An Analysis of Packet Loss Correlation in FEC-Enhanced Multicast Trees

Marc Mosko and J.J. Garcia-Luna-Aceves
Computer Engineering Department
University of California at Santa Cruz
Santa Cruz, CA 95064
{mmosko, jj}@cse.ucsc.edu

Abstract

We study group loss probabilities of Forward Error Correction (FEC) codes in shared loss multicast communication. We present a new analysis model with explicit state equations using recursive formulae. Our method looks at an FEC group as a whole, rather than analyze the number of transmissions of a particular packet. Our work applies to $C(n, k)$ erasure codes, where any k out of n packets may decode the entire group. We find the cumulative distribution function that all leaf nodes in a shared loss tree successfully decode a $C(n, k)$ FEC group, the probability mass function (pmf) for the number of leaf nodes hat successfully decode a transmission group, the expected number of packets received on successful decode and the expected number of missing packets on decode failure for a particular leaf node of the multicast tree, the pmf that all leaf nodes hold the same packets in common, and the expected height of packet loss. Most of our findings generalize to arbitrary trees with non-uniform link loss. Our results also apply to non-FEC trees. We illustrate applications of our work with examples.

1. Introduction

Supporting reliable multipoint communication efficiently over the Internet is becoming increasingly important for such applications as software updates, web caching and replication, and distributed simulations. There have been many proposals on how to organize nodes along multicast trees for efficient multipoint communication and how to recover from packet losses and bit errors to achieve reliable multicasting over such trees. Forward Error Correction (FEC) has been used in the past to recover from both corruption of packets and packet losses. In this paper, we focus on FEC codes that recover from erasures [14], given that packet losses in multicast trees due to congestion constitute the primary source of errors in multipoint communication over the Internet.

There is a large body of work on the effect of FEC in reliable multicast. FEC has applications to bulk data distribution (Digital Fountain [6], MFTP [16], and RMDP [20]), large bandwidth-delay scenarios [15], and real-time audio and video multicast (RTMC [7]), among other applications. Gemmel *et al.* [9] developed a scheme for one-to-many telepresentations. Barbeau uses RMDP to multicast management information to mobile agents [1]. Lee describes RAID-like technology for Video-on-Demand server redundancy [12]. One study found significant gains in reliable audio/video multicast over high loss links using FEC [8]. Bartal *et al.* developed a feedback free multicast protocol based on FEC with bounded worst case behavior [2]. Bhargava *et al.* use FEC codes to optimize battery life in mobile nodes over wireless networks [4]. Non-topology based reliable multicast schemes show significant improvement using FEC techniques [13] [1].

Studies have looked at using FEC for retransmission of data and pro-active loss reduction. Several papers have estimated appropriate FEC codes for certain situations (e.g. [11, 13, 17]). Some papers use experience or heuristics. Other quantitative analysis of FEC in multicast generally focuses on independent loss models or simulation. Chou, for instance, analyzes a hybrid FEC scheme for audio and video broadcast with up to 20% packet loss rate and independent transmission channels [8].

Our goal is to provide basic analytical tools to aide in the comparison of organizations and retransmission policies with and without FEC in shared loss multicast trees. In a shared loss tree, we explicitly account for correlated loss because of shared links. To achieve these goals, our work provides a means to calculate the following observables: (1) the cdf that all leaf nodes decode a $C(n, k)$ code on the first transmission; (2) the pmf that r leaf nodes decode on the first transmission; (3) the pmf that a particular leaf node received m packets given that the root sent l packets; (4)

[1] This same study found that topology based multicast schemes – those that use local recovery – benefit much less from FEC and may incure larger latencies with it.

the pmf that a particular leaf node received m packets (as in (3) above) and failed to decode; (5) the pmf that all leaf nodes received the same m packets in a sequence of n; (6) the pmf of any loss at height h.

Item (1) has applications for best-effort networks where communications use FEC to reduce loss. Other network topologies where the bandwidth-delay product is large might also use FEC in this mode. Items (2), (3), and (4) concern the first transmission of an FEC group. One may use them, for instance, to size a tree or find an appropriate code structure for a given tree. Item (5) is useful in comparing retransmission policies. By measuring the correlation of packet reception, one may determine the average number of packets in a group that need retransmission. Note that this is different than (3) and (4), which only concern a particular node. Item (6) gives us insight as to where a tree is most likely to lose a packet. We could use this information to place FEC repeaters or group leaders for subcasting repairs. Item (6) uses a Markov model which is useful for deriving other equations of interest. These include the probability of growth, decay, or equilibrium per tree height and the expected number of lost packets per tree height.

For a $C(n, k)$ FEC code, we say that a group of k message packets are encoded to n transmission packets, where $n \geq k$. Usually, $n = k + h$, where h is the number of "redundancy" packets. As long as a receiver correctly receives any k of the n packets, it may decode the entire group. For example, a $C(26, 20)$ code adds 6 extra packets. The rate of an FEC code is n/k. A $C(1, 1)$ code is equivalent to not using FEC. We use the term Transmission Group (TG) to mean the set of n encoded packets. The Message Size or Data Size is the set of k unencoded packets. A code may be systematic, and use unencoded data packets plus parity packets. Other codes may require that all n packets be encoded. We generally assume that codes are not systematic, but this is just to give a lower bound and only applies to some of our comparisons of FEC codes to non-FEC transmissions.

In this study, the particular nature of the FEC code is not of interest. Our analysis applies as long as we may express the code in terms of n, k. In our model of a multicast tree, internal nodes *do not participate* in the FEC code. While one may model receivers at internal nodes, the routers that forward message packets do not use the FEC code. Loss is uncorrected on a per-link basis. We also assume that the FEC code has perfect erasure correction as long as the receiver has at least k packets from a group.

Section 2 presents previous models for shared loss multicast trees. Our model is based largely on Bhagwat [3]. We extend this work's recursive formula for shared loss non-FEC trees to FEC-enhanced trees. We focus on the cumulative distribution function at a leaf node. In Section 3 we extend our results to a probability mass function for the number of leaf nodes that successfully decode an FEC group over an arbitrary tree. Section 4 develops the probability mass function for the number of packets in an $C(n, k)$ code received by a leaf node for a tandem tree. We further derive the pmf for the number of packets received when a leaf node fails to decode. From these we may generate the expected number of packets a leaf node is missing after the first transmission of an FEC group. The section also presents a Markov model to compute the expected height of packet loss in a non-FEC multicast tree. In Section 5, we develop theorems on leaf node packet correlation. Section 6 concludes with remarks on the implications of our study and the use of FEC in multicast communications. It also suggests further work in this area.

