Feedback-Free Multicast Prefix Protocols

Yair Bartal * John W. Byers † Michael Luby ‡ Danny Raz §

Abstract

Developing scalable, reliable multicast protocols for lossy networks presents an array of challenges. In this work we focus on scheduling policies which determine what *data the sender places into each sent packet. Our objective is to develop scalable policies which provably deliver a long intact prefix of the message to each receiver at each point in time during the transmission.*

To accurately represent conditions in existing networks, our theoretical model of the network allows bursty periods of packet loss which can vary widely and arbitrarily over time. Under this general model, we give a proof that there is an inherent performance gap between algorithms which use encoding *schemes such as forward error correction (FEC) and those which do not. We then present simple, feedback-free policies which employ FEC and have guaranteed worst-case performance. Our analytic results are complemented by trace-driven simulations which demonstrate the effectiveness of our approach in practice.*

Key Words: Reliable multicast, forward error correction, prefix protocol, scalability, encoding.

1 Introduction

The problem of reliably disseminating bulk data from a single source to a group of recipients is a fundamental and intensely researched problem underlying important networking applications. Among the challenges in developing solutions include ensuring scalability, tolerating packet loss and implementing rate-based flow and congestion control. Providing guaranteed performance of such protocols is a separate issue. As the network environment and protocols become more complex, it becomes very difficult to analyze the performance in the most general setting. In this paper we present truly scalable bulk multicast algorithms, which we analyze and evaluate in a general network model.

Our theoretical model of the network is motivated by many recent research results including [13, 18, 14] which indicate that packet loss patterns can be bursty and difficult to predict. Other factors which contribute to the sender's uncertainty about network behavior include high variability in end-to-end packet latency, and fluctuations in the transmission rate due to external factors such as network congestion. Packet loss is particulary problematic in that any packet which the source transmits might be discarded en route to any subset of the receivers. Moreover, the pattern of loss is largely outside of the source's control when best-effort protocols are employed, and network service is not guaranteed. These factors motivate studying the performance of transmission protocols on arbitrary transcripts of network behavior, rather than settling on a particular, but potentially contentious statistical model of the network.

In this paper, we focus specifically on the challenges involved in choosing *what* data the source should place in each sent packet. This policy, which we refer to as a data selection policy, can decide between transmitting a new portion of the source data, retransmitting a packet which has been lost, or transmitting a redundant codeword. We assume that the flow and congestion control policies which adjust transmission rates and determine *when* the source may inject data into the network are handled separately and independently. Thus, we do not consider these policies further in this paper. The benefits of this decomposition are twofold: it encourages the design of simpler, more modular protocols, and it allows us to study data selection policies in isolation, with-

*Bell Labs, Lucent Technologies, Murray Hill, NJ. Much of this research was performed while the author was a postdoctoral researcher at the International Computer Science Institute, Berkeley, CA. yair@lucent.com.

†International Computer Science Institute and U.C. Berkeley. Supported in part by NSF operating grants CCR-9304722 and NCR-9416101. byers@icsi.berkeley.edu.

‡International Computer Science Institute. Supported in part by NSF operating grants CCR-9304722 and NCR-9416101. luby@icsi.berkeley.edu.

§Bell Labs, Lucent Technologies, Holmdel, NJ. Much of this research was performed while the author was a postdoctoral researcher at ICSI. raz@lucent.com.

out the complicating influence of external factors. Further justification for this separation is provided in [1, 3].

An important consideration regarding these policies is the scalability of transmitting to large multicast groups. In particular, the problem of *feedback implosion* arises when a large set of receivers attempt to acknowledge or (negatively acknowledge) receipt of a packet. To mitigate this problem, many protocols are designed to minimize receiver-to-source feedback, often by using local recovery schemes. We concentrate on *feedback-free* protocols. Feedback-free protocols are important in related applications, such as reliable satellite communication, in which the back-channel can have high latency and low throughput. One would naturally expect that the absence of feedback makes it difficult and time-consuming to guarantee reliability. Indeed, approaches which use broadcast disks or data carouseling can be slow. But we show that this effect can be reduced by the use of forward error correction using erasure codes, which we describe momentarily.

