at the granularity of B_0 is realizable. Such a scheme is therefore TCP-friendly not only in the sense of achieving the same throughput over large time scales, but also has the advantage that it resembles TCP behavior even over time intervals on the order of a round-trip time.

Of course, relaxing the requirement of cumulative layering does not come without cost. In the naive scheme described above, receivers would have to perform a substantial number of join and leave operations to emulate a step of additive increase in the worst case. Also, when a large number of clients perform uncoordinated joining and leaving through a shared network link in this scheme, considerably more bandwidth will be consumed than by the largest consumer alone. Because of these obvious problems, non-cumulative layering schemes have not been studied; the perception is that they are too complex and too costly. In this paper, we demonstrate that an additive increase, multiplicative decrease multicast congestion control protocol can be realized and implemented with reasonable costs and complexity using novel non-cumulative layering schemes. We emphasize,

John Byers: Boston University, Computer Science Department. This work was done in part while visiting Digital Fountain, Inc. Supported in part by NSF Grant ANI-9986397. E-mail: byers@cs.bu.edu

Michael Luby: Digital Fountain, Inc. E-mail: luby@digitalfountain.com

Michael Mitzenmacher: Harvard University, Computer Science Department. 33 Oxford St., Cambridge, MA 02138. This work was done in part while visiting Digital Fountain, Inc. Supported in part by an Alfred P. Sloan Research Fellowship and NSF CAREER Grant CCR-9983832. E-mail: michaelm@eecs.harvard.edu.

however, that the question of appropriate tradeoffs is complex; hence we view the quantification and close inspection of the costs and benefits of non-cumulative layering as a major contribution of our work.

The remainder of this paper is organized as follows. In Section II, we survey the large body of related work in the area. In Section III, we provide a comparative assessment of various natural cumulative and non-cumulative layering schemes and the performance metrics we propose to analyze them. Then, in Section IV, we design novel layering sequences designed from Fibonacci sequences which are highly optimized for additive increase / multiplicative decrease congestion control for reliable multicast. In Sections V and VI, we design and give results from packet-level *ns* simulations to demonstrate the effectiveness of our approach, the potential benefits, and the friendliness with TCP traffic.

II. Previous work

A fundamental challenge of multicast congestion control is to define a protocol that is both scalable and compatible with currently extant congestion protocols, especially TCP. In this section, we survey the large body of recent work in this area. The technique of cumulative layered multicast was first proposed by McCanne, Jacobson and Vetterli [12] in the context of packet video transmission to large heterogeneous audiences. Their approach achieves scalability by using a *receiver-driven* approach, in which the hosts tune their subscription level by joining and leaving layers. Packet loss during normal transmission induces hosts to drop a layer; periodic join experiments to the next highest layer allow hosts to increase their rates in the absence of packet loss. One drawback of this approach is that one host's join experiments can introduce packet loss at other hosts, necessitating complex and undesirable coordination across hosts when performing join experiments. The difficulties associated with coordinating join and leave attempts motivated Vicisano, Rizzo and Crowcroft to propose their Receiver-driven Layered Congestion Control (RLC) algorithm[18]. Their approach called for synchronized join experiments, where the sender would temporarily increase the sending rate on a layer and the receiver would join a higher layer only if there was no packet loss during this experiment. One goal of this approach was to avoid the problem of long IGMP leave latencies by ensuring that a receiver joined a higher layer only if there appeared to be sufficient available bandwidth in the system. Their work also demonstrates that under the idealized conditions seen in simulation their algorithm is TCP-friendly.

In addition to the RLC algorithm, there are several other papers which have focused on TCP-friendly multicast congestion control. This line of research began with work which provided models for characterizing TCP throughput as a function of the round-trip time and the steady state packet loss rate [15], [4], [6]. This work led to proposals for equation-based congestion control [15], and the TCP-Friendly Rate Control (TFRC) protocol [6], which can be applied to multicast applications [9], or unicast applications which are not tolerant to bandwidth fluctuations [1]. In [2], the authors develop FLID/DL, a TCP-friendly congestion control scheme for layered multicast which generalizes the RLC scheme to avoid abrupt rate increases and addresses the challenge of large IGMP latencies, which make leave operations difficult to use in a timely fashion. This second issue is addressed in [2] by introducing the idea of *dynamic layers*. In this framework, a leave operation in a standard layering scheme can be simulated by the passive response of taking no action, while maintaining or increasing a reception rate requires joining new dynamic layers to make up for the fact that some dynamic layers may be decreasing their sending rate. With dynamic layers, the asymmetry between join and leave latencies under IGMP can be avoided; all operations take place at the speed of IGMP join operations, which are efficient. We note that using dynamic layers does introduce a cost in that it multiplies the number of layers required by a constant factor over static schemes. A central assumption we will use in this paper is that a receiver can join and/or leave a small number of layers efficiently at any time – one way to realize this is with dynamic layers.

In parallel with these innovations in receiver-driven layered congestion control, work on integrating forward error correction into layered multicast was emerging as an end-to-end solution for scaling *reliable multicast* to heterogeneous audiences. Work by [17], [18] and [13] demonstrated that Reed-Solomon codes could be used to provide protection against packet loss and described how to layer transmissions in a layered multicast session to reduce the likelihood of a host receiving redundant transmissions. Byers, Luby, Mitzenmacher and Rege [3] advocated an approach based on the much faster, more powerful Tornado codes [11], and introduced the concept of fast FEC codes which are capable of generating a virtually unbounded amount of forward error correction. The LT codes that are described in [10] provide a realization of this concept. This obviates the need for complex packet scheduling algorithms over layers. Finally, the Breadcrumb Forwarding Service model proposed in [19], motivates an architecture supporting receiver-driven requests at a fine granularity, i.e. at a packet level. The authors describe how this service model can be coupled with source-specific multicast (SSM) [8] and fast FEC codes described earlier to achieve congestion-controlled, reliable multicast. However, this approach requires major revamping of networking infrastructure to deploy.

