Fine-Grained Layered Multicast With STAIR

John W. Byers, Gu-In Kwon, Michael Luby, *Member, IEEE*, and Michael Mitzenmacher, *Member, IEEE*

***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 *noncumulative* layering and present a new, scalable multicast congestion control scheme called STAIR that embodies this approach. Our first main contribution is a set of performance criteria on which we base a comparative evaluation of layered multicast schemes. 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, aggregate bandwidth consumption and control traffic. The STAIR protocol that we subsequently define and evaluate is a multiple rate congestion control scheme that provides a fine-grained approximation to the behavior of TCP additive increase/multiplicative decrease (AIMD) on a per-receiver basis.**

***Index Terms*—Congestion control, Fibonacci sequence, noncumulative layering, reliable multicast, TCP-friendliness.**

I. Introduction

ONE OF THE significant challenges associated with multicast delivery to large audiences is providing a scalable congestion control mechanism that is not only compliant with TCP, but also addresses heterogeneity in end-to-end bandwidth across receivers. The now standard technique of *layered* multicast, which employs multiple multicast groups to transmit content at different rates, has been employed as a building block for these methods. A novel instantiation of this approach, receiver-driven layered multicast, was advocated by McCanne, Jacobson, and Vetterli [14] 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 [14], [15], [21]: the multicast group associated with the base layer transmits at a rate B_0 and all other layers $i \geq 1$ 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 with these layer rates is possible, it is necessarily coarse-grained. This is in contrast to TCP, which employs an additive increase/multiplicative decrease (AIMD) policy 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 [1], [5], [7], [8], [21].

A cumulative, layered organization has also recently been proposed for reliable multicast of large files [6], [21]. 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 join the session. Early work in this area addressed this challenge by combining use of Reed–Solomon forward error correction techniques together with careful organization of packet transmissions across layers [19], [21]. Subsequent work described a digital fountain model [6] which motivated the use of and employed new forward error correcting codes which can efficiently generate a virtually unbounded number of encoding packets [11]. In this model, to recover an original file packetized into P packets, it suffices for a receiver to receive *any* subset of encoding packets of size at least P. Unbounded encoding eliminates the need for complex packet scheduling algorithms across layers, since distinct encoding packets are abundant and equivalent. Moreover, such a fountain coding strategy can easily be combined with an arbitrary layered multicast organization, since subscription to an additional layer simply delivers encoded data more quickly. Furthermore, in contrast to the problem of layered video transmission, for encoded 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 noncumulative approaches for subscribing to sessions in a layered multicast.

Returning to the basic layering described above, it is clear that by using noncumulative 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 (not counting the base layer). This ability to fine-tune the

Manuscript received August 11, 2003; revised March 7, 2005; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor K. Almeroth. This work was supported in part by the National Science Foundation under NSF Grants ANI-9986397, ANI-0093296, and ANI-0205294, in part by NSF Grants CCR-9983832, CCR-0118701, CCR-0121154, and an Alfred P. Sloan Research Fellowship. Portions of this work appeared in preliminary form in papers in the Proc. IEEE INFOCOM 2001 and the Third International Workshop on Networked Group Communication 2001.

J. W. Byers is with the Department of Computer Science, Boston University, Boston, MA 02215 USA (e-mail: byers@cs.bu.edu).

G.-I. Kwon is with the School of Computer Science and Engineering, Inha University, Incheon 402-751, South Korea (e-mail: gikwon@inha.ac.kr).

M. Luby is with Digital Fountain, Inc., Fremont, CA 94538 USA (e-mail: luby@digitalfountain.com).

M. Mitzenmacher is with Harvard University, Cambridge, MA 02138 USA (e-mail: michaelm@eecs.harvard.edu).

Digital Object Identifier 10.1109/TNET.2005.863479

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 round-trip times.

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, noncumulative layering schemes have not been studied; the perception is that they are too complex and too costly. In this paper, we demonstrate that an AIMD multicast congestion control protocol can be realized and implemented with reasonable costs and complexity using noncumulative layering. We emphasize, however, that the question of appropriate tradeoffs is complex; hence we view the quantification and close inspection of the costs and benefits of noncumulative layering as a major contribution of our work.

In this paper, we propose STAIR (Simulate TCP's Additive Increase/multiplicative decrease with Rate-based control), a multicast congestion control algorithm built on noncumulative layering principles. The primary set of target applications are those requiring full reliability and sustained high throughput, such as large file transfers and video-on-demand. Our STAIR algorithm enables reception rates at each receiver to follow the familiar sawtooth pattern which arises when using TCP's AIMD congestion control. We facilitate this by providing two key contributions. First, we define a stair layer as a layer whose rate dynamically ramps up over time from a base rate of one packet per round-trip time (RTT) up to a maximum rate before dropping back to the base rate. The primary benefit of this component is to facilitate additive increase automatically, without the need for Internet Group Membership Protocol (IGMP) control messages. Second, we provide an efficient hybrid approach to combine the benefits of cumulative and noncumulative layering *below* the stair layer. This hybrid approach provides the flexibility of noncumulative layering, while mitigating several of the performance drawbacks associated with pure noncumulative layering. While our STAIR approach appears complex, the algorithm is straightforward to implement and easy to tune; it delivers data to each receiver at a rate that is in very close correspondence to the behavior of a unicast TCP connection over the same path; and it does so with a quantifiable and reasonable bandwidth cost. Finally, our congestion control scheme is primarily designed for users with high end-to-end bandwidth rates in the hundreds of kb/s range or higher. We expect that users with lower rates would wish to employ a different congestion control strategy than the one we advocate here.

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 noncumulative layering schemes and the performance metrics we propose to analyze them. In Section IV, we design novel noncumulative layering sequences designed from Fibonacci sequences which are optimized for AIMD multicast congestion control. We also provide a hybrid layering scheme which combines the benefits of cumulative and noncumulative layering schemes. In Sections V and VI, we define the STAIR protocol and present results of *ns* simulations that show the effectiveness of our approach.

II. Previous Work

We briefly survey the large body of relevant work in the area of multicast congestion control, and refer the reader to the excellent survey article of Widmer *et al.* [22] for further details. As noted in the introduction, cumulative layered multicast was first proposed in the context of the RLM protocol [14]. RLM employs 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. This limitation motivated Vicisano, Rizzo, and Crowcroft to propose the TCP-friendly Receiver-driven Layered Congestion Control (RLC) algorithm [21]. Their approach cleverly synchronizes join experiments by having the sender periodically and temporarily double the sending rate on each layer in turn. A receiver joins a higher layer only if there was no packet loss on its uppermost layer during such an experiment. We leverage synchronized rate changes such as these in our work as well, however, our approach uses much more *fine-grained* rate increases and thus does not run the danger of large-scale packet loss after a multiplicative rate increase like that in RLM and RLC.

Several other papers have also focused on TCP-friendly multicast congestion control. Models for characterizing TCP throughput as a function of the round-trip time and the steady state packet loss rate [7], [8], [17] led to proposals for unicast equation-based congestion control [17], and the TCP-Friendly Rate Control (TFRC) protocol [8]. Equation-based methods were subsequently applied to multicast congestion control in the TFMCC [23] and WEBRC [12] protocols.