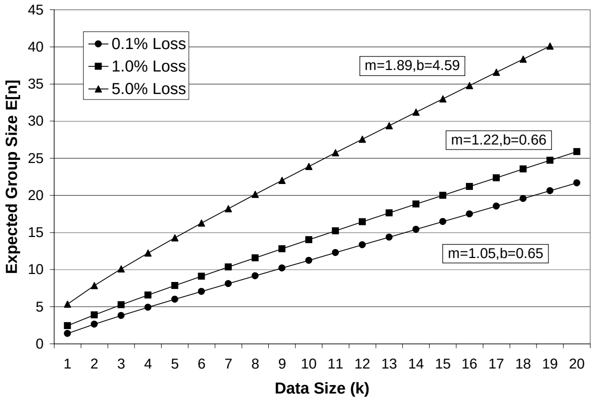

Figure 1. Expected TG size (256 receivers)

2. Single Packet Models

Several other papers have looked at FEC-enhanced multicast (e.g. [11], [21], [18]), but in general they consider independent loss. Nonnenmacher *et al.* [18] has a section on shared loss in FEC multicast, but it is a specific analysis of a restricted model. Our present analysis begins from the same roots, but yields general equations for shared loss trees.

Bhagwat [3] analyzes the distribution of the number of transmissions of a given packet over a shared loss tree until all nodes receive the packet . The analysis is done for non-FEC multicast. For a given node N, Bhagwat finds that if N's loss rate is p_N then the probability that i or fewer trials are needed to successfully deliver a packet to a receiver is given by a recursive geometric distribution. Letting $\mathsf{C}(N)$ be the set of N's children,

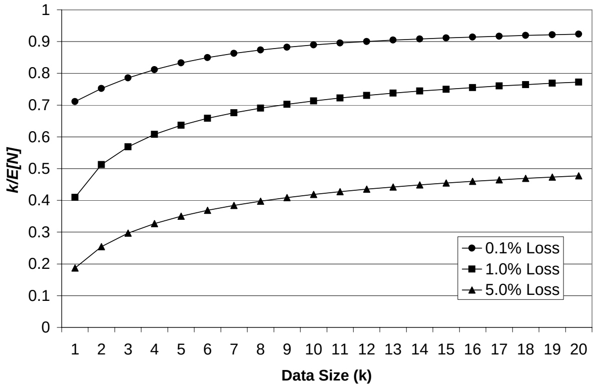

Figure 2. TG efficiency (256 receivers)

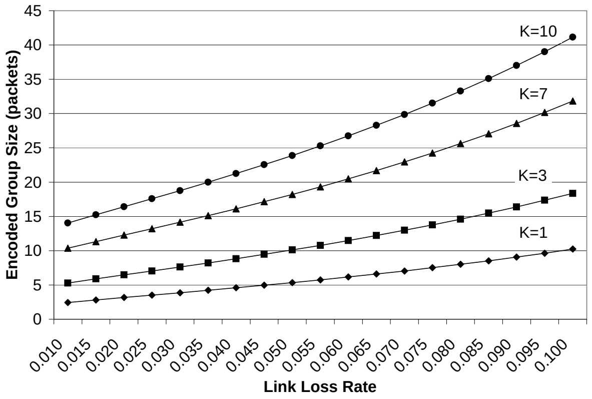

Figure 3. Expected TG size vs. loss rate (256 receivers)

$$F_N(i) = \begin{cases} 1 - p_N^i & N \text{ a leaf node} \\ \sum_{u=0}^{i-1}\left[\binom{i}{u} p_N^u (1-p_N)^{i-u} \right. & \\ \left. \prod_{K\in C(N)} F_K(i-u)\right] & N \text{ internal node} \end{cases} \tag{1}$$

Eq. 1 uses a geometric distribution to model leaf nodes and a binomial distribution to model internal nodes. We observe that since transit routers do not participate in FEC encoding, Bhagwat's model for internal nodes should still hold if we wish to find the probability that all leaf nodes decode.

For leaf nodes, we present a new distribution which accounts for FEC group recovery of packets. We use a negative binomial distribution for i trials until the k^{th} success. We may find the cumulative distribution function (cdf) that it takes i or fewer trials, $k \leq i \leq n$, to decode a $C(n,k)$ code. We note that Lemma 1 reduces to $1 - p^i$ when $k = 1$, in agreement with Eq. 1.

Lemma 1 *(proof omitted) For a leaf node L with link loss probability p in a shared loss multicast tree using an FEC code $C(n,k)$, where L's parent sends a total of i out of n packets, the probability that L decodes the group is $F_L(i) = \sum_{u=k}^{i} \binom{u-1}{k-1} p^{u-k}(1-p)^k$.*

In Eq. 1, we may use Lemma 1 in place of the geometric term to find the probability that all leaf nodes successfully decode a $C(n,k)$ FEC code. The subtlety of our transformation of Eq. 1 is that $F_N(i)$ is now the cdf that all nodes decode a $C(n,k)$ code where i is the number of packets out of a transmission group that a node receives. For a given data size, k, we may compute the probability that all nodes receive at least k packets with the source sending n. We may thus compute the expected group size for a given k such that all nodes receive at least k packets. In Fig. 1 we show the results of the calculation $E[n] = \sum_{i=k}^{z} i \cdot F(i)$. We terminated the summation at some z when the expected value increased by less than 0.01%. We show the slope (m) and y-intercept (b) of a linear regression against $E[n]$. All series fit a straight line very well. The coefficient of determination, R^2, is at least 0.998 in all three cases [10]. A value of $R^2 = 1$ indicates a perfect match. Fig. 2 shows the code efficiency $\nu = k/E[n]$. We use this definition as a measure of the overhead required to ensure, on average, that all nodes receive the data packet (for $C(1,1)$) or decode the transmission group. Since ν is relative to the message size k, we may compare efficiency between different codes. As one might expect, efficiency increases as the message size k increases.

Fig. 3 shows the expected transmission group size such that all receivers decode over link loss rates from 0.01 to 0.10. We see that the expected group size increases slightly faster than linear for a given k.

A more interesting question is how many nodes successfully decode a group on the first transmission, and for those nodes that fail, by how many packets they fail. Eq. 1 is not suited towards these questions. We must develop a new method of recursive tree analysis.

3. Recursive Tree Analysis

We wish to develop a general model for shared loss trees. We do not restrict the trees to fixed node degree or constant link loss. In our notation p_N is the uncorrected link loss probability of node N. It applies to packet reception. We set the root's loss probability to 0. The sender never loses a packet. We define the height of a node as the distance from the root.

In this section, we present a recursive formula to compute the probability mass function (pmf) that r leaf nodes in an arbitrary multicast tree successfully decode a $C(n,k)$ FEC code on the first transmission. We use different typefaces to denote a $\mathsf{S}et$ of nodes and a $\mathcal{F}amily$ of node sets.