III. Setup and Notation

We consider the problem of allocating rates to the set of multicast sessions in a layered multicast group so as to enable receiver subscriptions in the range $[1, R]$. Note that, by scaling and translating, our solutions then also apply to rates in the range $[B_0, cR + B_0]$ for any c and B_0. Because it is not necessarily clear a priori which parameters prove the most important in relaxed layering schemes, we proceed by considering examples, beginning with the standard cumulative scheme.

A. Cumulative Layering

Several metrics to quantify the resource requirements and performance of a layered multicast scheme are immediately apparent from considering the basic cumulative layering scheme introduced earlier. This layering scheme transmits on the base layer 0 at normalized rate 1 and transmits across all other layers $i \geq 1$ at rate 2^{i-1}. With the requirement of cumulative layering, each receiver can subscribe to layer $i \geq 1$ if and only if they subscribe to all layers j where $0 \leq j < i$. Two useful factors to consider in evaluating such a scheme are the number of multicast groups needed to span a given range of reception rates and the granularity with which a receiver can tune its rate within that range. The definitions below express those considerations.

Definition 1: The *density* of a layering scheme S which supports reception rates in the range $[1, R]$ is the number of multicast groups that the scheme employs as a function of R.

The density of a layering scheme is a measure of its scalability, as it is currently infeasible and undesirable to employ a large number of multicast groups to satisfy receivers of a single layered multicast session. As a rule of thumb, we view schemes whose density scales as a polynomial in R as unscalable, and schemes with logarithmic density in R as desirable.

Definition 2: For a layering scheme S which supports reception rates in the range $[1, R]$, and for $i \in [1, R]$, let A_i be the maximum rate achievable by S which satisfies $A_i \leq i$. The *reception granularity* of such a scheme is then defined to be

$$\max_{i \in [1,R]} \frac{i}{A_i}.$$

A reception granularity of 1 is optimal, and admits the possibility of fine-grained congestion control at the granularity of the base layer bandwidth. As mentioned in the introduction, layering schemes which have reception granularities $g > 1$ can only employ coarse-grained congestion control, since fine-grained rate adjustment is not possible in general. This factor is the primary motivation for the set of schemes which we consider momentarily.

Applying the first definition to the basic scheme above, it follows directly that its density is $\lceil \log_2 R \rceil + 1$. Considering the second definition, it is apparent that exactly those rates which are a power of two can be realized by the basic scheme, thus the reception granularity is at most two, and is in fact marginally better, i.e. $2 - \frac{1}{R}$.

Before moving to non-cumulative schemes, we mention a natural, but problematic, method for achieving fine-grained control with cumulative layering: allowing each layer to send at the rate $B_i = 1$; that is, we use a number of layers equal to the maximum allowable bandwidth, each sending at an equal rate. While the reception granularity of such a scheme is optimal (1), the density of the scheme is linear in R and is therefore unscalable.

Similarly, the reception granularity could naturally be reduced by modifying the transmission rates of the layers of a cumulative layering scheme. For example, for any $c > 1$, we may set $B_0 = 1$, $B_1 = c - 1$, and $B_i = c^i - c^{i-1}$. In this case each additional layer increases the total received bandwidth by a factor of c. The reception granularity is therefore bounded above by c, although for $c < 2$ the density increases to $\lceil \log_c R \rceil + 1$.

B. Relaxing Cumulative Layering

A more compelling possibility for reducing the reception granularity is to relax the requirement that a receiver must join a set of cumulative layers. For example, with the standard allocation in the basic scheme, all of the integral rates in the range $[1, R]$ can be achieved once we drop the cumulative requirement. (For convenience, we will assume that subscribing to the base layer is still mandatory.) This scheme has logarithmic density and optimal reception granularity; however, it is not clear how to efficiently implement additive increase and multiplicative decrease with this scheme, since those operations may require a large number of multicast joins and leaves. For example, suppose a receiver is subscribed to the first four layers $[1, 1, 2, 4]$, and therefore has a reception rate of eight. To achieve a reception rate of nine, the receiver must join one layer and leave three layers. Similarly, a receiver subscribed to layers $[1, 2, 8, 32]$ can halve their rate only by joining and leaving several layers to reach $[1, 4, 16]$. Even assuming join and leave operations can be performed efficiently, to minimize the significant impact of processing multicast control traffic at routers we wish to keep the number of such operations as small as possible. This motivates the following definitions:

Definition 3: The *join complexity* (respectively *leave complexity*) of additive increase under a layering scheme S is the worst case number of multicast join messages (respectively leave messages) a receiver must issue to increase its rate by B_0. Similarly, the join/leave complexity of multiplicative decrease under a layering scheme S is the worst case number of multicast join messages (respectively leave messages) a receiver must issue to decrease its rate by the relevant multiplicative factor.

Another significant problem of non-cumulative schemes is the need for extra bandwidth to accomodate receivers, which the following example illustrates. Consider two receivers R_1 and R_2 who share a bottleneck link L and wish to receive at rates 9 and 4, respectively. In the cumulative setting, R_1 must settle for a reception rate of 8, which it can achieve by subscribing to the first four layers $[1, 1, 2, 4]$. Meanwhile R_2 can achieve its target rate by subscribing to the first three layers $[1, 1, 2]$. Since R_2 subscribes to a subset of the layers that R_1 subscribes to, the demand on link L is identical to that placed by R_1.

But in a non-cumulative scenario, R_1 can now subscribe to layers one and five to achieve its target rate exactly, while R_2 still subscribes to the first three layers. This increases the end-to-end rate perceived by R_1 by a single unit, yet the load on link L now jumps from eight to twelve. The requirement of additional bandwidth is a fundamental consequence of non-cumulative layering and motivate the following definition:

Definition 4: For a layering scheme which supports reception rates in the range $[1, R]$, and for a given link ℓ in a multicast tree, let $M_\ell \leq R$ be the maximum reception rate of the set of receivers downstream of ℓ and let C_ℓ be the bandwidth demanded in aggregate by receivers downstream of ℓ. The *dilation* of link ℓ is then defined to be $\frac{C_\ell}{M_\ell}$. Similarly, the *dilation* imposed by a multicast session is taken to be $\max_\ell \frac{C_\ell}{M_\ell}$.