In parallel with these innovations in multicast congestion control, work on integrating forward error correction (FEC) into layered multicast was emerging as an end-to-end solution for scaling *reliable multicast* to heterogeneous audiences [15], [19], [21]. This work demonstrated how Reed–Solomon codes could be used to provide protection against packet loss and described how to scheduled transmissions across a layered multicast session to reduce the likelihood of a host receiving redundant transmissions. Subsequent work advocated the use of much faster Tornado codes [13] and introduced the concept of fast FEC codes which are capable of generating a virtually unbounded amount of forward error correction [6]. LT codes [11] provide a realization of this concept. Such an unbounded encoding obviates the need for complex packet scheduling algorithms over layers; we exploit these techniques in our noncumulative design.

III. Notation and Definitions

We now consider the problem of allocating rates to the set of multicast sessions in a layered multicast group so as to enable integral receiver subscription rates in the *normalized* range $[1, R]$. In practice, these values correspond to multiples of the base layer bandwidth 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. Performance Metrics

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 cumulative layering scheme transmits on the base layer at normalized rate 1 and transmits across all other layers $i \geq 1$ at rate 2^{i-1}, i.e., the rates for the first few layers are 1, 1, 2, 4, 8, ... 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 that supports integral reception rates in $[1, R]$ is the number of multicast groups that the scheme uses 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. The basic cumulative layering scheme has density $\lceil \log_2 R \rceil + 1$.

Definition 2: For a layering scheme S which supports a subset of 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 ideal, 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. With the basic cumulative scheme, only rates which are powers of two can be realized, and thus the reception granularity is only marginally better than two, i.e., $2 - (1/R)$.

Before moving to noncumulative schemes, we mention a natural, but problematic, method for achieving fine-grained control with cumulative layering: allow each layer to send at the rate $B_i = 1$. While the reception granularity of such a scheme is clearly the optimal value 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 real-valued $c > 1$, we may set $B_0 = 1$, $B_1 = \lceil c - 1 \rceil$, and $B_i = \lceil c^i - c^{i-1} \rceil$. 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, and the density becomes $\lceil \log_c R \rceil + 1$. Hence, there is a natural tradeoff available. By choosing $c < 2$, we can decrease the reception granularity at the expense of increasing the density by a constant factor; similarly, by choosing $c > 2$, we can increase the reception granularity and decrease the density.

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 its 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/leave complexity* of additive increase under a layering scheme S is the worst case number of multicast join and 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 and leave messages a receiver must issue to decrease its rate by the relevant decrease factor.

Another significant problem of noncumulative schemes is the need for extra bandwidth to accommodate receivers, which the example in Fig. 1 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 [Fig. 1(a)], 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 noncumulative scenario [Fig. 1(b)], 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

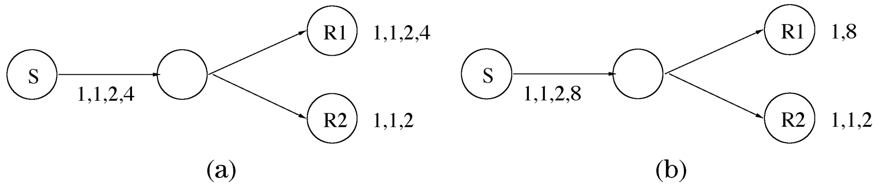


Fig. 1. Dilation of a shared bottleneck link. (a) Cumulative: Dilation $= 1$. (b) Noncumulative: Dilation $= 12/9$.

Sequence	Density	Reception Granularity	Dilation	Additive Increase	Multiplicative Decrease
Std. Cum	$\log_2 R$	2	1	N/A	1 leave
Std. NonCum	$\log_2 R$	1	2	$\mathrm{O}(\log R)$	$\mathrm{O}(\log R)$
Ideal	$\mathrm{O}(\log R)$	1	1	$\mathrm{O}(1)$	$\mathrm{O}(1)$
Fib1	$\log_{1.6} R$	1	1.62	2 joins, 1 leave	1 leave
Hybrid	$\log_{1.3} R$	1	1.27	2 joins, 1 leave	2 leaves

Fig. 2. Performance of various layering schemes defined in Sections III and IV.

on link l now jumps from eight to twelve. The requirement of additional bandwidth is a fundamental consequence of noncumulative layering and motivates 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 (C_ℓ/M_ℓ). Similarly, the dilation imposed by a multicast session is defined to be $\max_\ell(C_\ell/M_\ell)$.

In the example above, the dilation of l was 1 in the cumulative case and 12/9 in the noncumulative 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 noncumulative 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. This worst case dilation can be shown to be $2 - (4/R + 2)$.

We seek noncumulative 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 noncumulative scheme where layer sizes increase geometrically by a factor of two in Fig. 2. Our first set of main results is a pair of schemes, named Fib1 and Hybrid, that achieve a low join/leave complexity and a lower dilation than the basic noncumulative scheme with only a small increase in the number of layers.

IV. Layering Schemes

A. A Fibonacci-Based Scheme

We now provide an example of a noncumulative 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 $i \geq 2$.

By this definition, the first few rates of the layers for Fib1 are

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

As noted before, it will be useful to extend our Fib layering schemes 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 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 reception 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 nonnegative 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 $\tau = (1+\sqrt{5}/2) \approx 1.62$ (also known as 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 joining or leaving several layers. If we relax this requirement, so that we are only required to

[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.

approximately halve the reception 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 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 a binary notation, where the ith bit from the right (starting from 0) is set to 1 if the receiver is subscribed to the ith layer. For example, to denote that a receiver is subscribed to layers 0, 1, 3, and 5, we write 101011, with the base layer 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 a simple induction. ■

In the binary representation, decreasing the rate 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 lengthy induction which we omit, we find that this ratio is increasing in j. Hence, this ratio is upper bounded by the limiting value of the ratio, $1/\tau$.

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. In this case the ratio from leaving the jth layer is

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

Thus for large j, the ratio approaches $1/\tau^2$. ■

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.

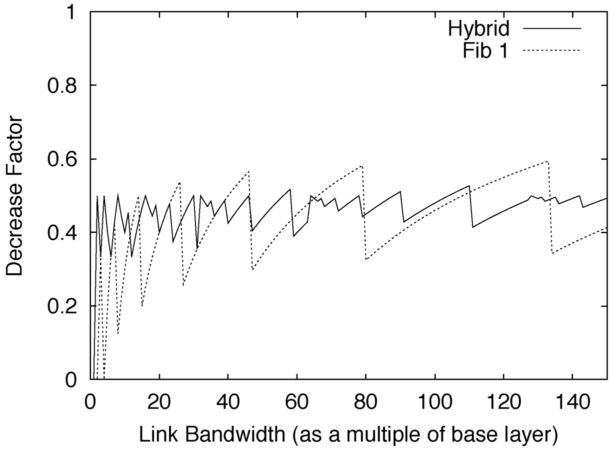


Fig. 3. MD factor as a function of available link bandwidth.

Fig. 3 shows the multiplicative decrease (MD) factor as the link bandwidth varies. The decrease factor is defined as a ratio of the rates before and after multiplicative decrease. For example, an MD factor of 0.5 corresponds to multiplicative decrease by half, and smaller MD factors correspond to larger decreases. In this plot, the range of MD factor periodically varies in the interval [0.34, 0.61] as j grows large, and we prove bounds of this form in Lemma 2. The MD factor is larger for small values of j, but STAIR is not recommended for low-bandwidth receivers such as this.