To measure the performance of a multicast data selection policy, one natural metric is to minimize the time at which the protocol has delivered all of the data to all of the receivers. But often an application can do useful work with a portion, usually a prefix, of the transmitted data, so it can be important to measure the progress of the algorithm at intermediate points in the transmission as well. We therefore choose to measure the performance of our policies more stringently: at each point in time, we measure our protocol's performance based on the length of the longest prefix of the message obtained by each receiver. One justification for this measure is the strategy used by commercial Internet browsers to display downloaded pages with graphical content: the browser incrementally updates the image as quickly as possible as an intact prefix of the stream arrives. In the multicast setting, we further distinguish between the *minimum prefix measure* which measures the smallest prefix of all active receivers and the *average prefix measure* which measures the average prefix of all active receivers at a given point in time.

The analysis of these performance measures is in the spirit of competitive analysis of on-line algorithms introduced in [16]. Our analysis compares the worst case *competitive ratio* between the prefix delivered by our policy with that delivered by an optimal policy. The worst case is taken over all points in time during the transmission and over all transcripts of network behavior. In this model, we develop scalable multicast data selection policies which perform provably well for these prefix measures.

The work of [1] considered the performance of adaptive, retransmission-based prefix protocols for point-to-point connections. In this work, we consider performance of feedback-free prefix protocols for multicast connections.

The use of Forward Error Correction (FEC) has recently seen wider use in proposals to improve reliability in higher level protocols. In implementations of reliable multicast using FEC, the source transmits the original data along with some redundant data, comprised of codewords generated by an erasure code at the source. Widely used Reed-Solomon based erasure codes [12, 15] provide the following decoding guarantee: a receiver can use any k redundant codewords to reconstruct any k missing words of source data. In practice however, Reed-Solomon codes are often too slow to encode and decode when used to protect large blocks of data, so the sender must either divide the original data into smaller blocks which are protected separately [12, 15], or use much more time-efficient codes with slightly worse decoding guarantees [9, 4]. The use of FEC in reliable multicast can dramatically reduce the need for receiver-to-source feedback, as receivers no longer need to receive specific packets, but instead, can use redundant codewords to recover different lost message data.

We begin with a more detailed specification of our network models and our prefix performance measure in Section 2. In Section 3, we derive an information-theoretic lower bound on the performance of any prefix protocol in the general network model, and prove lower bounds on algorithms which do not use encoding. Our information-theoretic lower bound can be matched by an oblivious randomized algorithm which uses encoding. Moreover, if the network obeys a weak QoS guarantee specified in Section 2, we can achieve a constant competitive ratio using a simple FEC-based multicast approach. Then, in Section 4, we demonstrate that this last algorithm delivers high performance and scales far better than algorithms which use feedback, but not encoding on network traces and on simulations. We conclude with a discussion of the advantages and limitations of our study along with directions for future research.

2 Network Models

The model of the network we develop is *sender-centric*, driven by events which occur at the sender. As described in the introduction, the rate at which the sender may inject packets into the network is determined by an external *when* policy, typically a function of flow and congestion control considerations. At each of the sending times s_i specified by this when policy, the source may inject a packet into the network. What the sender chooses to place in the payload of these transmitted packets is the focus of this paper.

In many transport protocols, the sender receives feedback about the progress of its transmissions based on receiver-driven acknowledgments. In practice, for large multicast groups, this acknowledgment style causes unscalable *feedback implosion* at the source. A large body of recent research has studied alternatives to this approach; in this paper we consider data selection policies which are feedback-free and do not employ any receiver-to-source feedback. This is not to say that we restrict feedback from being used for other policies, such as congestion control.

In our network model, we associate the following quantities with packet i:

- s_i, the packet transmission time
- b_{ij}, a boolean set to 1 iff packet i arrives at receiver j
- a_{ij}, the time at which packet i arrives at receiver j (set only if $b_{ij} = 1$)
- w_i, the data, or payload, contained in packet i

Network behavior is modelled as a loss *transcript*, consisting of the values of the boolean variables b_{ij}. In the most general model we consider, the values b_{ij} may be set arbitrarily, allowing for bursty periods of loss which need not be correlated across receivers. We refer to transcripts produced by this model as *Arbitrary* transcripts.