Lemma 2 *(proof omitted) The probability that a leaf node has r successful decodes of a $C(n,k)$ FEC group given that it successfully receives m packets is as follows. $P_{leaf}[r \mid m] = 1$ if $r = 1, k \leq m$ or $r = 0, m < k$, and $P_{leaf}[r \mid m] = 0$ otherwise. This follows from the definition of a $C(n,k)$ code. The $r = 0$ case is the probability that the node fails.*

We define a function that generates families of permutation sets. The function $\mathcal{W}(N,r)$ returns the family of sets of the children of node N, where each set has weight r. This is equivalent to the problem of enumerating the permutations of placing r marbles in $|\mathsf{C}(N)|$ urns where each urn may hold at most m marbles. m is the number of leaf nodes at or under N. A value of $r = 0$ for a child indicates that no node at or under the child decoded. A positive value indicates the number of leaf nodes that decode. These disjoint sets span the state space of N's children with r allocated across them. We select the i^{th} child of N as $\mathsf{C}_i(N)$. Since N should be understood, we shorten the notation to C_i. When we sum over $\mathsf{f} \in \mathcal{W}(N,r)$, we iterate each set. f_i is the value assigned to C_i for a particular set. We also denote the link loss probability of C_i as p_i. We define the binomial pmf as $\mathrm{B}(i,j,q) = \binom{i}{j} q^{(i-j)}(1-q)^j$.

Theorem 3 *For an internal node N, the probability that r leaf nodes under N successfully decode a $C(n,k)$ FEC code on the first transmission, given that N receives m packets, is given by*

$$P_N[r \mid m] = \begin{cases} 0 & m < k, r > 0 \\ 1 & m < k, r = 0 \\ \displaystyle\sum_{\mathsf{f} \in \mathcal{W}(N,r)} \prod_{l=1}^{|\mathsf{f}|} \sum_{s=0}^{m} \Big\{ & \\ \mathrm{B}(m,s,p_l) \cdot P_l[\mathsf{f}_l | s] \Big\} & m \geq k \end{cases} \tag{2}$$

Proof: $\mathcal{W}(N,r)$ is a family of disjoint sets that span the state space. $\mathsf{f} \in \mathcal{W}(N,r)$, by definition, enumerates all configurations where r leaf nodes may successfully decode. If we can find the probability of each configuration and sum, we will have the desired result.

Each element f_l corresponds to a node. The value f_l is the exact number of successes that must occur at (if it is a leaf node) or under (if it is an internal node) C_l.

Case I: C_l is a leaf node. The probability that C_l has exactly f_l successes given that it's parent N attempts delivery of m packets is $P_l[\mathsf{f}_l \mid m] = \sum_{s=0}^{m} \mathrm{B}(m,s,p_l) \cdot P_{leaf}[\mathsf{f}_l \mid s]$.

We partition the solution over s, the number of packets C_l receives. The probability C_l receives exactly s packets given that N sent m packets is given by the binomial term $\mathrm{B}(m,s,p_l)$. By Lemma 2, we know that $P_{leaf}[\mathsf{f}_l \mid s]$ is the probability that C_l has f_l successes given that it successfully received s packets. By summing over all s from $0 \ldots m$, we have found the desired probability for a specific C_l. By taking the product over all l elements of f, we find the probability of the set. By summing over all sets, we find the desired result $P_N[r \mid m]$.

Case II: C_l is an internal node. This case, per se, does not have any successful decodes. Only leaf nodes decode. An internal node is a repeater between its parent and its children. It allocates successes to its children based on $\mathcal{W}(N,r)$. This case has three subcases. The first two subcases terminate the recursion while the third case is a recursive function.

Subcase II.A: $m < k, r > 0$. By the definition of $C(n,k)$, it is always false that any node decodes if $m < k$.

Subcase II.B: $m < k, r = 0$. By the definition of $C(n,k)$, it is always true that $r = 0$ nodes decode if $m < k$.

Subcase II.C: $m \geq k$. We note that this case is analogous to Case I, above. We enumerate all permutations of ways to allocate r successes over N's children. This is the same as the binomial term of $P_l[\mathsf{f}_l \mid m]$. The second term is now a recursive probability that each child has the allocated number of successes. This repeats until the child is a leaf node, at which time we may calculate the actual probability and solve the system by back substitution.

■

In the case where all children are identical, we may simplify this equation. We no longer need to generate all permutations, but only unique combinations for f. We may multiply by the multinomial of set f.

Nonnenmacher and Biersack [17] also compute the expected number of nodes in a multicast tree that receive a packet on the first transmission in a non-FEC multicast tree. Our equations here are a more general case. A $C(1,1)$ code in Eq. 2 yields identical values (albeit with more numerical work).

4. Computing Expectation Values

We address computing three expectation values. We begin with the expected number of nodes that decode a $C(n,k)$ code. Next, we consider the expected number of packets at any given leaf node. Finally, we develop a Markov model to compute the expected height of packet loss.

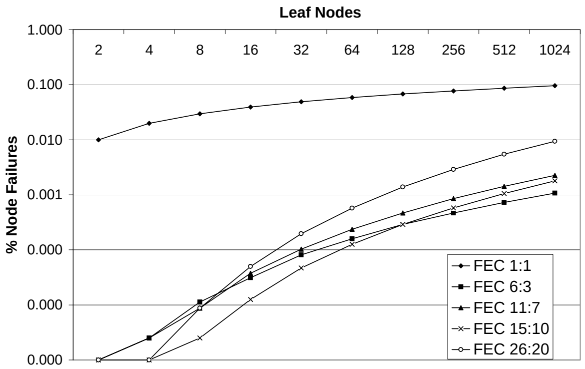

Figure 4. Percent expected leaf node failures, 1% loss rate

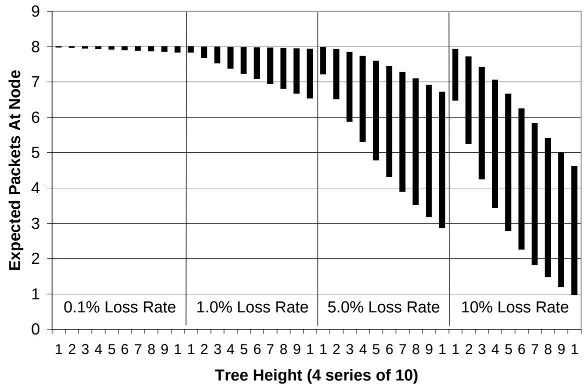

Figure 5. Mean number packets received at all nodes for $C(8,7)$ over four loss rates

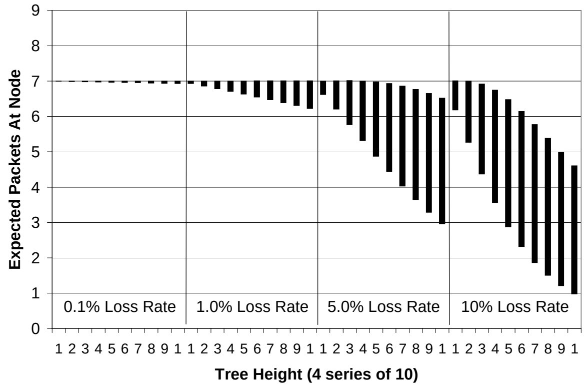

Figure 6. Mean number packets received at failed nodes for $C(8,7)$ over four loss rates

From Theorem 3, we may compute the expected number of nodes in a tree that successfully decode a $C(n,k)$ code. Let there be T total leaf nodes in the tree. We may compute the expectation value $E[R] = \sum_{r=0}^{T} r \cdot P_{root}[r,n]$.