We also note that there is considerable work on changing the aggressiveness of additive increase given a decrease factor, to achieve TCP-friendliness. Several works [8], [20], [24] have studied different variants of AIMD TCP while providing TCP-friendliness. Such variants are called AIMD (a, b) congestion control [8] 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. The principles derived in those papers are applicable to our work as well.

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

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

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

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

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

B. Other Fibonacci 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 noncumulative 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 $i \geq 3$.

By this definition, the first few layer rates for Fib2 are

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

Now, 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*. Simple inductions show 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, and for $\sigma \approx 1.466$.[3] The density of the Fib2 scheme is therefore 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 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.

C. Hybrid Layering Scheme

Next, we describe a method for minimizing the performance penalty associated with noncumulative layering by employing a *hybrid* strategy which involves both cumulative and noncumulative layers. This hybrid approach retains all of the benefits of noncumulative layering scheme described in the previous section and in [4], with the added benefit that the dilation can be reduced from 1.62 down to $1 + \epsilon$ with only a small increase in the number of multicast groups. Note that both cumulative layers (CLs) and noncumulative layers (NCLs) are *static* layers for which the transmission rate to the layer is fixed for the duration of the session.

We denote the set of cumulative layers that we use by CL_i and the set of noncumulative layers that we use by NCL_i. The base

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

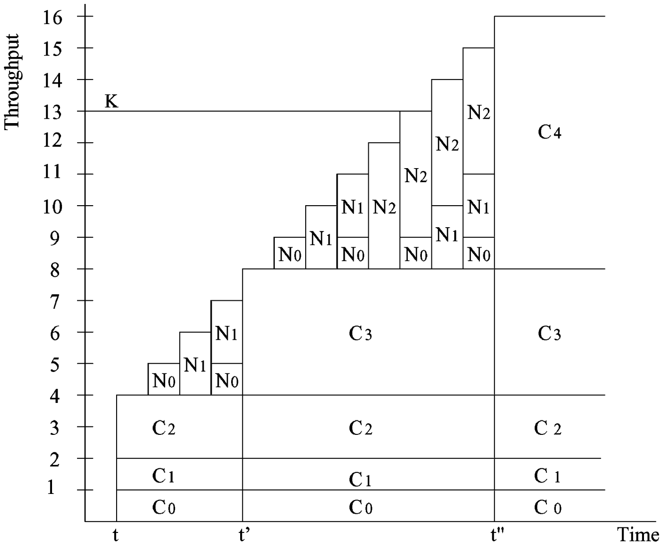


Fig. 4. Hybrid layer scheme: $K = \alpha^j + r$. For a target rate of 13, with $\alpha = 2$, $K = 2^3 + 5$. C_i denotes the rate on cumulative layer i, N_i denotes the rate on noncumulative layer i.

layer rate is $C_0 = 1$, and the ith cumulative layer CL_i has rate α^i for some real-valued parameter $\alpha > 1$. (When $\alpha = 2$, we have the standard doubling scheme.) The noncumulative layers use rates corresponding to the Fib1 scheme, also starting with $N_0 = 1$. To attain a given subscription rate K, a receiver will subscribe to set of cumulative layers to attain a rate that is the next lowest power of α, capped by a set of noncumulative rates to achieve a rate of exactly K, as depicted in Fig. 4. In particular, we let $j = \lfloor \log_\alpha K \rfloor$ and write $K = \lceil \alpha^j \rceil + r$, then subscribe to layers $\mathrm{CL}_0, \ldots, \mathrm{CL}_j$ as well as the set of noncumulative layers that the Fib1 scheme would employ to attain a rate of r. As prescribed in previous section, a fine-grained increase requires one join and two leaves, except for the relatively infrequent case when we move to a rate that is an exact power of α. In this case, we unsubscribe from all noncumulative layers and subscribe to one additional cumulative layer. Multiplicative decrease now requires one leave from a cumulative layer and one leave from a noncumulative layer. Leaving the highest CL reduces the reception rate on CLs by a factor of α, i.e., the rate on CLs is cut in half when $\alpha = 2$. Similarly, leaving the highest NCL decreases the reception rate on NCLs by approximately half.

Comparing against a standard noncumulative scheme, which used $\log_{1.6} R$ layers to obtain integral rates [1, R], we now require $\lceil \log_\alpha R \rceil$ layers for the cumulative part, plus roughly $\lceil \log_{1.6}(R - (R/\alpha)) \rceil$ noncumulative layers. This constitutes a constant factor increase in number of layers used. What we have gained is a dramatic improvement in MD factor and dilation. Fig. 3 compares the decrease factor of Fib1 and hybrid scheme with $\alpha = 2$. This figure shows experimentally that the MD factor for the hybrid scheme varies within a smaller range than Fib1 as j grows large, as well as providing better performance for small values of j. Analytically, we prove the following result:

Lemma 4: The dilation of the hybrid scheme is $1 + 1.62((\alpha - 1)/\alpha)$.

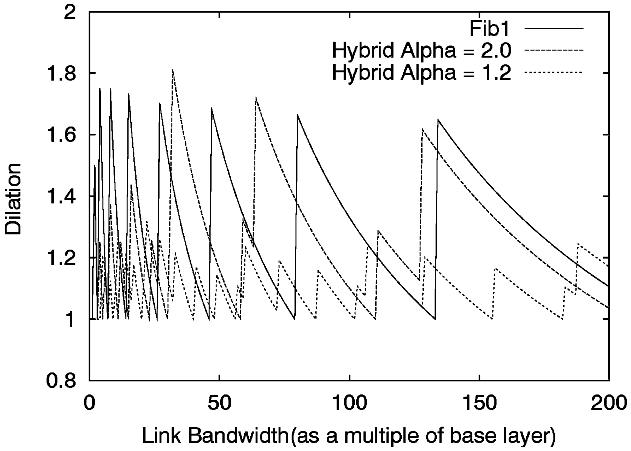


Fig. 5. Maximal dilation at a link as a function of available link bandwidth.

Proof: We proceed by proving an upper bound on the dilation D of an arbitrary link ℓ, which gives a corresponding bound on the dilation of the session. For each user U_i downstream of ℓ, denote the rate it obtains over the cumulative layers by a_i, the rate it obtains over noncumulative layers by b_i, and the total rate by u_i. Let the user with maximal total rate be denoted by $\hat{U}$, and let its corresponding rates be $\hat{a}$, $\hat{b}$, and $\hat{u}$ respectively. Now consider an arbitrary user U_i. By definition of $\hat{U}$, and from the organization of rates, $a_i \le \hat{a}$. If $a_i < \hat{a}$, then by the layering scheme employed, $u_i = a_i + b_i < \alpha a_i$. Adding αb_i to both sides gives $u_i + \alpha b_i < \alpha a_i + \alpha b_i = \alpha(u_i)$. Simplifying yields

$$b_i < \frac{u_i(\alpha - 1)}{\alpha} \le \frac{\hat{u}(\alpha - 1)}{\alpha}.$$

Otherwise, if $a_i = \hat{a}$, then by maximality $b_i \le \hat{b} < (\alpha - 1)\hat{a}$. In either case, $b_i < (\hat{u}(\alpha - 1)/\alpha)$, so $\max_i b_i < (\hat{u}(\alpha - 1)/\alpha)$. From Lemma 2, a set of users subscribing to noncumulative layers experiences a dilation of at most 1.62. Thus, the total bandwidth consumed by noncumulative layers across ℓ is at most $1.62 \max_i b_i$. Plugging these derived quantities into the formula in Definition 4 yields

$$D \le \frac{\hat{a} + 1.62 \max_i b_i}{\hat{a} + \hat{b}} < \frac{\hat{a} + \frac{1.62(\alpha-1)\hat{u}}{\alpha}}{\hat{a} + \hat{b}} < 1 + 1.62\frac{(\alpha - 1)}{\alpha}.$$

■

Applying this lemma to a hybrid scheme with a geometric increase rate of $\alpha = 1.2$ on the cumulative layers realizes the benefits of a noncumulative scheme, reduces the worst-case dilation in the limit from 1.62 to 1.27 (a 22% bandwidth savings) and requires only a modest increase in the number of groups. Fig. 5 shows the maximal dilation at a link as the link bandwidth varies as a function of N_0 for Fib1 and the hybrid scheme for two different values of α.

