Hierarchical Reliable Multicast: performance analysis and placement of proxies

Athina P. Markopoulou, Fouad A. Tobagi
Stanford University
Gates Bldg. 3A-339, Stanford CA 94305-9030
(+1) 650 723 9330

{amarko, tobagi}@stanford.edu

ABSTRACT
The use of proxies for local error recovery and congestion control is a scalable technique used to overcome a number of well-known problems in Reliable Multicast (RM). The idea is that the multicast delivery tree is partitioned into subgroups that form a hierarchy rooted at the source, hence the term Hierarchical Reliable Multicast (HRM). For each subgroup, there is a designated node, the proxy, which is responsible for collecting the feedback from the subgroup and for locally re-transmitting the lost packets. The performance of any RM protocol is affected by the underlying multicast routing tree and its loss characteristics. Furthermore, the performance of the HRM approach, in particular, strongly depends on the appropriate partitioning of the tree and the selection of proxies. In this paper, we first model the HRM problem, then define and compute appropriate performance metrics and finally give insights on the optimal location of proxies.

Keywords
Performance analysis, reliable multicast, proxies.

1. INTRODUCTION
Consider a single-source multicast tree. IP multicast routing enables the source to reach the receivers using this multicast routing tree. Many transport layer multicast protocols try to ensure reliability, thus the term *Reliable Multicast (RM).* The idea is that receivers acknowledge, positively or negatively, data packets back to the source and the source retransmits the lost packets until all the receivers get them. The cost paid for reliability is the bandwidth used for feedback and recovery traffic and the delay introduced in the delivery of multicast data.

One of the most successful approaches used to reduce this overhead and deal with a number of well-known problems of Reliable Multicast is the one that localizes the error recovery by using a hierarchy of proxies. According to this approach, which we call *Hierarchical Reliable Multicast (HRM*), the multicast delivery tree is partitioned into subgroups. Each subgroup has a representative, called proxy, which keeps copies of data packets, collects the feedback from the receivers in the subgroup and locally retransmits the packets, if needed. The subgroups form a hierarchy rooted at the source. A proxy for a downstream subgroup is a receiver of the upstream subgroup.

A lot of real protocols follow this hierarchical approach, [16, 15, 8, 10, 26, 3]. In their context, proxies are called designated receivers or DRs [16,15], log servers [10], group controllers [8], domain managers [26] or proxies [3]. Proxies may be members of the group [8, 16, 15] or special servers [10, 3]; they may be co-located with the routers or not.

The HRM approach has a number of desirable properties that make it scalable. State explosion is avoided because state is needed only for a small subgroup and not for the entire group. The recovery latency is also reduced, because the proxy is closer to the receivers than the source and it has to process less feedback. Limiting the feedback and the retransmissions locally, between the proxy and the subgroup members, saves bandwidth because feedback and recovery traffic affect a few links at the neighborhood of the loss as opposed to the entire multicast tree. The HRM approach may also address successfully the problem of receiver heterogeneity by appropriately grouping together receivers with similar loss characteristics and my allowing the proxies to translate between formats or locally adjust the transmission rate of the session, [3].

The performance of such hierarchical schemes, strongly depends on the underlying multicast tree topology and the loss characteristics, the appropriate partitioning of the tree, the hierarchy of subgroups and the selection of proxies. For example, an appropriate partitioning should put a proxy between the receivers of its subgroup and the source. Poor receivers should be separated behind a proxy, so that they do not bother the rest of the group with their requests and the recovery traffic destined to them. In this study, we focus on the impact of the multicast tree characteristics, i.e. topology and loss rates, on the performance and the placement of proxies.

The structure and the contributions of this paper are as follows. In section 2, we present our model for Hierarchical Reliable Multicast and we discuss to what extent it captures the features of real protocols. Our first contribution, in section 3, is that we

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Conference '00, Month 1-2, 2000, City, State.
Copyright 2000 ACM 1-58113-000-0/00/0000…$5.00.

define and analytically compute performance metrics that capture the "forward"[1] traffic needed to deliver a packet correctly to all members, at least once. Given a multicast tree, this forward traffic cannot be avoided whatever the details of the error recovery scheme are. In section 3.1, we extend the understanding about the ***E[M]*** measure, initially introduced in [1]. In section 3.2, the ***packet-hops*** measure is analytically calculated for the first time. Our second contribution is in section 4; we give guidelines for the optimal placement of proxies, with respect to the above two measures. We compute optimal solutions for special cases and give insights for the general case. This early work resulted in the [17] paper where the algorithm for optimal placement of proxies is completed. Section 5 concludes and discusses future work.