In the example above, the dilation of L was 1 in the cumulative case and $\frac{12}{9}$ in the non-cumulative case. In general, cumulative layering enforces a guarantee that links are never dilated, i.e. have a dilation of 1. The worst-case dilation imposed by the basic non-cumulative layering scheme grows to 2. In fact, the worst case is when one receiver is subscribed to just the base layer and the highest layer, and another is subscribed to the base layer and all other layers except the highest layer; hence the worst case dilation can be shown to be $2 - \frac{4}{R+2}$.

We seek non-cumulative layered schemes that have low reception granularity, dilation, and join/leave complexity. As a preview, we consider our results as compared with the standard cumulative scheme and the derived non-cumulative scheme where layer sizes increase geometrically by a factor of two in Figure III-B. Our main result is a scheme that achieves a low join/leave complexity and a lower dilation than the basic non-cumulative scheme with only a small increase in the number of layers.

IV. Schemata

A. A Fibonacci-based scheme

We now provide an example of a non-cumulative layering scheme that meets many of our desiderata.

Definition 5: The layering scheme *Fib1* is defined by $B_0 = 1$, $B_1 = 2$, and $B_i = B_{i-1} + B_{i-2} + 1$ for $i \geq 2$.

It will be useful to extend the Fib1 layering scheme by implicitly defining B_{-2} and B_{-1} to be zero, so the recurrence $B_i = B_{i-1} + B_{i-2} + 1$ holds for $i \geq 0$. The first few rates of the layers for Fib1 are

$$1, 2, 4, 7, 12, 20, 34, \ldots.$$

The sequence B_i is obviously similar to the Fibonacci numbers. Indeed, let the Fibonacci numbers be given by $F_0 = 1, F_1 = 1$, and $F_i = F_{i-1} + F_{i-2}$. Then a simple induction yields $B_i = F_{i+2} - 1$. It is for this reason that we call the B_i layer sequence Fib1.

Our motivation for studying the Fib1 sequence of layers is that it easily admits additive increase. Increasing the receive rate by one unit can be achieved by the following procedure:

Increase by 1: Choose the smallest layer $i \geq 0$ to which the receiver is not currently subscribed; then subscribe to layer i and unsubscribe from layers $i-1$ and $i-2$.

The increase by 1 rule increases the reception rate by $B_i - B_{i-1} - B_{i-2} = 1$. (Note that we may always think of a receiver as always subscribing to the empty layers -1 and -2 for the purposes of the rule, so the rule can always be applied to any non-negative layer.) Hence the reception granularity is the optimal value 1, and the complexity of the additive increase operation is thus at most just one join and two leave operations[1].

To analyze the density of Fib1, recall that the Fibonacci numbers satisfy

$$\begin{aligned} F_i &= \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{i+1} - \left(\frac{1-\sqrt{5}}{2}\right)^{i+1}\right] \\ &\approx \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{i+1}. \end{aligned}$$

Let $\tau = \frac{1+\sqrt{5}}{2} \approx 1.62$. (This is generally called the golden ratio.) Equation 1 implies that in order to handle transmission rates in the range $[1, R]$, Fib1 requires a density of at most $\ell = \log_\tau R$ layers, instead of the $\log_2 R$ layers for the standard cumulative scheme. Hence using Fibonacci layering maintains the desired property that the density is logarithmic in the maximum bandwidth R.

A further question is to find a convenient method for a multiplicative decrease of the transmission rate. Exactly halving the rate, as is done with the cumulative layering scheme and generally with TCP, might require modifying joining or leaving several layers. If we relax this requirement, so that we are only required to *approximately* halve the receiver rate, then other, simple approaches are available to us. For example, in Fib1, a receiver can approximately halve its reception rate by unsubscribing from its highest subscription layer.

Using this decrease approach, we can prove the following lemma describing the structure of valid subscription levels. To describe this structure, it is useful to express a receiver's subscription level in binary notation. For example, to denote that a receiver is subscribed to layers 0, 1, 3, and 5, we write 101011, with the base layer subscription as the rightmost bit.

Lemma 1: The sequences achieved in the Fib1 layering scheme when starting from 1 and repeatedly increasing by 1 have the following form:

Starting from the first one on the left, all runs of zeroes are only one or two long; if there is a run of zeroes that is two long, there are no further zeroes to the right.

Proof: The lemma follows by induction. ■

In the binary representation, this corresponds to removing the leftmost one from the binary representation of the subscribed layers. Although this does not yield an exact halving of the transmission rate, it necessitates leaving only one layer.[2] Let us consider the impact of such a decrease operation more carefully in the context of leaving the jth layer.

Lemma 2: Suppose a receiver unsubscribes from the highest subscribed layer j using the Fib1 scheme. Then the reception rate decreases by a factor that is bounded above by $1/\tau$. When j is sufficiently large, the reception decreases by a factor that is bounded below by $1/\tau^2 - \epsilon$ for any constant $\epsilon > 0$.

Proof: We may bound the factor by which the rate decreases as follows. The ratio between the new rate and the previous rate is maximized when the new rate is as large as possible; that is, when the receiver also subscribed to all lower layers. In this case the ratio between rates is

$$\frac{\sum_{i=0}^{j-1} B_i}{\sum_{i=0}^{j} B_i} = \frac{\sum_{i=0}^{j+1} F_i - j - 2}{\sum_{i=0}^{j+2} F_i - j - 3} = \frac{F_{j+3} - j - 3}{F_{j+4} - j - 4}.$$

(The last equality uses the identity $\sum_{i=0}^{j} F_i = F_{j+2} - 1$.) Through a tedious induction which we skip here, we find that this ratio is increasing in j. Hence this ratio is upper bounded by the limiting value of the ratio, $\frac{1}{\tau}$.

Similarly, the ratio between the previous rate and the new rate is minimized when the new rate is as small as possible. Adding these operations for multiplicative decrease and increase does not change the result of Lemma 1. Hence from Lemma 1 the minimum possible value when the highest layer subscribed to is layer j is given by the binary representation $100111\ldots$. In this case the ratio from leaving the jth layer is

$$\frac{\sum_{i=0}^{j-3} B_i}{B_j + \sum_{i=0}^{j-3} B_i} = \frac{F_{j+1} - j - 1}{F_{j+2} + F_{j+1} - j - 2}$$

[1] Of course, decreasing the rate by one is accomplished simply by inverting the corresponding increase operation, and hence requires two joins and one leave.