D. Introducing Stair Layers

While we can achieve a fine-grained approximation to additive increase by using the hybrid scheme directly, one salient problem is that the base layer bandwidth C_0 is fixed once for all receivers. Setting C_0 to a small value mandates frequent subscription changes (via IGMP control messages) for the receivers with small RTTs. Setting it to be large causes the problems of

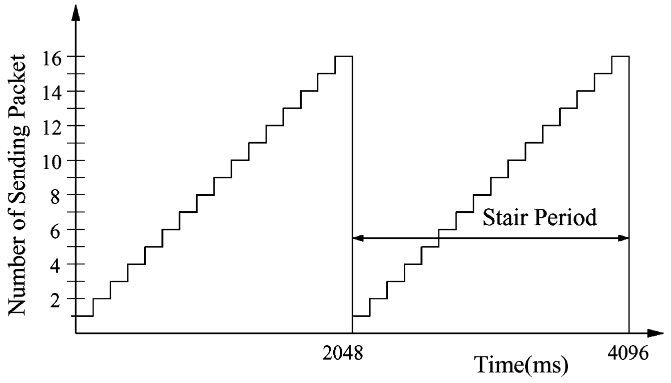


Fig. 6. Depiction of SL_{128} (a stair layer with $t = 128$ ms) in isolation. $R = 1$ Mb/s, packet size $= 1$ kB.

abrupt rate increases and buffer overruns that the hybrid scheme is designed to avoid.

We solve this problem using stair layers, so named because the rates on these layers change dynamically over time, and in so doing resemble a staircase. This third layer that a sender maintains is used to automatically emulate the additive-increase portion of AIMD congestion control, without the need for IGMP control traffic. Different stair layers are used to accommodate additive increase for receivers with heterogeneous RTTs from the source. These layers also smooth discontinuities between subscription levels of the underlying CLs and NCLs, which provide rather coarse granularity. In the subsequent discussion, we assume that these underlying layers have base rates $C_0 = N_0 = 1$ Mb/s for simplicity.

Stair Layers (SLs) are defined as follows. Every SL has two parameters: a round-trip time t that it is designed to emulate and a maximum rate R. The rate transmitted on each SL is a cyclic step function with a minimum bandwidth of 1 packet per RTT and a maximum bandwidth of R, a step size of one packet, and a stride rate of one step per emulated RTT. Upon reaching the maximum attainable rate, the SL restarts at a rate of one packet per RTT. Unlike CLs and NCLs, SLs are *dynamic* layers whose rates change over time. Dynamic layers were first used by [21] to probe for available bandwidth and later defined and used in [5] to avoid large IGMP leave latencies.

Fig. 6 shows the transmission pattern of SL_{128} (a stair layer for a 128 ms RTT) with maximum rate $R = 16$ packets per RTT. Also depicted in Fig. 7 is a third useful parameter of a stair layer:

Definition 8: The *stair period* of a given stair layer is the duration of time that it takes the layer to iterate through one full cycle of rates.

Given a stair layer with an emulated RTT t and a maximum rate R the stair period p satisfies $p \propto Rt^2$. Typically, we will set the maximum rate R of a stair layer to be the base rate of the standard cumulative scheme C_0 (in Mb/s), in which case we substitute for R and perform the appropriate conversions, assuming a fixed packet size S:

$$p = \left(C_0 \frac{t}{8S}\right) t.$$

We now consider the simple example depicted in Fig. 7, which depicts the throughput of a receiver when it subscribes to NCLs and SL_{128}. To simplify the description of this example,

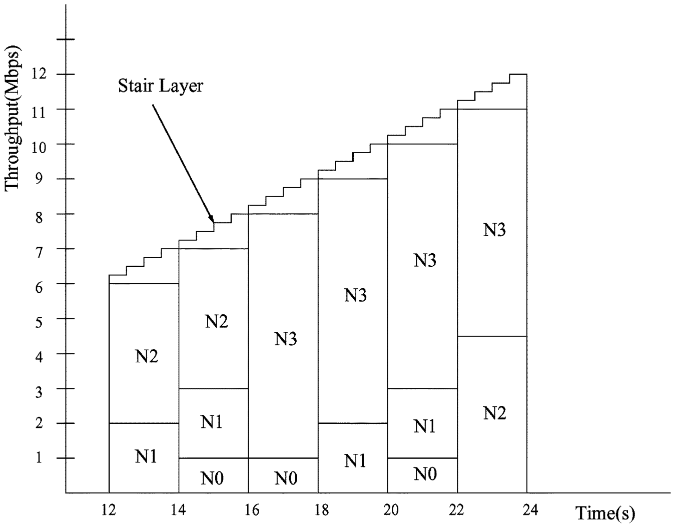


Fig. 7. SL_{128} used in conjunction with underlying noncumulative layers.

we employ stair layers on top of a pure noncumulative scheme; however, our algorithm and experiments use all three types of layers. For simplicity, let the stair period of SL_{128} be 2 seconds. Let $N_0 = 1$, $N_1 = 2$, $N_2 = 4$, $N_3 = 7$ (all rates in Mb/s). The receiver is subscribed to NCL_1 and NCL_2 at 12 seconds and has a total reception rate of 6 Mb/s. By subscribing to SL_{128} on top of NCL_1 and NCL_2, the receiver receives one more packet in every RTT. The sending rate of SL reaches B_0 at 14 seconds. SL then drops the sending rate to one packet per RTT at 14 seconds and resumes sending one more packet in every RTT. The receiver compensates for the drop by subscribing to NCL_0 at 14 seconds for a total reception rate of 7 Mb/s. At 16 seconds, the receiver unsubscribes from NCL_1 and NCL_2 and subscribes to NCL_3 to increase total reception to 8 Mb/s.

Finally, we note that the addition of stair layers increases the dilation beyond that proven in Lemma 4, but when stair layers are a very small fraction of the overall bandwidth (as is typical), their contribution in aggregate to the dilation is only a small additive term.

V. Stair Congestion Control Algorithm

We now describe how the techniques we have described come together into a unified multirate congestion control algorithm. We employ a hybrid scheme as described in Section IV-C, from which each receiver selects an appropriate subset of layers, used in concert with *one* stair layer, appropriate for its RTT. The two most significant challenges to address are providing the algorithms to performing additive increase and multiplicative decrease, respectively. Two additional challenges we address are: 1) incorporating methods for estimation of multicast RTTs and 2) establishing a set of appropriate stair layers.