Using the optimum $C(n,k)$ codes from Section 2, we computed Theorem 3, as shown in Fig. 4. Fig. 4 shows the percentage of receiver that fail, $E[R_{fail}]/T$, where $E[R_{fail}] = T - E[R]$. We used the FEC codes $C(1,1)$, $C(6,3)$, $C(11,7)$, $C(15,10)$, $C(26,20)$ at a link loss rate of 1%. In Fig. 4, the value “0.000%” means 0% loss to within double precision accuracy even though the scale is logarithmic. We chose the FEC codes based on a binary tree of height 8 (256 receivers). We show the percent node failures for a tree up to height 10 (1024 leaf nodes). At height 8, the expected number of node failures is $E[R_{fail}] = \{19.78, 0.12, 0.22, 0.15, 0.74\}$ for each FEC code in order of k value. At height 10, the values are $E[R_{fail}] = \{97.91, 1.10, 2.31, 1.83, 9.59\}$.

If we consider only the first transmission of a group, a poorly chosen FEC code may perform far worse than no FEC code at all. For a 1% per link loss rate in a binary tree of height 10 without FEC, we calculated that $E[R_{fail}] = 97.91$. For the code $C(8,7)$, $E[R_{fail}] = 179$. Over 1.8 times more nodes failed with FEC than without it. If $C(8,7)$ were a systematic code, the nodes would have usable packets but even still, more nodes would potentially ask for retransmission than without FEC (depending on the underlying multicast protocol). As we see from Fig. 6, when a node fails on $C(8,7)$ at height 10 with 1% loss, it most likely is only missing one packet. However, between the 179 failed nodes, there is very little correlation between lost packets. The source must retransmit most, if not all, of the group. In Section 5, we look at packet correlation in more detail. The present work, however, does not include a detailed analysis of retransmission behavior.

Computing the expected number of packets received at a leaf node is more challenging. We restrict ourselves to a tandem tree (a tree of degree 1, a path). We wish to find the probability that the leaf node received m packets given that the root sent l.

Our simplification to a tandem tree makes the equations tractable, but has one significant limitation. We may compute the expected number of packets received (or missing) at a given leaf node, but we do not know the correlation between leaf nodes. In Section 5, we address this issue.

Theorem 4 *For a tandem tree of total height H, the probability that the leaf node received m packets given that the sender sent l packets is*

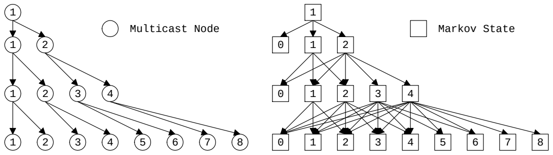

Figure 7. Markov chain for binary multicast tree

$$P_h[m|l] = \begin{cases} \sum_{i=m}^{l} B(l,i,p_h) \cdot P_{h+1}[m|i] & \text{h} < \text{H} \\ \\ B(l,m,p_h) & \text{h} = \text{H} \end{cases} \tag{3}$$

Proof: The $h < H$ recursive term sums a binomial term that an internal node sent i packets to it's child. These are all the possible ways that the child could receive at least m packets. The $h = H$ term is the probability that with the $H - 1$ node sending l packets the leaf node receives only m of them. ■

To compute the expected number of packets the leaf node is missing when it fails to decode a $C(n,k)$ code, we need to look at conditional probabilities. We wish to find the probability that the $H - 1$ node sent m packets and the leaf node failed to decode the transmission group. For $m < k$, this is just the probability that the parent sent m packets. When $m \geq k$, we must find the probability that leaf's parent sent m packets and that the leaf node received fewer than k packets.

Lemma 5 *(proof omitted) For a tandem tree, the probability that a leaf node fails to decode a $C(n,k)$ code after being sent m packets is $P[m \cap Fail] = P[m \cap Fail|m \geq k] + P[m|m < k]$.*

We may compute the first term from Theorem 3. By summing $P[1|m]$ for $m = k \ldots n$, we find the cdf $F[1|m]$ that the leaf node in our tandem tree decoded with $m \geq k$ packets. Using the relation that $P[m \cap Fail|m \geq k] = 1 - F[1|m]$, we have solved half of Lemma 5. We may compute the second term from the definition of conditional probability, $P[m|m < k] = P[m, m < k]/P[m < k]$ and Theorem 4.

We plot the results for a binary tree with a $C(8,7)$ code using loss rates of 0.1%, 1.0%, 5.0%, and 10%. Each loss rate covers a tree of heights 1 to 10. Fig. 5 shows the mean number of packets received in a group of 8. The vertical bars represent the variance with the mean in the middle. Fig. 6 shows the expected number of packets received when a node fails to decode.

We now turn our attention to a more general development for the probability of packet loss at height h in a multicast tree. We wish to know the most likely height of packet loss. We proceed by constructing a transient Markov chain. It applies to FEC-enhanced trees to the extent that such trees lose individual packets in the same manner as non-FEC trees. We shall assume a fixed node degree d and loss probability p. One may generalize to arbitrary configurations.

As shown in Fig. 7, we define a network $\mathcal{N}$ to be a transformation of the original multicast tree $\mathcal{T}$ as follows. Let there be $r(h)$ nodes in $\mathcal{T}$ at height h. For $h > 0$, there are $r(h) + 1$ nodes at height h in $\mathcal{N}$. We number nodes $v_{h,i}$ from $i = 0 \ldots r(h)$. The subscript i denotes the number of nodes at height h in $\mathcal{T}$ that receive a packet. At each height, node 0 is an absorbing state. It indicates that all packets sent from height $h - 1$ were lost at h.

Each directed arc in $\mathcal{N}$ has a weight equal to the probability of moving between end nodes $v_{(h-1),i}$ and $v_{h,j}$ where $0 \leq j \leq d \cdot i$. Since loses between adjacent tree levels are independent, the weight assigned to each arc is the binomial $p_{ij} \equiv P[X_h = j|X_{h-1} = i] = \binom{d \cdot i}{j} p^{d \cdot i - j}(1-p)^j$, which is the single-step transition probability. We always have the initial condition that $P[X_0 = 1] = 1$.

Theorem 6 *In a Markov network $\mathcal{N}$ as described above where h is a node's height, we may find the probability of any loss at height h, $P_{any}[h]$ and the expected height of packet loss, $E[H]$ as*

$$P_{any}[h] = \sum_{i=1}^{d^h} \sum_{j=0}^{d \cdot i - 1} p_{ij} \cdot P[X_h = i] \tag{4}$$

$$E[H] = \sum_{i=1}^{h} i \cdot P_{any}[i] / \sum_{i=0}^{h} P_{any}[i] \tag{5}$$

Proof: Eq. 4 finds the probability of any packets loss at height h. For each sending node i, we sum the probability that our final state $j < d \cdot i$. Eq. 5 is a standard expectation value. ■

Fig. 8 shows the expected height of packet loss for binary and ternary trees over 7 loss rates. The binary tree has a maximum height of 10 (1024 leaf nodes) and the ternary tree has a maximum height of 6 (729 leaf nodes). We see from the graphs that for $p < 0.3$, the expected loss height is closer to the bottom of the tree than to the top. This would argue for subcasting repairs to leaf nodes, similar to the findings of Linder *et al.* [13].