[2] Similarly, we may approximately double the rate using a single join operation.

Sequence	Density	Reception Gran.	Dilation	Add. Increase	Mult. Decrease
Std. Cum	$\log_2 R$	2	1	N/A	1 leave
Std. NonCum	$\log_2 R$	1	2	O($\log R$)	O($\log R$)
Ideal	O($\log R$)	1	1	O(1)	O(1)
Our result	$\log_{1.6} R$	1	1.6	2 joins, 1 leave	1 leave

Fig. 1. Performance of various layering schemes.

$$= \frac{F_{j+1} - j - 1}{F_{j+3} - j - 2}.$$

Hence, for large j, the ratio approaches $\frac{1}{\tau^2}$. ■

For small values of j, the decreases can be larger; for example, when we are subscribed to layers 1001, dropping the top layer reduces the rate from 8 to 1. If the possibility of decreasing the rate too quickly at low levels is a concern, the problem can be ameliorated somewhat by changing the decrease rule to use more leaves and joins. Another alternative which we recommend is to handle situations at the lowest levels with explicit cases – this is also useful in the context of emulating TCP slow start, as mentioned in Section V.

By similar methods, we may bound the dilation associated with use of sequence Fib1.

Lemma 3: Suppose that in a layered multicast session using the Fib1 scheme, the maximum subscription level is up through the jth layer. For j sufficiently large, the dilation imposed by the session is then bounded above by $\tau + \epsilon$ for any constant $\epsilon > 0$.

Proof: Let us suppose the highest layer subscribed to by any downstream receiver is the jth layer. Then the maximum total volume of traffic through the router is $\sum_{i=0}^{j} B_i$, but the receiver obtaining the most traffic receives at a rate of at least $\sum_{i=0}^{j} B_i - B_{j-1} - B_{j-2} = \sum_{i=0}^{j-1} B_i + 1$. Hence the dilation is bounded above by

$$\frac{\sum_{i=0}^{j} B_i}{\sum_{i=0}^{j-1} B_i + 1} = \frac{\sum_{i=0}^{j+2} F_i - j - 3}{\sum_{i=0}^{j+1} F_i - j - 1} = \frac{F_{j+4} - j - 4}{F_{j+3} - j - 2}.$$

Again, this is decreasing in j, and hence it approaches τ for large j, although it can be larger when the maximum receive rate is small. ■

In fact the dilation converges quite rapidly to τ, as we will demonstrate in Section VI, so in practice we may say that the worst-case dilation is essentially τ.

B. Other sequences

Given the behavior of Fib1, it is natural to ask if there are other sequences that have a reception granularity of one but allow different tradeoffs between the density, dilation, and join and leave complexity. In fact the sequence Fib1 is just an example of a large class of possible sequences that might be useful for non-cumulative layering. The best sequence may therefore depend on the system goals and requirements. One fundamental tradeoff present in all fine-grained Fibonacci-based layering schemes is that using fewer layers leads to greater dilation. Another tradeoff is that by allowing receivers to send more control messages per increase or decrease operation, one has more flexibility in setting the approximate multiplicative decrease factor. In general, the tradeoffs associated with using alternative sequences can be quite complex and are best explained via examples.

Definition 6: The layering scheme *Fib2* is defined by $B_0 = 1$, $B_1 = 2$, $B_2 = 3$ and $B_i = B_{i-1} + B_{i-3} + 1$ for $i \geq 3$.

Again, it will be useful to extend the definition of Fib2 by implicitly defining $B_i = 0$ when $i < 0$, so the recurrence $B_i = B_{i-1} + B_{i-3} + 1$ holds for $i \geq 0$. The first few layer rates for Fib2 are

$$1, 2, 3, 5, 8, 12, 18, 27, \ldots,$$

Let G_i be the sequence defined by $G_i = G_{i-1} + G_{i-3}$, with $G_0 = G_1 = G_2 = 1$. The G_i are an example of a *generalized Fibonacci sequence*. Then simple inductions yield that $B_i = G_{i+3} - 1$ and $\sum_{i=0}^{k} G_i = G_{i+3} - 1$. Using these facts, we may analyze Fib2 in a manner similar to Fib1.

We summarize the important points of comparison for Fib2 and Fib1. First, Fib2 grows more slowly, so more layers will be necessary; that is, Fib2 has larger density. We can determine the behavior of the B_i by considering the generalized Fibonacci sequence G_i. The characteristic polynomial for the recurrence of the G_i is $x^3 - x^2 - 1 = 0$. This polynomial has three roots, r_1, r_2, and r_3; and G_i can be expressed as $G_i = c_1 r_1^i + c_2 r_2^i + c_3 r_3^i$ for some constants c_1, c_2, and c_3. By Descartes' rule of signs, there is exactly one real root, and it is positive. It is clear that this root must be larger than 1. Since the product of the three roots is the constant term 1 from the polynomial $x^3 - x^2 - 1$, the other two complex roots must have magnitude less than 1. Hence, if we let σ be the unique real root of the polynomial $x^3 - x^2 - 1 = 0$, then B_i grows approximately like $c\sigma^i$ for some constant c. Note $\sigma \approx 1.466$.[3] The Fib2 scheme therefore has density approximately $\log_\sigma R$; in fact, this is an upper bound. Although the density is larger than that of the Fib1 scheme, it is still only logarithmic in R.

In return for a larger density, the Fib2 scheme has a smaller dilation. When the highest subscription layer grows large, the dilation approaches σ, which is slightly better than the dilation of τ for the Fib1 scheme. The complexity of an additive increase is still just one join and two leave operations. If we implement a multiplicative decrease as we did in Fib1, i.e. by dropping the highest subscribed layer, the rate drop is bounded above by $1/\sigma$, and as the number of layers grows large, the largest rate drop approaches $(\sigma + 1)/\sigma^4$.