A. Additive Increase, Multiplicative Decrease

In order for a set of stair layers to complement a set of CLs and NCLs, the maximum rate of the stair layer must be calibrated to the base rate of the CLs and NCLs. The effect of appropriately calibrated rates can be seen in Fig. 7: at exactly those instants when the stair layer recycles, the subscription rate on the NCLs increases by N_0, to compensate for the identical decrease on the stair layer. Now in order to conduct AIMD congestion control, the receiver measures packet loss over each stair period, during which additive increase takes place automatically. If there is no loss, then the receiver performs an increase of N_0. As described earlier, this entails one join and two leaves, or k leaves (where k is the number of subscribed NCLs) when the stair period is an exact power of α. (As an aside, we note that it may be much more efficient for a last-hop router to handle such a *batch* of IGMP leave requests, rather than handling them as k separate requests).

Conversely, if there is a packet loss event in a stair period (of one or more losses), then one round of multiplicative decrease is performed. Approximately decreasing the rate by half is straightforward—it is necessary to drop the top cumulative layer as well as the top noncumulative layer. We also note that there is no particular reason to wait until a stair period terminates before conducting multiplicative decrease—it can be done any time.

B. Configuration of Stair Layers

As motivated earlier, to accommodate a wide variety of receivers, stair layers must be configured carefully. We choose to space the RTTs across the available stair layers exponentially while noting that our methods generalize to other settings. Let the RTT in the base stair layer be 2^i ms. The base stair layer increases its sending rate every 2^i ms and all the other stair layers j will increase the sending rate in every 2^{j+i} ms. The TCP throughput rate R, in units of packets per second, can be approximated by the formula in [8] (derived by applying simplifying assumptions to a formula in [17]):

$$R = \frac{1}{\mathrm{RTT}\sqrt{q}\left(\sqrt{\frac{2}{3}} + 6\sqrt{\frac{3}{2}}q(1+32q^2)\right)} \tag{1}$$

where R is a function of the packet loss rate q and RTT is the TCP round-trip time. Since the throughput is inversely proportional to RTT, the receiver with a small RTT is more sensitive to the throughput than the receiver with large RTT, thus we recommend that RTTs provided by stair layers be exponentially spaced. Note that with an exponential spacing of stair layers, a receiver may subscribe to a different SL if its measured RTT changes significantly: it can subscribe to a faster layer at the end of its current stair period, or drop down to a slower stair layer every other stair period.

C. RTT Estimation and STAIR Subscription

In order to be TCP-friendly, each STAIR receiver must measure or estimate its RTT to subscribe to appropriate stair layers. Our goal is to minimize the discrepancy between the throughput received by TCP and by STAIR for a given RTT, using the following measure.

Definition 9: The *throughput discrepancy* of STAIR with a round-trip time R is the ratio of the throughput of TCP with round-trip time R to the throughput of a STAIR receiver with round-trip time R under identical loss rates. A variety of methods can be employed to measure the RTT; we describe three such possibilities, with the expectation that any scalable method can be employed in parallel with our approach. Golestani *et al.* [9] provide an effective mechanism to measure RTT in multicast using a hierarchical approach. However, their approach requires clock synchronization among the sender and receivers and depends on some router support which is not widely available. Another simple way to estimate RTT is to use one of various *ping*-like utilities. However, one cost associated with use of ping is that as the number of receivers increase the sender faces a "ping implosion" problem. Finally, WEBRC [12] defines a natural notion of multicast round-trip time (MRTT) that can be measured at a receiver without the involvement of the sender. MRTT is defined as the time between when a multicast join is sent for a group until the packet flow from that group is received. In the event there exists other receivers subscribing to that group in the multicast distribution tree, the MRTT reflects the round-trip time from a receiver to the branching point. In cumulative schemes such as WEBRC, the set of layers to which a slow receiver would subscribe is a subset of layers which are joined by a fast receiver, so the presence of a nearby fast receiver induces smaller MRTTs. In noncumulative schemes such as ours, this close correspondence between multicast groups is not present, so we leave the study of MRTT with noncumulative layering for future work.

With an estimate of its RTT, E, a STAIR receiver then subscribes to appropriate stair layers. In our evaluation, we use the following simple option involving a single subscription, but also describe a more complex option that reduces the worst-case throughput discrepancy.

Simple Option: Subscribe to the unique stair layer i satisfying $2/3\mathrm{RTT}_i < E \leq 4/3\mathrm{RTT}_i$.

This simple subscription policy emulates AIMD of a TCP experiencing a round-trip time of RTT_i. When compared with a TCP experiencing the "correct" round-trip time of E, it is clear that the throughput discrepancy lies in the interval [0.66, 1.33].

Complex Option: If there exists an i such that $(\mathrm{RTT}_i/\sqrt[4]{2}) < E \leq \sqrt[4]{2}\mathrm{RTT}_i$, then subscribe to stair layer i. If E is within a $\sqrt[4]{2}$ factor from the geometric mean of RTT_i and RTT_{i+1}, then subscribe to stair layers $i+1$ and $i+2$.

The intuition behind this complex option is that when the measured RTT lies midway between available options, the superposition of two stair layers provides a closer approximation to the appropriate additive increase rate than any one layer can. Using this approach, the throughput discrepancy lies in the interval [0.73,1.19]. Further details of the analysis, examples, and other options are provided in the STAIR technical report [3]. In the next section, we present results of *ns* simulations that demonstrate the effectiveness of our approach.

VI. Experimental Evaluation

We have tested the behavior of STAIR using the *ns* simulator [16]. Results of extensive simulations (reported on in more detail in [3] and [10]) show that STAIR exhibits good inter-path fairness in heterogeneous environments and when competing with TCP traffic in a wide variety of scenarios. Here, we report on a representative set of those experiments, paying particular attention to complex interactions over rich network topologies.

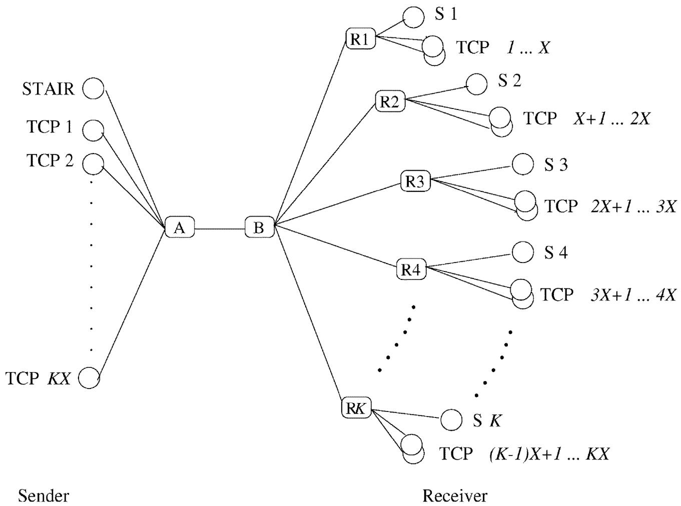


Fig. 8. Our network configuration.

The general network configuration we consider is depicted in Fig. 8. This topology generalizes both the standard "dumbbell" topology and the tree topology, and it allows for heterogeneous receivers (both bandwidth and delay) and multiple bottlenecks. On top of this topology, each of k STAIR receivers competes with X TCP flows that experience the same network conditions as that receiver. To test TCP-friendliness, our experiments compare the behavior of each STAIR receiver with the behavior of the X competing TCP flows.