The Markov construction also allows us to compute other interesting characteristics of multicast trees. We may, for instance, compute the probability of decay $P_{dec}[h]$, equilibrium $P_{eq}[h]$ and growth $P_{gro}[h]$ at each tree level.

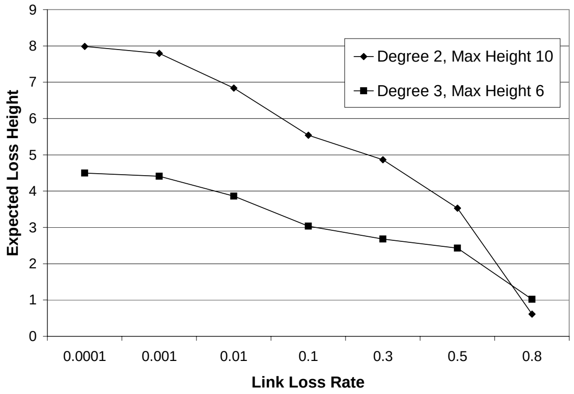

Figure 8. Expected loss height for binary and ternary trees

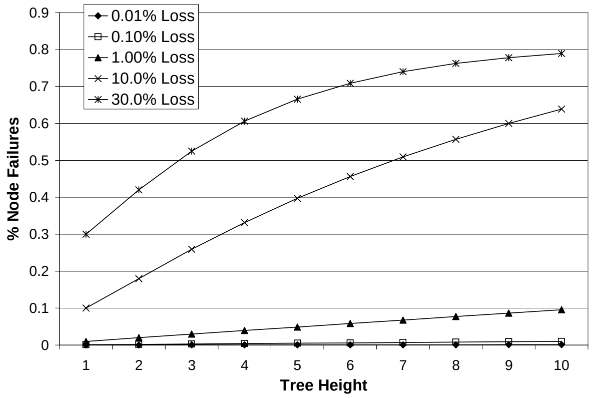

Figure 9. $E[N_h]/2^h$ **for binary tree**

$$
\begin{aligned}
P_{eq}[h] &= \sum_{i=1}^{d^h} p_{ii} \cdot P[X_h = i] && (6)\\
P_{gro}[h] &= \sum_{i=1}^{d^h} \sum_{j=i+1}^{d \cdot i} p_{ij} \cdot P[X_h = i] && (7)\\
P_{dec}[h] &= \varepsilon(h) + \sum_{i=1}^{d^h} \sum_{j=0}^{i-1} p_{ij} \cdot P[X_h = i] && (8)\\
\varepsilon(h) &= \varepsilon(h-1) + P[X_{h-1} = 0] && (9)\\
\varepsilon(0) &= 0
\end{aligned}
$$

$\varepsilon(h)$ is a normalization term such that $P_{decay}[h] + P_{eq}[h] + P_{grow}[h] = 1$ for all h.

Two other interesting expectation values to compute are the expected number of node failures per level $E[N_h]$ and

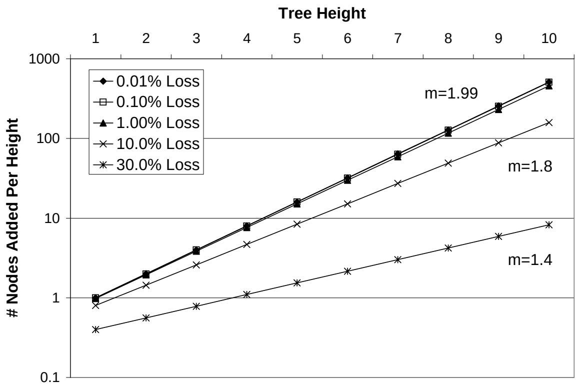

Figure 10. $E[G_h]$ **for binary tree (m is log slope)**

the expected growth rate of the tree per level $E[G_h]$. $E[N_h]$ is related to the number of NAKs generated in receiver initiated negative acknowledgment multicast. $E[G_h]$ gives a sense of how fast a tree will deteriorate. For non-FEC trees, it will always be less than the node degree for positive loss rates. We use the notation $f(x)^\dagger$ to mean the maximum of $f(x)$ or 1.

$$
\begin{aligned}
P[N_h = k] &= \sum_{i=\lceil d^h - \frac{k}{d} \rceil^\dagger}^{d^h} p_{i,(d^{h+1}-k)} \cdot P[X_h = i] && (10)\\
E[N_h] &= \sum_{k=1}^{d^{(h+1)}} k \cdot P[N_h = k] && (11)\\
E[G_h] &= \frac{\sum_{i=1}^{d^h} \sum_{j=0}^{d \cdot i} (j-i) \cdot p_{ij} \cdot P[X_h = i]}{E[G_{h-1}]}\\
E[G_0] &= 1 && (12)
\end{aligned}
$$

Fig. 9 shows Eq. 11 for a binary tree over 5 loss rates. It shows us what we would intuitively expect. We see that for the lower loss rates, nodes fail in roughly linear proportion to tree height. At higher loss rates, nodes fail faster and earlier in the tree. Fig. 10 plots Eq. 12. Since these are for a binary tree, the upper bound on the slope is 2. For the lower loss rates (0.01%, 0.1%), the plots are almost exactly two (they overlap in the graph). The 1% series is slightly worse. We would expect that for multicast trees that maintain a near exponential growth, the addition of extra error recovery might not be necessary. For plots such as the 10% or 30% series which are substantially under exponential growth, FEC may be a viable solution.

5. Leaf Node Packet Correlation

In Section 4, we found how to compute $P[m]$, the probability that any given leaf node received m packets. It should be clear that the analysis of that section does not give any information about the correlation between two nodes. We may find the expected number of packets at a node, $E[M]$. Naively, one may say that the expected number of negative acknowledgements (NAKs) would be $n - E[M]$. But, two nodes with the same $E[M]$ may not share any common packets in a sequence. In which case, the number of NAKs would be n. This has a significant effect on, among other things, NAK generation in reliable multicast. In this section, we develop an analysis of packet correlation, first for non-FEC trees and then for FEC-enhanced trees.

In general, we restrict our analysis to time invariant loss rates and regular full trees. These restrictions make the notation more tractable. When we say a packet is "held in common" at height h, we mean that all nodes at height h successfully received that packet. A "shared loss multicast tree" means a tree where we explicitly account for dependent packet loss. Each node v_i has a packet loss probability p_i that applies to packet reception. We define the *node family* of node v as the set of v and it's adjacent children. *Siblings* of node v are those nodes that share the same parent as v.

5.1. Example

We begin with an example to illustrate the objective of our current calculations. The example introduces the present material and also ties together material from previous sections.