Similar patterns requiring a larger number of layers but with a smaller bandwidth expansion ratio can be found by considering recurrences of the form $B_i = B_{i-1} + B_{i-k} + 1$ for some constant k. Sequences of this form all have the property that the complexity of an additive increase is just one join and two

[3] Calculations reveal $\sigma = \frac{1}{3}[1 + \frac{1}{2}(116 + 12\sqrt{93})^{1/3} + \frac{2}{(116+12\sqrt{93})^{1/3}}]$.

leave operations. They also all have the property that the density is logarithmic in the maximum reception rate R. Indeed, the larger the value of k, the smaller the rate at which the bandwidth grows over layers. Hence, the larger the value of k, the larger the density, but the smaller the dilation. Also, if we use the same approach of leaving the highest subscribed layer to implement an approximate multiplicative decrease, as k increases the factor by which the reception rate falls decreases. Note that if leaving the highest subscribed layer is insufficiently aggressive, then the operation can be enhanced by possibly leaving two layers, slightly increasing the complexity of the multiplicative decrease operation.

Another possibility we consider is to allow three (or more) join operations in the additive increase operation.

Definition 7: The layering scheme *Fib3* is defined by $B_0 = 1$, $B_1 = 2$, $B_2 = 4$ and $B_i = B_{i-1} + B_{i-2} + B_{i-3} + 1$ for $i \geq 3$.

Again, implicitly we let $B_i = 0$ if $i < 0$. Let H_i be the sequence defined by $H_i = H_{i-1}+H_{i-2}+H_{i-3}$, with $H_0 = H_1 = H_2 = 1$. A simple induction yields that $B_i = (H_{i+3} - 1)/2$, which again makes Fib3 easy to analyze. The characteristic polynomial for the recurrence of the H_i is $x^3 - x^2 - 1 = 0$. This polynomial has one positive real root γ with $\gamma \approx 1.839$, and two complex roots with magnitude smaller than 1. The Fib3 scheme therefore has density of approximately $\log_\gamma R$, but the density in the worst case decreases to γ.

One may consider similar generalizations given by recurrences of the form $B_i = (\sum_{j=i-k}^{i-1} B_j) + 1$. Interestingly, in the limiting case as $k \to \infty$, we obtain the standard layering scheme, where the layers double in size. Of course one may consider schemes of various forms similar to both Fib2 and Fib3, based on recurrences such as $B_i = B_{i-1} + B_{i-3} + B_{i-5}$. We expect, however, that such general recurrences are of limited practical interest.

V. Non-Cumulative Congestion Control Algorithms

The non-cumulative layering sequences we have introduced are well suited to fine-grained congestion control and specifically enable additive increase / multiplicative decrease at the receiver. In our algorithm, receivers infer the level of congestion in the network either by observed packet loss or by explicit congestion notification (ECN) and adjust their subscription levels accordingly, i.e. in the manner of TCP. Ideally, our adjustment algorithm would be faithful to TCP additive increase / multiplicative decrease, and achieve TCP-friendliness both on short time scales and match the functional relationship between packet loss rate and throughput achieved by TCP on longer time scales. To that end, each multicast receiver autonomously attempts to match the rate it would receive across a TCP congestion-controlled stream.

TCP's congestion control mechanism can be summarized as follows: when packet loss is experienced, halve the congestion window; otherwise, if no congestion is experienced in a round-trip time, increase the congestion window by a single packet. Obviously this is a vast oversimplification of a complex control protocol, but it is at this granularity that we intend to emulate TCP behavior.

In deriving congestion control protocols which are TCP-friendly, it is also worth recalling the formula for $\hat{T}$, an approximation of the TCP throughput rate T. Here $\hat{T}$ is given in units of packets per second as a function of the packet loss rate p and the TCP round trip time R: [15], [6]:

$$\hat{T} = \frac{\sqrt{1.5}}{R\sqrt{p}}. \tag{1}$$

This equation is easily derived by approximating TCP behavior as a deterministic sawtooth. That is, given a maximum sending window size W, a TCP stream will additively increase its sending window by one packet per round trip until it reaches W, at which point its sending window is cut in half to size $W/2$. The throughput and the loss rate are determined by the window size W; this yields a functional relationship between the two that matches equation 1.

In fact, this relationship is naturally generalized to variants where the sending window is increased by a packets once every R seconds and cut by a factor $(1 - b)$ in case of a packet loss. Such variants are called AIMD(a, b) congestion control, and are analyzed in [5]. The appropriate generalization of equation 1 in this scenario is

$$\hat{T} = \frac{\sqrt{2-b}\sqrt{a}}{R\sqrt{2bp}}. \tag{2}$$

We will use this equation to determine appropriate settings for our fine-grained multicast congestion control scheme.

Before proceeding, we note that in the context of multicast, it is often impossible or impractical to estimate the RTT as accurately as TCP can; moreover, there are fundamental limitations to the extent to which TCP-compatible fairness can be achieved when the RTT cannot be measured accurately [7]. Therefore, the rate at which we perform additive increase should be thought of as an "aggressiveness" parameter which equates performance to that of a TCP connection with a specific RTT. To make this correspondence clear, we now derive a formula equating TCP additive increase with non-cumulative multicast increase.

In our setting, we choose an aggressiveness parameter Q. Every Q seconds the transmission rate increases by a fixed amount B_0 when there is no packet loss, where here the units of B_0 are in packets per second. In the event of packet loss, the transmission rate decreases by a multiplicative factor that is approximately $1/\tau$, as analyzed in Section IV. Hereafter we treat the multiplicative decrease factor as fixed. Hence our congestion control scheme mimics an AIMD$(QB_0, 1 - \frac{1}{\tau})$ scheme, so the throughput rate $\hat{T}^*$ using equation 2 is

$$\hat{T}^* = \frac{\sqrt{1+\frac{1}{\tau}}\sqrt{B_0}}{\sqrt{2(1-\frac{1}{\tau})Qp}}.$$

To equalize the long-term throughput, we equate $\hat{T}$ for a standard TCP stream and $\hat{T}^*$ for our multicast scheme under the same loss rate p. That is, we picture our multicast stream and our TCP stream as sharing a bottleneck link, so that they experience the same overall loss rate. So given a base layer rate B_0 and a target TCP round trip time RTT, we wish to solve for Q so that