But often, this worst-case model is overly pessimistic, and does not reflect the actual behavior of the network. For example, when TCP encounters congestion and packet loss, it uses multiplicative backoff to throttle back its transmission rate. This strategy ensures that packet loss periods are not too long-lived, and allows transmission to resume successfully, albeit at a slower rate. Similarly, proposed strategies for multicast congestion control such as [17] use receiver-driven dynamic rate adjustment in the event of high packet loss rates. These techniques motivate a less adversarial model of the network which enforces a limited quality of service (QoS) guarantee. The QoS guarantee provided by *Bounded* transcripts in this model of the network is the following: *In each period of W consecutive transmissions, at least a constant fraction cW of the transmissions arrive at each receiver.* Using the definitions provided earlier, this can be expressed as:

$$\forall i, j \sum_{k=i}^{i+W} b_{kj} \geq cW$$

In Section 4, we discuss how to select the parameters c and W to satisfy this bounded QoS guarantee in practice.

In Figure 2, we provide a pictorial representation of a transcript for a single receiver. As time elapses along the horizontal axis, we separately plot the transmission and arrival times of the packets. In the event that a packet does not arrive, we use coarse-grained timeouts to set an upper bound on the amount of time a receiver waits for a packet before it times it out. In this particular transcript the packets arrived at the receiver in-order, and we denote the vertical distance between the two plots at any time t by the backlog B_t, the number of packets in flight to the receiver at time t. Parameters such as the backlog and packet round-trip times play

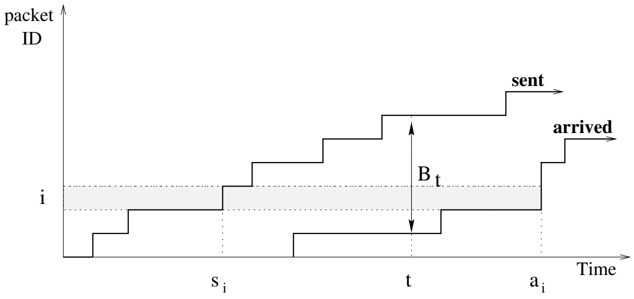

Figure 1. A transcript for a single receiver

an important role in policies which use receiver-to-source feedback, but are not as important in feedback-free policies. A more detailed discussion of latency effects in the model can be found in the full version of this paper.

3 Performance Analysis

3.1 General Network Model

In this section, we consider the case when *Arbitrary* transcripts are employed. Our objective in this model is to give bounds on the length of the prefix which a receiver can obtain given an arbitrary sequence of packet arrivals from a feedback-free source transmission algorithm. The encoding results developed in Priority Encoding Transmission (PET) [2] are ideally suited to this prefix measure and apply directly. In this work, a message of length n is assigned a set of priorities $k_1, \ldots, k_n$. The goal is to then generate an encoding of N packets with the property that, for all i, recovery of *any* k_i packets of this encoding suffices to recover the first i bits of the message. One of their results gives information-theoretic lower bounds on the feasible settings for $k_1, \ldots, k_n$. Then, using erasure codes, they constructively show how to generate an encoding for all settings above this information-theoretic threshold. With standard erasure codes, the scheme can be rather inefficient in terms of encoding and decoding speeds. But use of the ideas presented in [4] can substantially speed things up at the expense of sacrificing an additional small length overhead in the decoding operation.

Translated into the terminology of this paper, the main theorem in [2] can be stated as a result about feedback-free prefix protocols as follows:

Theorem 1 *Consider a prefix protocol which takes as input a file consisting of n bits and produces a sequence of N packets, each consisting of ℓ bits. Suppose the protocol wishes to guarantee that the first i bits of the file can be* *recovered from any set of k_i of the N packets. Then,*

$$\sum_{i=1}^{n} \frac{1}{k_i} \leq \ell$$

is a necessary and sufficient condition for realizing the protocol.

Note that if $k_i = \ln(n) \cdot i/\ell$ then this inequality is tight. Thus, by an averaging argument it follows that for at least one value of i it is the case that $k_i \geq \ln(n) \cdot i/\ell$, i.e., that the client needs to receive at least $\ln(n) \cdot i$ bits to recover the first i bits of the file. This indicates that there is at least one prefix for which the competitive ratio has to be at least logarithmic in n.

The practical difficulty with this scheme is that for this setting of the k_i, a receiver must receive $n \ln n$ bits to recover the entire message. In practice, it is possible to scale the specifications of the k_i to settings k'_i also obeying the PET condition in which $k'_i \leq c \ln n \cdot i/\ell$ and $k'_n \leq c' \cdot i/\ell$ for constants c and c'. This setting ensures that at the worst-case point in time, there is a logarithmic factor overhead, but upon completion of the transmission, the total overhead is constant.