By setting link bandwidths and delays appropriately (for example, see Fig. 12), we can establish situations with multiple target rates across the STAIR receivers. Typically, we configure the A-B link to have ample bandwidth; alternatively, we can establish it as an additional bottleneck link. With this configuration, we can tune the cross-traffic multiplexing level by varying the values of X and K, we can vary the link bandwidths and latencies, and we can scale the queue sizes.

Throughout our experiments, we set $C_0 = N_0 = 512$ kb/s and set $\alpha = 2$, i.e., the rate $C_i = 2^{i-1} * 512$ kb/s for $i > 0$, and employ a fixed packet size of 512B throughout. We use stair layers emulating exponentially spaced RTTs starting at 16 ms. In our experimental set up, each receiver periodically samples the RTT using *ping*. Also, while there is theoretical justification for smaller settings of α, we did not observe worst-case dilation often in our simulations.

In the experiments we describe here, we follow recommended guidelines for conducting multicast congestion control simulations [2]. In particular, we use RED gateways, primarily as a source of randomness to remove simulation artifacts such as phase effects that may not be present in the real world. Use of RED versus drop-tail gateways does not appear to materially affect performance of our protocol. The RED gateways are set up in the following way: we set the queue size to twice the bandwidth-delay product of the link, set *minthresh* to 5% of the queue size and *maxthresh* to 50% of the queue size with the *gentle* setting turned on. Our TCP connections use the standard TCP Reno implementation provided with *ns*.

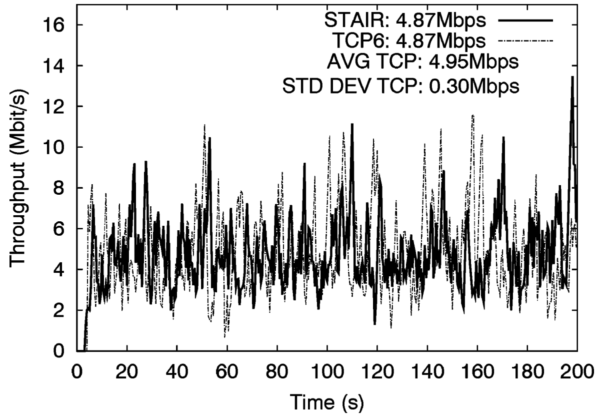


Fig. 9. TCP flows and one STAIR with RED.

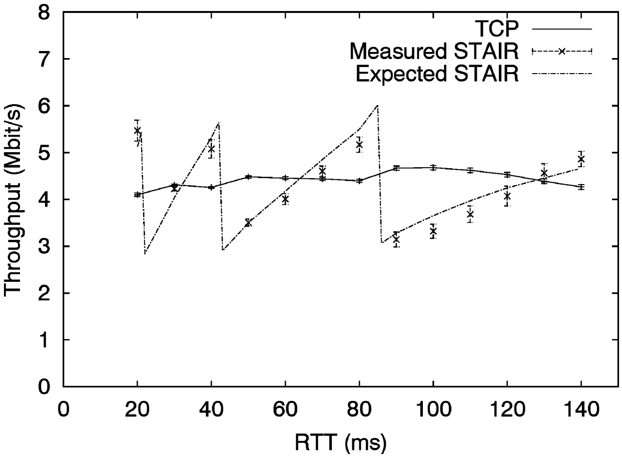
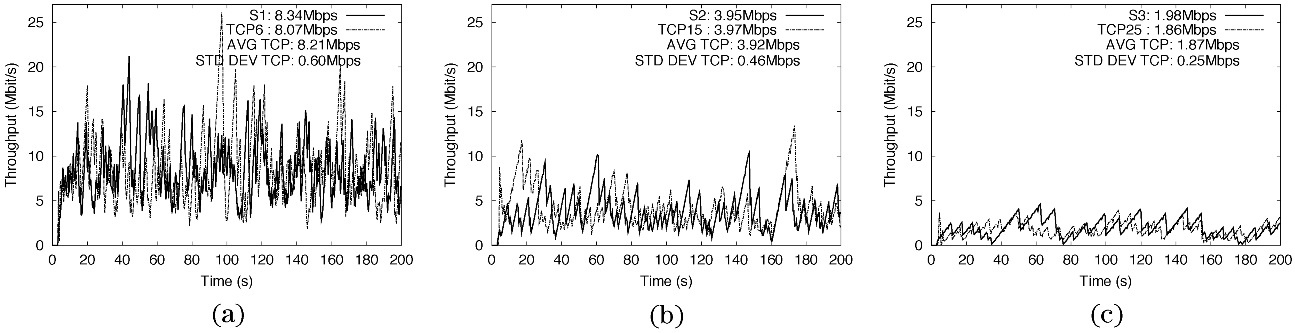


Fig. 10. Throughput comparison on different RTTs.

We begin with a simple scenario to test TCP-fairness in which one STAIR receiver flow is competing with seven TCP Reno flows (i.e., $K = 1$ and $X = 7$). The RTT is set to 32 ms and the bottleneck link is set to 50 Mb/s. We plot the throughput of STAIR flow and the throughput of a representative competing TCP flow in Fig. 9. Throughout this section, we choose representative TCP connections to avoid excess clutter in the plots. Our methodology is to choose the TCP connection whose mean rate was closest to the average of all TCP flows. Both the average rate across all TCP flows and the standard deviation are depicted in the plots. In this example, the average throughput attained by the STAIR receiver versus the TCP flows was 4.87 Mb/s versus 4.95 Mb/s, demonstrating fair sharing of the bottleneck link.

In the first experiment, the RTT was favorably set to a value (32 ms) that provided an exact match to a stair layer. Next, we vary the RTT between the sender and the receiver on the link to see the presence of throughput discrepancy induced by rounding to the nearest stair layer as described in Section V-C. We consider one STAIR receiver competing with ten TCP flows on the bottleneck link while varying the RTT by 10 ms from 20 ms up to 140 ms, and using the simple subscription option. The measured throughput of STAIR receiver and TCP receivers are depicted in Fig. 10 with a 98% confidence interval over 100 trials. We also plot the expected STAIR throughput, which is computed by multiplying the average TCP throughput by the ratio of the rounded RTT to the actual RTT on the link. As we expect, the throughput of STAIR is closest to that of TCP when the



(a) (b) (c)

Fig. 11. Throughput of STAIR and TCP flows sharing bottleneck link with different RTT. (a) S1 and TCP6 on 32 ms RTT; (b) S2 and TCP15 on 64 ms RTT; (c) S3 and TCP25 on 128 ms RTT.

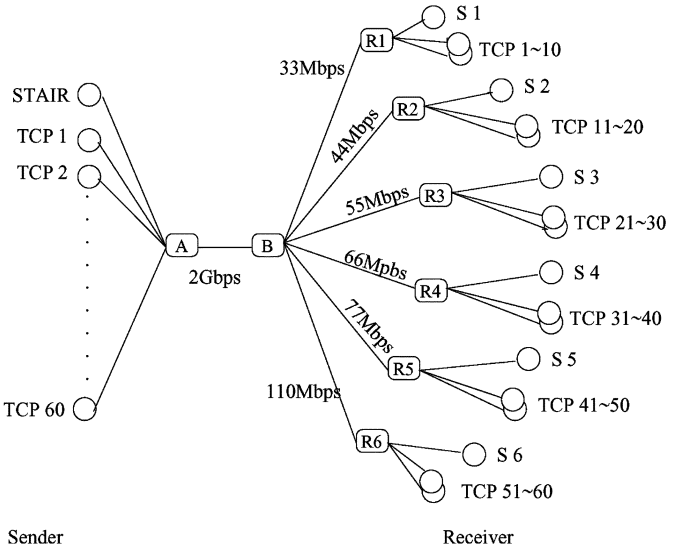


Fig. 12. Topology for many heterogeneous receivers.