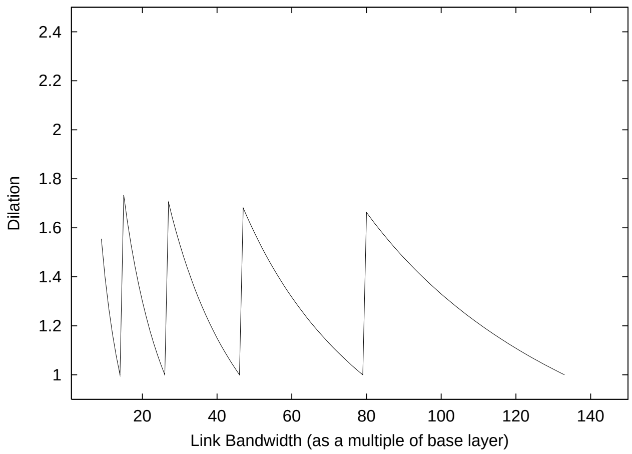
$$\frac{\sqrt{1+\frac{1}{\tau}}\sqrt{B_0}}{\sqrt{2(1-\frac{1}{\tau})Qp}} = \frac{\sqrt{1.5}}{R\sqrt{p}}.$$

After simplifying,

$$Q = R^2 \frac{(\tau+1)B_0}{3(\tau-1)}. \quad (3)$$

As an example, if the TCP RTT is 500ms, the packet size is 512 bytes, and the base bandwidth is 4 packets per second (16Kbps), then setting $Q = 1.412$ seconds yields TCP-friendly throughput. Note that the frequency and granularity of rate increase in our multicast session is likely to differ from that of a competing TCP flow; however, from Equation (3), in the event that $Q = R$ and $B_0 = \frac{3\tau-1}{(\tau+1)R}$ the multicast session's estimate of the RTT matches that used by the TCP flow.

Another aspect of TCP which we can emulate is TCP slow start. In slow start, TCP doubles the congestion window once per round trip time until congestion is observed. Using the Fib1 scheme, we can efficiently implement multiplicative increase with a single multicast join; however, as noted earlier, this does not achieve an exact factor of two increase. Careful emulation of TCP slow start is beyond the scope of this paper; we expect that incorporating this into non-cumulative layered multicast would require careful design of extra layers specifically included for this reason.



.

Fig. 2. Maximal dilation at a link as a function of available link bandwidth

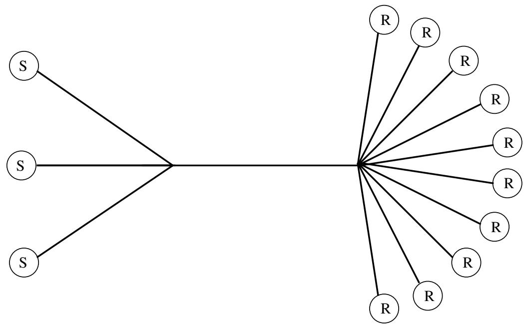


Fig. 3. Our basic experimental topology.

VI. Experiments

A. Static sharing ratios

An important consideration is the bandwidth consumed by a set of multicast clients behind a shared bottleneck. Returning to the model in which the available bandwidth through the bottleneck is in the range $[1, R]$ (for an appropriate rescaling), we observe that the bandwidth expansion ratio is a function only of the bottleneck bandwidth. Recall that during additive increase in Fib1, a client will at some point subscribe to a new maximum layer j and unsubscribe from layers $j-1$ and $j-2$ across the bottleneck. When a client crosses one of these *transition points*, it can cause a significant increase in bandwidth consumption through the bottleneck.

Just prior to crossing a transition point j, the client was subscribing to all layers less than j. At this instant, we reduce to a scenario similar to that present in a cumulative scheme, where the bandwidth expansion ratio is one, as all clients with lesser or equal allocations through the link are subscribing to a subset of the client nearing the transition. Therefore, if we take the bottleneck link bandwidth to be a fixed, static constant, the bandwidth expansion ratio of that fixed link can be computed directly. We plot the values of that ratio in Figure 2 in the context of sequence Fib1, and where the spikes of the resulting sawtooth correspond to transition points measured in the bandwidth space.

B. Fairness

In the remainder of the paper, we describe our experimental results derived from packet-level simulations with the network simulator ns [14]. The primary objective of these simulations is to demonstrate inter-session fairness and fairness with TCP flows with representative background traffic. A secondary objective is to depict the essential differences between coarse-grained congestion control achieved by cumulative layered multicast and the fine-grained congestion control that we are able to achieve.

Our experimental work compares three protocols: the TCP Reno implementation available within ns, a faithful re-implementation of the RLC protocol defined by Vicisano, Rizzo and Crowcroft [18] compatible with the current release of ns version 2, and our additive-increase, multiplicative-decrease congestion control protocol built from the Fib1 layering sequence, which we implemented in ns. All of our experiments involve single source multicast; we use DVMRP as the multicast routing protocol and to simulate graft and prune messages. The network topology we employ in our simulation is depicted in Figure 3. TCP traffic, multicast traffic and background traffic originate at the nodes on the left-hand portion of the plot and crosses a shared bottleneck en route to receivers on the right-hand side of the bottleneck. This simple topology allows us to consider both intra-session fairness for clients employing non-cumulative congestion control and inter-session fairness across non-cumulative congestion control, other multicast congestion control protocols and TCP. The one-way bottleneck link delay is set to be 40ms and in these initial simulations, the gateways we use are exclusively drop-tail. For all sessions, we have used a packet size of 256 bytes and we have set the gateway queue sizes to 50 packets.

In order to emulate the bursty loss conditions observed in wide-area network traffic, we subjected the bottleneck link in our topology to background traffic resulting from a set of Pareto-distributed ON/OFF constant-bit rate (CBR) UDP flows. We have the ability to modulate the intensity and burstiness of the background traffic by tuning the mean ON and OFF periods, the Pareto shape parameter, the number of flows and the bit rate. Given the buffer sizes above, an appropriate setting to generate bursty traffic which in the expectation consumes roughly half of the bottleneck bandwidth of a 1Mbps link is 20 flows transmitting at 36Kbps, a mean ON time of 2 seconds, a mean OFF time of 1 second and a Pareto shape parameter of 1.2. Extensive experiments with a variety of traffic mixes will be reported in the full version of the paper.