3.2 The Necessity of FEC

In this section, we demonstrate that the performance of data selection policies which do not use an encoding scheme can achieve at best a logarithmic factor from optimal, even with full feedback and on *Bounded* transcripts. Next we present an algorithm which uses encoding to guarantee a constant competitive ratio on *Bounded* transcripts, and which scales with the number of receivers R.

Theorem 2 *No data selection policy which transmits only message words can achieve a better worst-case competitive ratio for either prefix measure than*

$$\Omega\left(\frac{\log(\min(R, W))}{\log\left(\frac{1}{c}\right)}\right)$$

Proof: Define a *loss pattern* to be a subset of ck packets which are lost in a sequence of k consecutive transmissions, where k is specified momentarily. To prove the lower bound, we will allow the adversary to map all distinct loss patterns onto one of the R receivers. Clearly, the number of distinct loss patterns of this form is $\binom{k}{ck}$, and we choose k as the minimum integer such that $\binom{k}{ck} \geq \min(R, W)$. We then specify a one-to-one, onto mapping of loss patterns to a subset of receivers of size $\binom{k}{ck}$. Transmissions to a given receiver in this subset are lost according to the corresponding loss pattern, repeated indefinitely. This setting ensures that the quality of service guarantee provided by the bounded adversary is satisfied. A diagram of loss patterns specified in this manner is given in Figure 2 for six receivers where $c = \frac{1}{2}$ and $k = 4$.

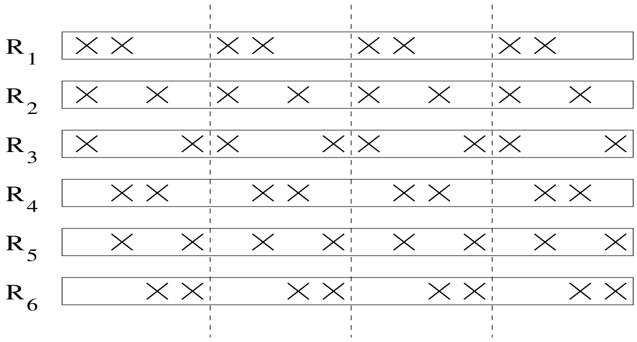

Figure 2. Loss patterns for 6 receivers with $c = 1/2$ and $k = 4$.

It is not difficult to observe that any data word that is sent less than ck times does not arrive at some receiver. This observation implies that in order for a data word to arrive at all receivers, it must be transmitted at least $ck + 1$ times. By standard bounds on $\binom{k}{ck}$:

$$\min(R, W) \simeq \binom{k}{ck} = \Omega\left(\frac{1}{c}\right)^{ck}.$$

Therefore, each data word must be transmitted at least

$$ck = \Omega\left(\frac{\log(\min(R, W))}{\log\left(\frac{1}{c}\right)}\right)$$

times, which immediately gives the bound on the ratio for the minimum prefix measure A similar argument gives the same bound for the average prefix measure and the theorem follows. ■

3.3 The Multicast Algorithm

Given the definitions and descriptions of efficient forward error correcting codes and the properties of the bounded adversary, the algorithm whose performance proves the following theorem is simple to state:

Theorem 3 *There is a multicast data selection policy which uses FEC with competitive ratio $O(1/c)$ with respect to both the minimum and average prefix measure on transcripts produced by a (c, W) bounded adversary.*

Proof: Our algorithm uses FEC codes and relies on the property of the bounded adversary that each user receives at least cW packets out of every consecutive set of W transmitted packets. For each *block* of cW consecutive message words, the algorithm constructs a set of redundant words of size $(1-c)W$ using FEC encoding. The algorithm then runs in a sequence of phases, each consisting of W transmissions, such that during each phase i, the algorithm transmits the cW message words and $(1-c)W$ codewords generated for the block of message words in the interval $(ciW, ..., c(i+1)W)$. By the bounded adversary, each receiver receives at least cW of the W transmitted words in each block. Then by the FEC coding guarantee described earlier in the introduction, every receiver has sufficient information to quickly reconstruct the original cW message words of this block in their entirety.