RTT is close to the power of two, and the measured throughput discrepancy in our experiments lies very close to the predicted value.

We next consider topologies with considerable receiver heterogeneity, to verify STAIR's ability to perform fine-grained multiple rate control. We consider a single STAIR session with $K = 3$ and $X = 10$, but with different RTTs to the three receivers. The bandwidth on all links is set to 200 Mb/s, and the RTT of STAIR receiver 1, S1, is 32 ms, the RTT of S2 is 64 ms, and the RTT of S3 is 128 ms. This experimental set up makes the A-B link (see Fig. 8) the bottleneck for all STAIR and TCP receivers. Since the throughput of TCP is inversely proportional to RTT, the receiver S2 should have approximately half of S1's average throughput. The throughput of each of the flows is plotted in Fig. 11. All of the STAIR flows share fairly with the parallel TCP flows with the same RTT and the receiver with large RTT gets the small throughput as expected. In this experiment, the average throughput attained by S1, S2, and S3 was 8.34 Mb/s, 3.95 Mb/s, and 1.98 Mb/s, respectively, so each STAIR value was well within a standard deviation of the mean across competing TCP connections.

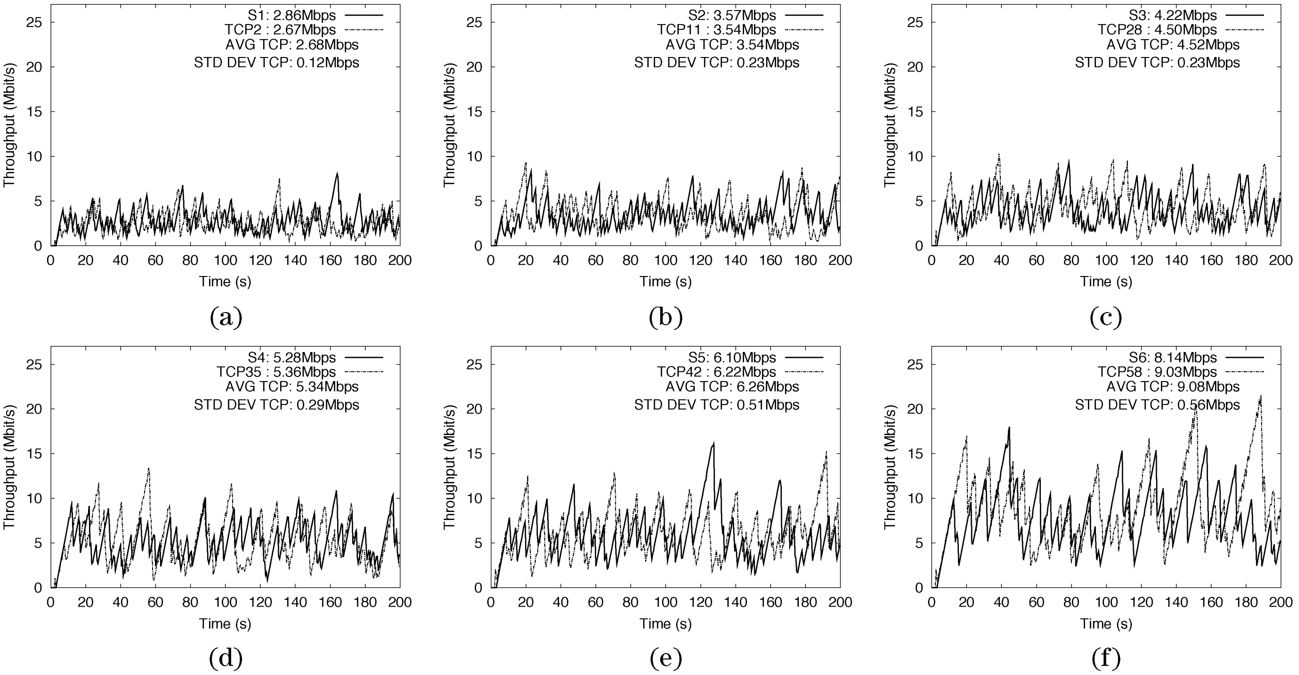
Finally, we used a topology with multiple bottlenecks (Fig. 12) to test the performance of STAIR with a set of heterogeneous reception rates. We consider a single STAIR session with $K = 6$ and $X = 10$, but each STAIR receiver is not behind the same bottleneck link. S1 competes with 10 TCP connections on a 33 Mb/s link, giving a fair rate of 3 Mb/s and the fair rates of the other STAIR receivers (S2 to S6) are 4 Mb/s, 5 Mb/s, 6 Mb/s, 7 Mb/s, and 10 Mb/s, respectively. We plot the throughput of each STAIR flow and the throughput of one of the competing TCP flows in Fig. 13. The level of fairness between the six STAIR receivers and the competing TCP connections is again high.

We measured the dilation as a function of time of link (A-B) in Fig. 12 and plot this dilation in Fig. 14. Recall that the dilation of a link was defined and bounds on the worst case dilation of STAIR were proven in Section IV-C. Fig. 14 shows that in this example, the measured dilation is much smaller than the worst case bound of 1.81.

VII. Conclusion

We have advocated a departure from standard cumulative layering for multiple rate multicast. Our approach, noncumulative layering, admits fine-grained multicast congestion control, which in turn enables each receiver to closely match its desired rate. Our work demonstrates that the costs of noncumulative layering need not be substantial, first by quantifying the parameters in the layered multicast design space, then by careful design and implementation of the STAIR congestion control protocol. Our approach has the appealing scalability advantage that it allows receivers to operate asynchronously with no need for coordination. Moreover, receivers with widely differing RTTs or amounts of available bandwidth may simulate different, TCP-friendly rates of additive increase.

Our hope is that noncumulative layered congestion control can be coupled with existing transport mechanisms to become a viable alternative to current coarse-grained multicast congestion control for a variety of multicast applications. We observe that congestion control approaches which use noncumulative layering cannot be considered general purpose (just as TCP's congestion control mechanism is not general purpose) since not all applications can take full advantage of highly layer-adaptive congestion control techniques. However, advances in fast FEC encoding for reliable multicast [6] and fine-grained rate-adaptive video coding [18] are enabling technologies in two application domains.



(a) (b) (c)

(d) (e) (f)

Fig. 13. Throughput of STAIR receivers and TCP flows. (a) S1 and TCP2; (b) S2 and TCP11; (c) S3 and TCP28; (d) S4 and TCP35; (e) S5 and TCP42; (f) S6 and TCP58.