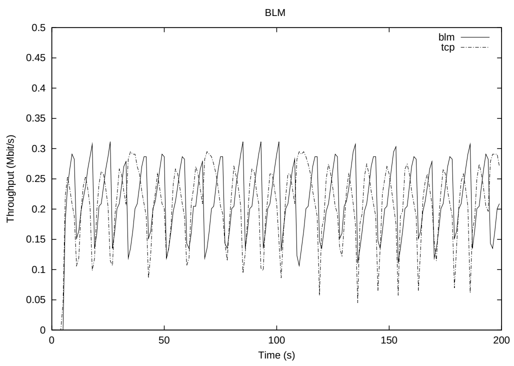


Fig. 5. Fib1 multicast congestion control competing with a TCP flow

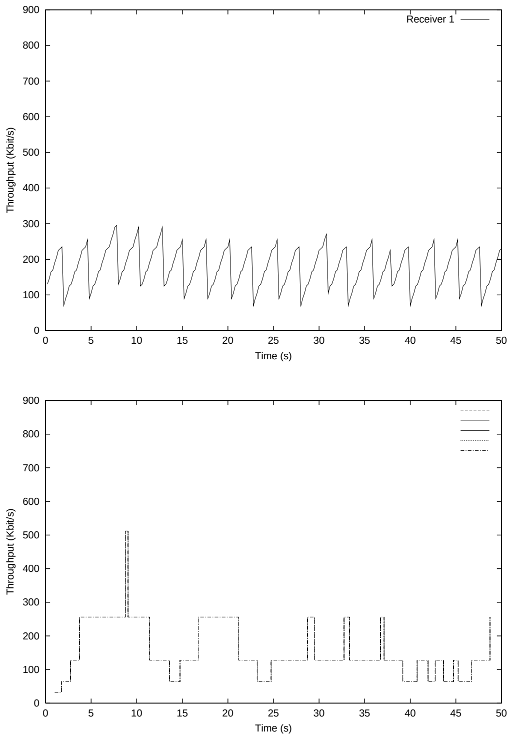


Fig. 4. A comparison of Fib1 (top) and RLC (bottom) multicast congestion control sharing bottleneck bandwidth

We now give an overview of the RLC multicast congestion control protocol. The RLC sender transmits over cumulative layered multicast groups with geometrically increasing rates. A novel aspect of RLC is that at periodic intervals on each layer, the rate briefly doubles. For a receiver subscribing to a layer during this burst, this has the effect of simulating the congestion level the receiver would experience if it were to join the next higher layer. At well-specified synchronization points, the receiver makes a decision. If it received the previous burst of packets without loss, it infers that it may safely join the next higher layer and does so. If it has lost packets outside of the burst, i.e. during normal transmission, it drops a layer; otherwise it does not change its subscription level. The key parameters of this scheme are W, the distance between synchronization points on the base layer, and P, the relative frequency of synchronization points and bursts. In our implementation of RLC, we have employed the parameter settings they suggest: $W = 8$ packets; $P = 1$, and the base bandwidth B_0 is 16Kbps. The authors of RLC assume (as we do) that it is not generally feasible to generate sufficiently accurate estimates of RTTs; the settings above are comparable to and friendly with a TCP connection whose RTT is 100ms.

In the context of our non-cumulative protocol, a packet size of 256 bytes implies that base bandwidth B_0 of 16Kbps is 8 packets per second. Along with a TCP RTT of $R = 100$ms, this leads to an aggressiveness parameter of $Q = 112$ms. In steady-state, we attempt emulate the behavior of TCP without coarse-grained timeouts, i.e. additive increase will occur if there has been no packet loss in the previous eight RTTs; we perform multiplicative decrease half an estimated RTT after packet loss has been detected. After one such decrease, we suppress additional multiplicative decrease for a deaf period of an additional RTT. This general strategy is comparable to those proposed for cumulative layered multicast.

In Figure 4, we present a comparison of a Fib1 receiver and an RLC receiver competing for bottleneck bandwidth in this environment. In the RLC plot, we only plot fluctuations in bandwidth due to multicast joins and leaves; the bandwidth introduced by failed join experiments (which often introduce packet loss for the competing Fib1 session) is averaged in to the steady-state plots. This comparison visually demonstrates the essential difference between fine-grained and coarse-grained congestion control schemes motivating this paper; however, the long-term throughput achieved by these two schemes in the simple topologies we studied was typically comparable.

In a similar vein, Figure 5 provides a comparison between a single TCP flow and a Fib1 receiver. These tests are run with slowly time-varying background traffic absorbing all but approximately 500Kbps of the link bandwidth and running with a TCP RTT of 200ms. The visual comparison between the performance delivered by the two schemes is striking; however, we do note that when the multicast session makes a poor estimate of the TCP round-trip time R, the corresponding choice of Q is also incorrect and the relative performance of the two schemes do not couple as nicely. In this experiment, the average throughput attained by TCP was 217 Kbps and the average throughput attained by our non-cumulative multicast session was 220 Kbps.

Our final experiment considers the question of intra-session fairness. Depending on the timing of session joins and the sequence of packet loss, clients' rates may differ at the loca-

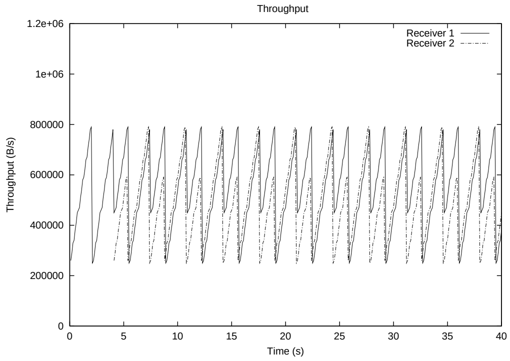


Fig. 6. Asynchronous receivers sharing bottleneck bandwidth

tion of loss events, and the resulting rate of multiplicative decrease may not be identical. In the example presented in Figure 6, two asynchronous receivers compete for 1Mbps bottleneck bandwidth (without cross-traffic). In such a scenario, synchronization effects causing persistent unfairness can occur and the effects of bandwidth dilation can be seen, since neither client reaches the bottleneck bandwidth alone. In spite of these factors, the throughput they achieve remains within 20% of one another, and within a factor of 1.5 of the long-term rate which would be achieved by a TCP flow.