The algorithm clearly maintains the following invariant: *At the end of phase i, each receiver has a prefix of length exactly ciW.* By the bound on the number of transmissions in each phase, no algorithm could deliver a prefix of length larger than iW to any receiver by the end of round i. Since this argument also bounds the maximum and average prefix at the end of round i, an algorithm which achieves the above invariant performs within a factor of $1/c$ of optimal under both measures. The performance ratio for the algorithm follows. ∎

Of course, this algorithm is limited by the extent to which it is possible to accurately estimate the values of the parameters c and W, which may be difficult to realize in practice on existing networks. But in the event that packet loss rates and the duration of bursty loss periods can be bounded, the use of efficient FEC codes can provide dramatic performance improvements over retransmission based strategies which still predominate in existing multicast protocols. The empirical work described in the following section provides a preliminary indication of the magnitude of the improvement.

4 Empirical results

We tested the performance of our data selection policies by simulating their behavior under various conditions. One component of the testing was done on traces of MBone audio broadcasts, each lasting approximately one hour, broadcasting packets of size 2.5Kb at a constant rate of every $40ms$, taken from [18]. Each broadcast was received by between ten and sixteen receivers from the U.S. and Europe, and each receiver recorded which packets it received. Although timing information was not recorded in the traces, in our subsequent experiments, we estimated a typical round-trip time to the most distant receiver of about 1200 ms. In this paper, we did not consider the effects due to variance in round-trip times, nor was this information available in these traces. Additional tests were done on transcripts we generated at random as described in more detail below.

Using these packet loss traces as transcripts on which the performance of data selection policies can be tested, we compared the throughput of our multicast algorithm which employs FEC against two natural deterministic retransmission-based algorithms which we define momentarily. For the purpose of the comparison, we assumed that the retransmission based algorithms received full feedback from each of the receivers on the packets they received after an estimated round-trip time. (Of course, this would not be possible in practice because of feedback implosion, but we chose to give these algorithms this extra advantage). Both of the retransmission-based policies are greedy, in that they deterministically choose the word not currently in transit which, if delivered successfully, maximizes receiver benefit. We refer to those words which are not currently in transit as *unlocked.*

The **GreedyMin** algorithm attempts to send a word which will increase the length of the minimum prefix across all receivers: *At each transmission time, send the smallest unlocked message word that has not arrived at one or more of the receivers.*

The **GreedyAvg** algorithm attempts to increase the average prefix length across all receivers, preferably by sending a word which would extend the prefix of many receivers simultaneously. *At each time step, send the smallest unlocked message word which* if successfully received *would increase the lengths of the prefixes of the receivers by the maximum aggregate amount.*

Notice that both of these algorithms have the property that it could be the case that no word can increase the minimum (or average) prefix, and hence the algorithm must then send the smallest unlocked word. This observation indicates a potential flaw in these algorithms that is borne out in our simulations: it is possible that in a setting with substantial backlog, all words which can contribute to a receiver's prefix are depleted rapidly, after which the algorithm must transmit words which are not immediately useful. This in turn tends to cluster important words close together in the transmission sequence. But with the bursty loss patterns observed in these traces by Yajnik et al [18], temporal correlation of loss implies that entire clusters may be lost and not be retransmitted until a round trip time elapses. This provides some intuition as to why randomization in the unicast setting can be beneficial, as it can separate clusters of important words from close temporal proximity, making them less susceptible to burst loss. Similarly, the use of forward error correction in the multicast setting makes all words in the stream equally important, and diminishes the harmful consequences of a burst loss event.

In Figure 3 we present the result of our simulations using a representative MBone trace as our transcript. For our algorithm we used $W = 200$ packets and $c = .75$, implying that the network would guarantee delivery of some 150 packets out of every 200. On this transcript, two overseas receivers experienced a blackout period in which they received no packets for several minutes, so these receivers