Our example studies a multicast tree with 256 receivers. The tree has a special shape, such as to make the calculations from Section 5.3 computationally easier. In Fig. 11, the root has 64 children. There is zero loss between the root and each child. Each family under the root has four leaf nodes and uniform link loss. In our example, we use a 1% link loss rate. This structure allows us to treat each family independently, which is a great simplification to Eq. 13.

From Section 2, we calculated an optimum FEC code for the given tree. We chose our data size to be 3. Eq. 1 and Lemma 1 predict an expected group size of 4.42. We will use the FEC code $C(5, 3)$. For non-FEC transmission, the same equations predict a group size of 2.07. We would expect to need to send each data packet twice before every node received it. Using our definition of efficiency $\nu = k/E[n]$, the FEC code has $\nu = 0.6$ and the non-FEC code has $\nu = 0.5$. The FEC code can transmit groups of three packets to all receivers with less overhead than the non-FEC code.

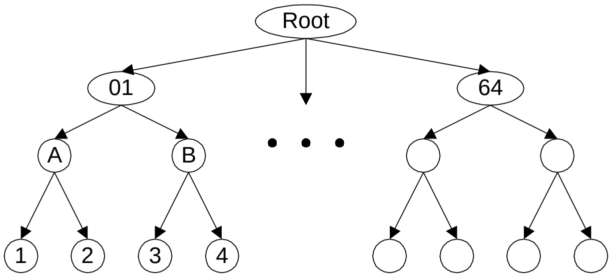

Figure 11. Example tree structure

From Section 3, we find that the expected number of leaf decodes is 255.98 for $C(5, 3)$. For non-FEC $C(1, 1)$, we find the expected number of leaf decodes is just under 234. About 22 nodes out of 256 will fail on each packet.

Imagine that we have a transmission window of 3 data packets (conveniently chosen to match our FEC code). Let us examine the results of Section 4. From Lemma 5, we compute for the $C(5, 3)$ code that any given node holds 2.98 packets when it fails to decode. It only needs 1 additional packet. This result, combined with the result that 255.98 nodes decode on average, would lead one to guess that only 1 packet might be missing on retransmission. Similarly for $C(1, 1)$, we find that in a sequence of 3 packets, any given node receives 2.94 packets per transmission. There are 22 nodes, on average, that fail. This probably means that all three packets must be retransmitted.

The equations of this section remove the "guess" and "probably" from the analysis. By examining the packet correlation at receivers, we may calculate precise expectation values and pmfs. The equations of this section tell us that in a sequence of 3 data packets without FEC, the expected number of packets held in common is 0.12. In other words, all three data packets must be resent on each transmission. This corresponds to our earlier calculation that we would need to send each packet 2.07 times until it is received. In the case of our $C(5, 3)$ code, we find that the expected number is 4.95 packets are held in common. On those occasions when a node fails, most of the time only a single packet from the group is missing.

5.2. Non-FEC Analysis

Lemma 7 *(proof omitted) In a shared loss multicast tree, let the root send a sequence of n packets over the tree. If at height h, m_h packets are held in common, then $m_{h+1} \leq m_h$ packets are held in common at height $h + 1$.*

Lemma 8 *In an independent loss multicast tree of height 1 (i.e. a single family), the root node v_0 sends sequence S of n packets to it's children $v_i \in \mathsf{C}(v_0), i = 1...c$ over links with loss probability p_i. The probability that all c chil-*

dren hold the same m packets in common is $P_{ind}[m] = \binom{n}{m}\rho^m(1-\rho)^{n-m}$ where $\rho = \prod_{i=1}^{c}(1-p_i)$

Proof: Represent the system state as a reception matrix where each row is a packet in S and each column is a child node. The array elements $a_{ij} = 1 - p_j$.

The probability that a particular packet i is received by all nodes is the row product $\rho_i = \prod_{j=1}^{c} a_{ij} = \prod_{j=1}^{c}(1 - p_j)$. The probability that any m out of n packets are received by all nodes is the binomial term $\binom{n}{m}\rho^m(1-\rho)^{n-m}$, where we drop the subscript i and assume time-invariant packet loss probability. ■

We now wish to extend Lemma 8 to multiple independent families. By Lemma 7, we know that the families of two siblings, call them v_1 and v_2, can only have at most those packets in common that v_1 and v_2 have in common. We wish to compute the probability that the children of v_1 and v_2 have the same m packets in common, knowing that they were sent the same sequence of common packets. We simply ignore any packets not held in common by v_1 and v_2. This assumes no loss correlation between packets, which may not hold when losses are related, for example, to buffer capacities.

Lemma 9 *Let there be f families rooted at height $h - 1, 0 < h$. Let family j have c_j children. We denote the link loss probability of child i of family j as $p_{j,i}$. The probability that all nodes at height h receive the same packet given that all parents transmitted the packet is $\rho_h = \prod_{j=1}^{f}\prod_{i=1}^{c_j}(1 - p_{j,i})$*

Proof: This follows as an easy extension of the proof of Lemma 8. We claim that when one only considers the families rooted at height $h - 1$ and packets held in common, the packet loss of each family is independent of other families. Let v_{h-1} and u_{h-1} be two distinct nodes at height $h - 1$. Let $v_{h,i}$ and $u_{h,i}$ be the sets of children of nodes v_{h-1} and u_{h-1}, respectively. By our premise, all nodes at height $h - 1$ have the same packet l. Since the families (sets) $\{v_{h-1}, v_{h,i}\}$ and $\{u_{h-1}, u_{h,i}\}$ are disconnected components, there can be no dependence between them for all packets held in common at $h - 1$.

Since each family rooted at height $h - 1$ is independent in this regard, imagine a set of f reception matrices. The probability that a packet in row i is received by all families is the product of all row products for row i. ■

In the case where all nodes have the same packet loss and all nodes have degree d, one could express $\rho_h = (1-p)^{d^h}$. One can see that as the breadth of the tree increases, the probability that all nodes hold a particular packet in common rapidly diminishes. In the case of reliable multicast, this plays an important role. To reduce shared loss, one may be tempted to reduce depth and increase width for the same number of receivers. As was desired, shared loss (correlation) is decreased, and as such it becomes less likely that receivers hold the same set of packets. An interesting problem would be to compute the optimum breadth versus height for a given receiver set size and link loss.

Theorem 10 *In a shared loss multicast tree where the root sends a sequence S of n ordered packets, the probability that the same m packets are held in common by all nodes at height $h = 0$ is 1 if $m = n$ or 0 if $m \neq n$ and for $h > 0$ is $P_h[m] = \sum_{i=m}^{n} B(i, m, 1 - \rho_h) \cdot P_{h-1}[i]$.*

Proof: From Lemmas 8 and 9, we claim that the binomial term is the probability that all nodes at height h receive the same m packets given that their parents sent the same i packets. It is analogous to the proof for Lemma 8. Since ρ_h is the probability that all nodes at height h hold a particular packet, the binomial term is the probability that all nodes at height h hold the same m out of i packets in common. We partition the space over the probability $P_{h-1}[i]$ that all parents hold the same i packets in common and sum. ■

5.3. FEC-enhanced Analysis

We now wish to include the effects of FEC erasure correction at leaf nodes. Because of FEC, there is a chance that a leaf node will have all n packets of a sequence, even if it's parent only had m packets, $k \leq m < n$. This violates Lemma 7. Thus, our above analysis for non-FEC trees does not hold. We may construct some approximations through column transforms of ρ, but for a general exact solution, we need to modify Theorem 3.