Experiments providing other comparison points regarding the fairness and performance of various unicast and multicast congestion control protocols will be provided in the full version of this paper.

VII. Conclusions

Whereas traditional approaches to receiver-driven layered multicast have advocated the benefits of cumulative layering and have employed such an approach, we argue for a careful study of non-cumulative layered approaches. Non-cumulative layering admits the possibility of fine-grained multicast congestion control and can improve end-to-end performance by closely matching a receiver's demanded rate. Prior to this paper, non-cumulative layering had not been studied, in part because of the difficulty of framing application-level data in a manner compatible with non-cumulative layers, but also because of the perception that the performance penalty is substantial. Advances in fast FEC encoding for reliable multicast [3] and fine-grained rate-adaptive video coding [16] are eliminating the framing problems in the domain of source-specific reliable multicast and packet video. Our work demonstrates that the costs of non-cumulative layering need not be substantial by carefully quantifying the parameters in the layered multicast design space.

Our work goes beyond the metric of TCP-friendliness to define additional appropriate metrics for evaluating the effectiveness of a multicast congestion control scheme from the standpoint of resource utilization and reception granularity. We argue that while standard cumulative approaches make effective use of network resources, they are incapable of performing fine-grained congestion control. In contrast, carefully engineered non-cumulative layering sequences and corresponding congestion control algorithms allow receivers to perform AIMD, TCP-friendly congestion control with quantifiable bounds on performance degradation. Our hope is that non-cumulative layered congestion control can be coupled with existing transport mechanisms to become a viable alternative to current coarse-grained multicast congestion control for a wide variety of multicast applications.

Acknowledgments

We thank Gu-In Kwon for helping us to conduct the *ns* simulations reported in this work and thank the anonymous INFOCOM 2001 reviewers for their helpful comments.

References

[1] D. Bansal and H. Balakrishnan. TCP-Friendly Congestion Control for Streaming Internet Applications. To appear in Proceedings of IEEE INFOCOM '01, April 2001.

[2] J. Byers, M. Frumin, G. Horn, M. Luby, M. Mitzenmacher, A. Roetter, and W. Shaver. FLID-DL: Congestion Control for Layered Multicast. In *Proceedings of NGC 2000*, pages 71–81, November 2000.

[3] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege. A Digital Fountain Approach to Reliable Distribution of Bulk Data. In *Proceedings of ACM SIGCOMM '98*, Vancouver, September 1998.

[4] S. Floyd and K. Fall. Promoting the Use of End-to-End Congestion Control in the Internet. *IEEE/ACM Transactions on Networking*, 7(4), August 1999.

[5] S. Floyd, M. Handley, and J. Padhye. A Comparison of Equation-Based and AIMD Congestion Control. Manuscript available at http://www.aciri.org/floyd/papers.html, May 2000.

[6] S. Floyd, M. Handley, J. Padhye, and J. Widmer. Equation-Based Congestion Control for Unicast Applications. In *Proceedings of ACM SIGCOMM 2000*, August 2000.

[7] S. J. Golestani and K. K. Sabnani. Fundamental Observations on Multicast Congestion Control in the Internet. In *Proceedings of IEEE INFOCOM '99*, pages 990–1000, New York, NY, March 1999.

[8] H. Holbrook and D. Cheriton. IP Multicast Channels: EXPRESS Support for Large-Scale Single-source Applications. Proceedings of ACM SIGCOMM '99, August 1999.

[9] A. Legout and E. Biersack. PLM: Fast Convergence for Cumulative Reliable Multicast Transmission Schemes. Proceedings of ACM SIGMETRICS 2000, June 2000.

[10] M. Luby, J. Gemmell, L. Vicisano, L. Rizzo, J. Crowcroft, and B Lueckenhoff. Asynchronous layered coding: A scalable reliable multicast protocol. Technical report, Work in progress presented in the 47^{th} IETF, Adelaide, Australia, March 2000.

[11] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, and V. Stemann. Practical Loss-Resilient Codes. In *Proceedings of the* 29^{th} *ACM Symposium on Theory of Computing (STOC)*, April 1997.

[12] S. McCanne, V. Jacobson, and M. Vetterli. Receiver-Driven Layered Multicast. In *Proceedings of ACM SIGCOMM '96*, pages 117–130, Stanford, CA, August 1996.

[13] J. Nonnenmacher, E. Biersack, and D. Towsley. Parity-Based Loss Recovery for Reliable Multicast Transmission. In *Proceedings of ACM SIGCOMM '97*, September 1997.

[14] `ns`: UCB/LBNL/VINT Network Simulator (version 2). http://www-mash.cs.berkeley.edu/ns/ns.html.

[15] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP throughput: a simple model and its empirical validation. In *Proc. ACM SIGCOMM*, September 1998.

[16] R. Rejaie, M. Handley, and D. Estrin. Quality Adaptation for Congestion Controlled Video Playback over the Internet. In *Proceedings of ACM SIGCOMM '99*, Cambridge, MA, September 1999.

[17] L. Rizzo. Effective erasure codes for reliable computing. In *Computer Communication Review*, April 1997.

[18] L. Vicisano, L. Rizzo, and J. Crowcroft. TCP-like Congestion Control for Layered Multicast Data Transfer. In *Proceedings of IEEE INFOCOM '98*, San Francisco, CA, April 1998.

[19] K. Yano and S. McCanne. The Breadcrumb Forwarding Service: A Synthesis of PGM and EXPRESS to Improve and Simplify Global IP Multicast. *ACM SIGCOMM Computer Communication Review (CCR)*, 30 (2), April 2000.