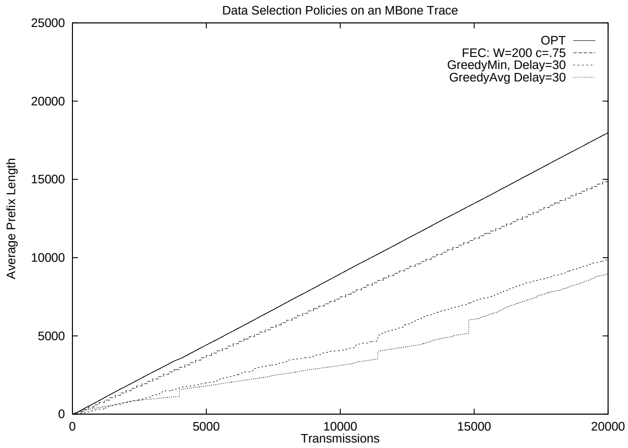

Figure 3. Multicast policies on an MBone Trace with 8 Receivers

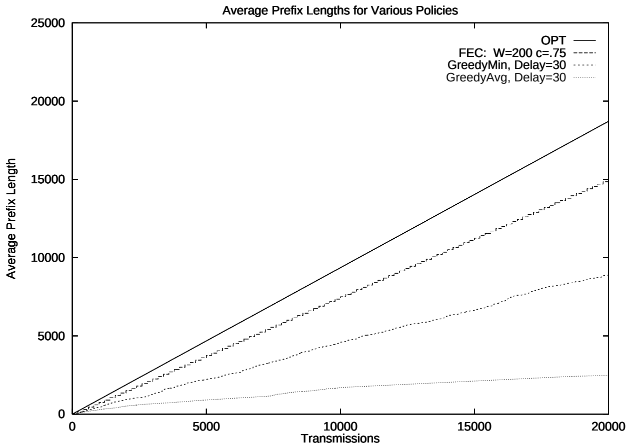

Figure 4. Simulated multicast with 24 receivers.

were not included in the simulation. In the plot in Figure 3, we plot the performance of our algorithms in delivering a bulk message to the 8 remaining receivers in the transcript. The available bandwidth is a plot of the (unachievable) value of the average prefix length at each point in time, under the assumption that *every* successful transmission to a receiver contributed to that receiver's prefix. This baseline gives a theoretical upper bound on the performance of any data selection policy. For the GreedyAvg policy, which does not attempt to provide any guarantee with respect to the smallest prefix over all receivers, we plot the length of the average prefix obtained over time. For both the FEC policy and for GreedyMin, the minimum prefix and the average prefix achieved by the algorithm are essentially the same at any point in time, as both algorithms focus exclusively on the receiver with minimal prefix. To provide a fair comparison, we also plot the performance of these algorithms with respect to the average prefix measure.

In order to test the scalability of the algorithm we also tested it on larger group sizes. In these experiments, we ran on simulated data where packets were lost independently at random for each receiver with probability 0.2. We present the performance result for a receiver set of size 24 in Figures 4 and for other receiver set sizes in the full version.

The main observation we make is that while there is a performance penalty associated with using the FEC algorithm due to transmission of redundant words, the magnitude of the penalty is fixed as the number of receivers increases. This can be contrasted with the performance of the greedy, retransmission based strategies, whose performance is acceptable for very small group sizes and when packet loss rates are small, but whose performance degrades rapidly as group sizes scale and packet loss rates increase. Similar observations were made in related studies of the benefits of FEC for reliable multicast of Nonnenmacher et al [12]. Surprisingly, while one might expect the GreedyAvg policy to deliver a larger average prefix than the GreedyMin policy, in practice, the GreedyAvg policy performs much more poorly. An explanation for this phenomenon is the shortsightedness of the GreedyAvg policy, in that it frequently transmits words which benefit a single receiver immediately, but which do not contribute to the prefix of other receivers until much later in time. The GreedyMin policy on the other hand focuses attention on receivers which have fallen behind, and once those receivers catch up, subsequent transmissions increase the prefix length of many receivers simultaneously.

Another important benefit of using the FEC scheme worth re-emphasizing which is not depicted in the plots is the lack of feedback required to employ the policy in practice. If the network quality of service guarantee is realized, receivers need not transmit any feedback to the sender, and in the event that the service guarantee is not realized, the receiver need only inform the sender the quantity of additional redundant packets it requires to reconstruct the block. Without an underlying service guarantee, one might consider combining the FEC technique with a sampling technique in order to tune the parameters W and c dynamically in periods of bursty loss. This would enhance the scalability of the algorithm as the number of receivers grows very large and providing service guarantees is no longer realistic.

5 Conclusions and Future Work

We have developed policies for selecting *what* data to place in packets of a multicast bulk transmission so as to maximize the lengths of the prefixes of the message obtained at each receiver at each point in time. Our main results indicates that the use of forward error correction is both necessary and sufficient for delivering scalable, high performance. The algorithm we present for networks which provide a loose QoS guarantee delivers high performance in trace-driven simulations and is backed by provable performance guarantees as well.