Below, we use the concept of a packet distribution matrix (pdm), similar to the reception matrices previously discussed. Each row is a packet i out of the sequence of n packets. Each column is a node j out of the d^h nodes (for a full regular tree). The pdm indicates the presence of a packet if element $a_{i,j} = 1$ or the omission of a packet if $a_{i,j} = 0$. The row product of a pdm is either 0 or 1. The column sum (number of packets at a node) is $0 \ldots n$. If the column sum for column j is at least k, then node j decodes.

For each call to our new recursive function, we specify an exact pdm for the leaf nodes. If we feed in, for example, all pdm's that have 2 packets in common, we may compute the probability that there are two packets in common. Since the pdm specifies the exact state at the leaf nodes, we may also compute the probabilities that r nodes decode, as in Theorem 3. We may also compute the joint distribution that r nodes decode with m packets in common. For that matter, we may compute almost any desired statistic concerning packet distribution at leaf nodes.

The unfortunate aspect of this method is that we must generate all permutations for a given condition and feed them through the root node. The pdm's must also obey rules about allowable packet distributions. If we wish to generate all pdm's for, say, 2 packets in common, we must generate the pdm's for actual packet reception at leaf nodes with the understanding that any column with $m \geq k$ packets will decode and actually have all n packets. If there are T total receivers, then the pdm is an $n \times T$ 0-1 matrix. There are 2^{nT} permutations. We are studying ways to reduce the state space.

We use the notation of $\mathbf{R}$ for a matrix and $\mathbf{r}$ as a vector (column unless otherwise noted). The typeface C still denotes a set. We take on an object-oriented notation with method selection. Matrices have the method $\mathbf{R}.\,\mathrm{cols} \rightarrow integer$, which returns the number of columns in $\mathbf{R}$ (the number of rows is fixed at n). A matrix also has $\mathbf{R}.\,\mathrm{req}(i) \rightarrow \{0, 1\}$, which indicates if packet i is required by the pdm $\mathbf{R}$. We may select a specific column from a matrix by $\mathbf{R}_i \rightarrow vector$, which returns column i as a vector. We define the matrix method $\mathbf{R}.\,\mathrm{split}(\mathsf{C}(N), l) \rightarrow matrix$. This function splits out the portion of matrix $\mathbf{R}$ that applies to the l^{th} child from the given set.

The $\mathbf{R}.\,\mathrm{req}(i)$ method may be computed as $\mathbf{R}.\,\mathrm{req}(i) = \bigvee_{j=1}^{\mathbf{R}.\,\mathrm{cols}} r_{ij}$, the inclusive OR of row i. If any child governed by $\mathbf{R}$ requires packet i, then $\mathbf{R}.\,\mathrm{req}(i) = 1$. If no child requires packet i, then the function returns 0. As we shall see below, the matrix $\mathbf{R}$ will be partitioned such that only those columns that apply to the leaf nodes under a node are communicated to that node. Thus, the required rows will vary as the partitioning progresses.

Analogous to Theorem 3, we will need a function to distribute packets over children in the tree. The new function $\mathcal{W}(N, \mathbf{R}, \mathbf{m})$ serves this purpose, similarly to $\mathcal{W}(N, r)$. $\mathbf{R}$ is the required leaf node pdm. $\mathbf{m}$ is the packet reception vector for node N. It indicates which packets N received from its parent and thus may send to its children. $\mathcal{W}(N, \mathbf{R}, \mathbf{m})$ returns a set of matrices $\mathbf{A}$, which are allowable packet transmission given $\mathbf{R}$ and $\mathbf{m}$. Each column of $\mathbf{A}$ is a child of N. $\mathcal{W}(N, \mathbf{R}, \mathbf{m}) = \big\{\mathbf{A} \mid \forall\ child\ j : a_{ij} \in \{\mathbf{R}.\,\mathrm{split}(\mathsf{C}, j).\,\mathrm{req}(i), m_i\}\big\}$. If packet i is required by any leaf node under child j then we must transmit it to and it must be received by child $j : a_{ij} = \{1\}$. If packet i is optional for all leaf nodes under child j, then a_{ij} may be $\{0\}$ or possibly $\{0, 1\}$ if N received packet i. From the distributions of higher nodes, we may not have received an optional packet. We note without proof that we always receive required packets by these rules. That is, if $\mathbf{R}.\,\mathrm{split}(\mathsf{C}, j).\,\mathrm{req}(i) = 1$, then $m_i = 1$.

Theorem 11 *Let $\mathbf{R}$ be a packet distribution matrix that specifies which packets are to be received or lost by each leaf node in a shared loss multicast tree with a $C(n, k)$ FEC code. At the root, $\mathbf{R}$ is an $n \times T$ matrix, where T is the total number of leaf nodes. Let $\mathbf{m}$ be a column vector that indicates which packets were received successfully by node N. At the root, $\mathbf{m}$ is the all 1's vector. If N is a leaf node, the probability that it receives $\mathbf{R}$ is given by $P_N[\mathbf{R} \mid \mathbf{m}] = 0$ if $\mathbf{R} \neq \mathbf{m}$ or $P_N[\mathbf{R}|\mathbf{m}] = 1$ if $\mathbf{R} = \mathbf{m}$. If N is not a leaf node, then*

$$P_N[\mathbf{R} \mid \mathbf{m}] = \sum_{\mathbf{F} \in \mathcal{W}} \prod_{l=1}^{\mathbf{F}.\,\mathrm{cols}} \Big({p_l}^{(|\mathbf{m}| - |\mathbf{F}_l|)} \cdot (1 - p_l)^{|\mathbf{F}_l|} \cdot P_{\mathsf{C}_l}\big[\mathbf{R}.\,\mathrm{split}(\mathsf{C}, l) \mid \mathbf{F}_l\big] \Big) \tag{13}$$

Proof: This form is essentially the same as Theorem 3. We sum over each component of $\mathcal{W}$, which enumerates the allowable spanning set of packet transmissions from node N to its children. The product selects each column vector from $\mathbf{F} \in \mathcal{W}(N, \mathbf{R}, \mathbf{m})$. Each child must exactly match the distribution specified in $\mathbf{F}$, so we multiply the probabilities that child l lost exactly $|\mathbf{m}| - |\mathbf{F}_l|$ packets and received exactly $|\mathbf{F}_l|$ packets. The operation $|\mathbf{v}|$ indicates the weight (norm) of the vector. Each element of the column vector $\mathbf{F}_l$ is either 1 to indicate the presence of a packet or 0 to indicate the omission of a packet.