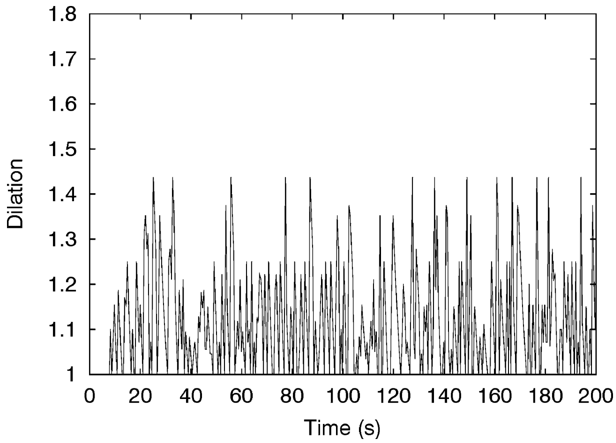


Fig. 14. Dilation on the shared link (A-B).

References

[1] D. Bansal and H. Balakrishnan, "Binomial congestion control algorithms," in *Proc. IEEE INFOCOM*, Anchorage, AK, Apr. 2001, pp. 631–640.

[2] J. Byers, M. Handley, G. Horn, M. Luby, and L. Vicisano. (2000, Aug.) More thoughts on reference simulations for reliable multicast congestion control schemes. [Online]. Available: http://www.cs.bu.edu/fac/byers/pubs/mrefsims.ps

[3] J. Byers and G. Kwon, "STAIR: practical AIMD multirate multicast congestion control," in *Proc. 3rd Int. Workshop on Networked Group Communication*, London, U.K., Oct. 2001, pp. 100–112. Full version appears as BU-CS-TR-2001-018, Boston University, 2001.

[4] J. Byers, M. Luby, and M. Mitzenmacher, "Fine-grained layered multicast," in *Proc. IEEE INFOCOM*, Apr. 2001, pp. 1143–1151.

[5] J. W. Byers, G. Horn, M. Luby, M. Mitzenmacher, and W. Shaver, "FLID-DL: congestion control for layered multicast," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 8, pp. 1558–1570, Oct. 2002.

[6] J. W. Byers, M. Luby, and M. Mitzenmacher, "A digital fountain approach to asynchronous reliable multicast," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 8, pp. 1528–1540, Oct. 2002.

[7] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Trans. Netw.*, vol. 7, no. 4, pp. 458–472, Aug. 1999.

[8] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *Proc. ACM SIGCOMM*, Stockholm, Sweden, Aug. 2000, pp. 43–56.

[9] S. Golestani, "Fundamental observations on multicast congestion control in the Internet," in *Proc. IEEE INFOCOM*, New York, Mar. 1999, pp. 990–1000.

[10] G.-I. Kwon, "Scalable architectures for multicast content distribution," Ph.D. dissertation, Boston University, Boston, MA, Jan. 2005.

[11] M. Luby, "LT codes," in *Proc. 43rd Symp. Foundations of Computer Science (FOCS 2002)*, Vancouver, BC, Canada, Nov. 2002, pp. 271–282.

[12] M. Luby, V. Goyal, S. Skaria, and G. Horn, "Wave and equation based rate control using multicast round trip time," in *Proc. ACM SIGCOMM*, Pittsburgh, PA, Aug. 2002, pp. 191–204.

[13] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Efficient erasure correcting codes," *IEEE Trans. Information Theory*, vol. 47, no. 2, pp. 569–584, Feb. 2001.

[14] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," in *Proc. ACM SIGCOMM*, Stanford, CA, Aug. 1996, pp. 117–130.

[15] J. Nonnenmacher, E. Biersack, and D. Towsley, "Parity-based loss recovery for reliable multicast transmission," *IEEE/ACM Trans. Netw.*, vol. 6, no. 4, pp. 349–361, Aug. 1998.

[16] The network simulator – ns-2. [Online]. Available: http://www.isi.edu/nsnam/ns

[17] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: a simple model and its empirical validation," in *Proc. ACM SIGCOMM*, Vancouver, BC, Canada, Sep. 1998, pp. 303–314.

[18] R. Rejaie, M. Handley, and D. Estrin, "Quality adaptation for congestion controlled video playback over the Internet," in *Proc. ACM SIGCOMM*, Cambridge, MA, Sep. 1999, pp. 189–200.

[19] L. Rizzo, "Effective erasure codes for reliable computing," *ACM Comput. Commun. Rev.*, vol. 27, no. 2, pp. 24–36, Apr. 1997.

[20] N. Sastry and S. Lam, "A theory of window-based unicast congestion control," in *Proc. IEEE ICNP*, Paris, France, 2002, pp. 144–154.

[21] L. Vicisano, L. Rizzo, and J. Crowcroft, "TCP-like congestion control for layered multicast data transfer," in *Proc. IEEE INFOCOM*, San Francisco, CA, Apr. 1998, pp. 996–1003.

[22] J. Widmer, R. Denda, and M. Mauve, "A survey on TCP-friendly congestion control," *IEEE Network*, vol. 15, no. 3, pp. 28–37, May 2001.

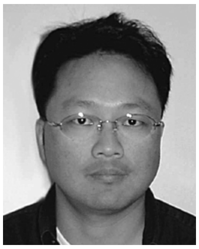
[23] J. Widmer and M. Handley, "Extending equation-based congestion control to multicast applications," in *Proc. ACM SIGCOMM*, San Diego, CA, 2001, pp. 275–285.

[24] Y. Yang and S. Lam, "General AIMD congestion control," in *Proc. IEEE ICNP*, Osaka, Japan, 2000, pp. 187–198.

John W. Byers received the Ph.D. degree in computer science from the University of California at Berkeley in 1997 and was a Postdoctoral Researcher at the International Computer Science Institute in Berkeley in 1998.

He is Associate Professor of computer science at Boston University, Boston, MA. His research interests include algorithmic aspects of networking, content delivery, and network measurement.

Dr. Byers received a National Science Foundation CAREER Award in 2001 and the IEEE ICDE Best Paper Award in 2004. He is active on numerous program committees, including ACM HotNets and ACM SIGCOMM. He is currently on the editorial board of IEEE/ACM TRANSACTIONS ON NETWORKING, and has been a member of the Association for Computing Machinery (ACM) since 1999.



Gu-In Kwon received the M.A. degree in computer science from City University of New York (Queens College) in 1998, and the Ph.D. in computer science from Boston University, Boston, MA, in January 2005.

He is Assistant Professor of computer science and engineering at Inha University, Incheon, South Korea. His research is on Internet content delivery, multicast congestion control, and overlay networking.

Michael Luby (M'02) received the Ph.D. degree in theoretical computer science from the University of California at Berkeley in 1983.

He cofounded Digital Fountain, Fremont, CA, in 1998, where he holds the position of Chief Technology Officer. He is a world-renowned scientist in the areas of coding theory, randomized algorithms, cryptography, and graph theory. He has been a Professor of computer science at the University of Toronto, Canada, and University of California at Berkeley. He is the inventor of the Luby Transform, the unique breakthrough technology that the Digital Fountain products are built upon.

Michael Mitzenmacher (M'01) received the Ph.D. degree in computer science from the University of California at Berkeley, where he shared the Sakrison Award for the best Ph.D. thesis in electrical engineering and computer science in 1997.

He worked at the Digital Systems Research Center until 1999, when he joined Harvard University, Cambridge, MA, where he is a John L. Loeb Associate Professor. He has authored over 90 conference and journal publications, on topics including load balancing, erasure codes, error-correcting codes, applications of codes to the Internet, Internet algorithms, bin-packing, and power laws. He recently co-wrote a textbook on probabilistic techniques in computer science with E. Upfal.

Dr. Mitzenmacher's work on low-density parity-check codes shared the 2002 IEEE Information Theory Society Best Paper Award.