Our abstraction of focusing on the data selection policy places the burden of deciding *when* to transmit packets to a different scheduling policy, which is not the focus of this paper. Recent work in this area, especially studies of congestion control using layered multicast groups [11, 17] are ideal candidates for interoperability with our data selection policy. Verifying that these policies interoperate well in practice is one departure point for future study. Another avenue for future study would be to consider adaptive, dynamic tuning of the parameters c and W used in our algorithm. Ideally, one would hope to keep W relatively small to minimize delay effects, and to keep c large, but small enough to tolerate bursty periods of high packet loss.

Finally, we note that our prefix model can be adjusted so that it more generally applies to performance of real-time streaming applications, where receivers are not concerned with the completion time of the broadcast, so much as the quality of service during the broadcast. A possibility in this area would be to incorporate the fraction of recently transmitted packets which have actually arrived at the receiver into the performance measure, as stale transmissions are typically no longer relevant to current performance.

References

[1] M. Adler, Y. Bartal, J. Byers, M. Luby and D. Raz. A Modular Analysis of Network Transmission Protocols. *Proc. of the 5th Israeli Symp. on Theory of Computing and Systems,* June 1997.

[2] A. Albanese, J. Blömer, J. Edmonds, M. Luby, M. Sudan, "Priority Encoding Transmission", 35^{th} **FOCS**, 1994, *IEEE Transactions on Information Theory* (special issue devoted to coding theory), Vol. 42, No. 6, November 1996.

[3] J. Byers. "Maximizing Throughput of Reliable Bulk Network Transmissions." PhD Thesis, University of California, Berkeley, December 1997.

[4] J. Byers, M. Luby, M. Mitzenmacher, A. Rege, "A Digital Fountain Approach to Reliable Distribution of Bulk Data," *To appear in ACM SIGCOMM '98*, September 1998. Preliminary version available as *ICSI Technical Report TR-98-005*, January 1998.

[5] S. Floyd and V. Jacobson. The Synchronization of Periodic Routing Messages. *ACM Transactions on Networking, Vol. 2, No. 2*, pp. 122-136, April 1994.

[6] S. Floyd, V. Jacobson, C. Liu, S McCanne, and L. Zhang, A Reliable Multicast Framework for Lightweight Sessions and Application Level Framing. To appear in IEEE/ACM Transactions on Networking.

[7] C. Huitema. The Case for Packet Level FEC. *Proc. of PfHSN'96,* INRIA, France, October 1996.

[8] V. Jacobson. Congestion Avoidance and Control. In *Proc. ACM SIGCOMM '88*, pp. 314-329, 1988.

[9] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman and V. Stemann. Practical Loss-Resilient Codes. In *Proc. 29th ACM Symposium on Theory of Computing*, 1997.

[10] M. Mathis and J. Mahdavi. Forward Acknowledgment: Refining TCP Congestion Control. In *Proc. ACM SIGCOMM '96*, pp. 281-291, 1996.

[11] S. McCanne, V. Jacobson and M. Vetterli. Receiver-driven Layered Multicast. In *Proc. ACM SIGCOMM '96*, pp. 117-130, 1996.

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

[13] V. Paxson. End-to-End Routing Behavior in the Internet. In *Proc. ACM SIGCOMM '96*, pp. 25-38, 1996.

[14] V. Paxson. "Measurements and Analysis of End-to-End Internet Dynamics." PhD Thesis, University of California, Berkeley, 1997.

[15] L. Rizzo, "Effective Erasure Codes for Reliable Computer Communication Protocols." In *Computer Communication Review*, April 1997.

[16] D. Sleator and R. Tarjan. Amortized Efficiency of List Update and Paging Rules. In *Communications of the ACM*, 28(2):202-208, 1985.

[17] L. Vicisano, L. Rizzo, and J. Crowcroft. "TCP-like Congestion Control for Layered Multicast Data Transfer." In *Proceedings of IEEE INFOCOM '98.*

[18] M. Yajnik, J. Kurose, and D. Towsley. Packet Loss Correlation in the MBone Multicast Network, In *Proceedings of IEEE Global Internet '96,* November, 1996.