Now that we have the probability that the child node C_l received the packet vector $\mathbf{F}_l$, we need to multiply by the probability that the leaf nodes under C_l receive the desired pdm specified in $\mathbf{R}$. Via the split method, we extract only those columns of $\mathbf{R}$ that apply to C_l and its children.

After appropriate recursion, we shall eventually reach a leaf node. At this point, we may compare the received packet vector $\mathbf{m}$ with the desired pdm vector (by the time $\mathbf{R}$ reaches a leaf node, the matrix is reduced to a vector). If they are equal, the leaf node returns a true value of 1. The recursion that lead to the leaf node was a valid recursion. If the vectors do not match, the recursion was invalid and the leaf node returns 0. Since this value is inside the product over l, a value of 0 will remove the invalid distribution from the overall probability calculation. In general, $\mathcal{W}$ may distribute optional packets to a leaf node, so this check is required. An actual implementation would have penultimate nodes distribute only $\mathbf{m}$. ∎

6. Conclusion

We found seven main results. In Section 2, we calculated the cdf $F_N[i]$ that all nodes decode an FEC code $C(i, k)$. We also computed the expected transmission group size $E[n]$ such that all nodes decode. In Section 3, we extended our results to the pmf $P_N[r|m]$ that r leaf nodes at or under node N decoded a $C(n, k)$ code given that N received m packets in the group. In Section 4, we developed three results. First, we found for a tandem tree $P_h[m|l]$, the pmf that the leaf node at height h holds m packets in given that the source sent l packets. Second, we developed a Markov tree to analyze the expected height of packet loss. Third, using the Markov tree, we found the pmf $P_{any}[h]$, the probability of any packet loss at height h and the corresponding expectation value $E[H]$. In Section 5 we found the pmfs for packet correlation between nodes for both non-FEC and FEC trees. In a non-FEC tree, the pmf $P_h[m]$ that m packets are held in common at height h uses a simple recursive formula. For an FEC tree, we gave the pmf $P_N[\mathbf{R} \mid \mathbf{m}]$ that the leaf nodes under node N received the packet distribution matrix (pdm) $\mathbf{R}$ given that N received the packet vector $\mathbf{m}$.

In Section 5, we also presented an example that ties together our results. It showed how to compute a good FEC code for a given multicast tree and data size k. The example then ran through the calculations of the other sections and showed how the results supported one another. It also compared the FEC results with non-FEC results to show the expected performance gain.

Future work could extend our analysis to include second order statistics as a measure of FEC code stability. The results in Section 5 for FEC trees, while tantalizing, are computationally infeasible because of the extremely large state space for the pdm (2^{nT}, T being the total number of leaf nodes with a $C(n, k)$ code). Further work may be able to reduce the state space by exploiting symmetries in the pdm based on row and column permutations. Our analysis only touched on retransmission performance and a more detailed analysis of specific FEC multicast schemes is in order. In particular, we would like to see an analysis of RTP with redundant encodings [5, 19]. This encoding structure has correlation between packets, since redundant information is piggy-backed with message packets. Also along these lines, we would like to see burst loss analysis with Gilbert channel models.

References

[1] M. Barbeau. Implementation of Two Approaches for the Reliable Multicast of Mobile Agents over Wireless Networks. *Proc. IEEE ISPAN'99*, pages 414–419, June 1999.

[2] Y. Bartal, J. Byers, M. Luby, and D. Raz. Feedback-Free Multicast Prefix Protocols. *Proc. IEEE ISCC'98*, pages 135–141, July 1998.

[3] P. Bhagwat, P. Mishra, and S. Tripathi. Effect of Topology on Performance of Reliable Multicast Communication. *Proc. INFOCOM '94*, 2:602–609, June 1994.

[4] A. Bhargava, L. Li, D. Agrawal, and P. Agrawal. D^2PAMN: Distributed Dynamic Power and Error Control Algorithm for Mobile Networks. *Proc. MASCOTS'98*, pages 295–300, July 1998.

[5] J.-C. Bolot, S. Fosse-Parisis, and D. Towsley. Adaptive FEC-Based Error Control for Internet Telephony. *Proc. IEEE INFOCOM '99*, pages 1453–1460, Mar. 1999.

[6] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege. A Digital Fountain Approach to Reliable Distribution of Bulk Data. *Proc. ACM SIGCOMM'98*, 28(4):56–67, Oct. 1998.

[7] G. Carle and J. Ottensmeyer. RTMC: An Error Control Protocol for IP-based Audio-Visual Multicast Applications. *Proc. IEEE ICCCN'97*, pages 566–573, Oct. 1997.

[8] P. Chou, A. Mohr, A. Wang, and S. Mehrotra. FEC and Pseudo-ARQ for Receiver-Driven Layered Multicast of Audio and Video. *Proc. IEEE DCC'2000*, Mar. 2000.

[9] J. Gemmell, E. Schooler, and R. Kermode. Feedback-Free Multicast Prefix Protocols. *Proc. IEEE Intern. Conf. on Multimedia Computing and Systems*, pages 128–139, June 1998.

[10] R. Jain. *The Art of Computer Systems Performance Analysis*. John Wiley & Sons, Inc., New York, 1991.

[11] R. G. Kermode. Scoped Hybrid Automatic Repeat request with Forward Error Correction (SHARQFEC). *ACM SIGCOMM '98*, Sept. 1998.

[12] J. Lee. Parallel Video Servers: A Tutorial. *IEEE Multimedia*, 5(2):20–28, Apr-Jun 1998.

[13] D. Li and D. Cheriton. Evaluating the Utility of FEC with Reliable Multicast. *Proc. IEEE ICNP'99*, pages 97–105, Oct. 1999.

[14] S. Lin and D. Costello. *Error Control Coding: Fundamentals and Applications*. Prentice-Hall, 1983.

[15] J. Linder, I. Miloucheva, and H. Clausen. A Forward Error Correction Based Multicast Transport Protocol for Multimedia Applications in Satellite Environments. *IEEE International Performance, Computing and Communications Conference*, pages 419–425, Feb. 1997.

[16] C. Miller. Reliable Multicast Protocols: A Practical View. *Proc. IEEE LCN'97*, pages 369–378, Nov. 1997.

[17] J. Nonnenmacher and E. Biersack. The Impact of Routing on Multicast Error Recovery. *Computer Communications*, 21:867–879, July 1998.

[18] J. Nonnenmacher, E. Biersack, and D. Towsley. Parity-Based Loss Recovery for Reliable Multicast Transmission. *IEEE Trans. On Networking*, pages 349–361, Aug. 1998.

[19] C. Perkins and *al.* RTP Payload for Redundant Audio Data. *RFC 2198*, Sept. 1997.

[20] L. Rizzo and L. Vicisano. RMDP: an FEC-based Reliabel Multicast Protocol for Wireless Environments. *ACM Mobile Computing and Communications Review*, 2(2):23–31, Apr. 1998.

[21] D. Rubenstein, S. Kasera, D. Towsley, and J. Kurose. Improving Reliable Multicast Using Active Parity Encoding Services (apes). Technical Report TR98-79, Computer Science Department, University of Massachusetts at Amherst, July 1998.