2. MODELING THE HRM PROBLEM

Consider a single-source multicast tree where the root is the source and all the other nodes, intermediate or leaves, are the receivers. We the model shown in Figure 1 to capture the basic characteristics of Hierarchical Reliable Multicast protocols [16, 15, 8, 10, 26, 3]. Let us describe our model.

1. Consider a single-source multicast tree where the root is the source and all the intermediate nodes and the leaves are receivers.
2. The topology and the loss probabilities at the links of the multicast tree are given.
3. The tree is partitioned into subtrees/subgroups that form a hierarchy rooted at the source. Each subgroup has a proxy located at its root. The source is a proxy itself. A node functioning as a proxy for the downstream subgroup may be a simple member for the upstream subgroup.
4. Scheme inside a single subgroup: the proxy multicasts an original data packet to the whole subgroup. All members send feedback back to the proxy, in some way that we ignore here. If one or more receivers lost the packet, the proxy retransmits the lost packet again to the whole subgroup, even if some receivers don't need it. If all receivers have received the packet correctly at least once, then the proxy multicasts the next data packet.
5. Subgroups are independent in the following sense: feedback (ignored here) and transmissions are limited only between a proxy and the members of its subgroup and they do not reach members of any other subgroup.

Before proceeding with the analysis, let us first discuss how realistic this model is.

Assumption 1 is not a strong one. Whether intermediate nodes are receivers or not, does not change our bottom-up analysis, because losses are anyway correlated. Downstream members receive a packet if and only all intermediate nodes on the path from the source have received it.

[1] "Forward" traffic: initial multicast transmission + retransmissions, needed to correctly deliver a packet to all members at least once; it does not include feedback messages.

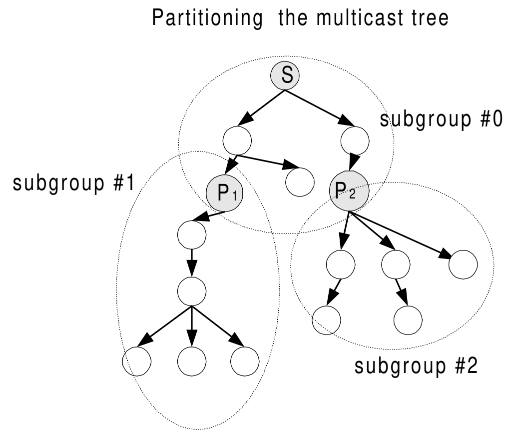

Figure 1. Hierarchical Reliable Multicast

Assumption 2 states that the topology and the loss rates, due to either link failures or congestion, are given for all links of the multicast tree. One might wonder how realistic such an assumption is, given that IP multicast by definition does not reveal the identity of members and allows for dynamic membership. In some cases, it is indeed possible to acquire knowledge about the multicast tree. First, a receiver can use the MTRACE tool to find the multicast route and collect link loss statistics on the path towards the source. This is exactly what Tracer, a component proposed in [13] by Levine et al., does: it traces the multicast routes and collects link loss statistics. [15] also assumes a backtrack capability, that allows each receiver to track its path back to the multicast source. Second, recent work by Ratnasamy and McCanne [23], made it possible to infer the topological structure and the loss probabilities based only on end-to-end loss prints and their correlation. Finally, even if topology and loss rates are not known in real life, our methodology may still be useful for comparison and assessment purposes.

Assumption 3 is realistic and captures what hierarchical reliable multicast protocols, [16, 15, 8, 10, 26, 3] actually do: build a hierarchy of subgroups and localize the error recovery.

Assumption 4 describes a generic scheme inside each subgroup: proxy-based error recovery with multicast retransmissions. There are many other available options very well summarized in [9], analytically compared in a number of studies [22, 25, 12]. Details of feedback and congestion control are ignored here. Our performance metrics capture the total transmissions needed to deliver a packet correctly to all members. If there is no loss, there should be only one transmission per packet. If there are losses, there are extra retransmissions which create additional recovery traffic and increase the delivery latency. These extra transmissions are due to the underlying multicast tree loss characteristics and they cannot be avoided, whatever error recovery and flow control schemes one chooses inside a subgroup. In reality, there may be additional feedback traffic and latency introduced by the error recovery scheme, which are outside the scope of this study.

Assumption 5 states that subgroups are separated from each other. Retransmissions are limited locally, i.e. they reach the members of the subgroup and no other node. For this purpose, the proxy needs a way to address the members of its subgroup. In [8] a new multicast address per subgroup is used. Other schemes [5, 26] use the group's multicast address and TTL scoping. [16] uses subcasting and TTL scoping. Other options for localizing the error recovery messages include administrative scoping and explicitly knowing the members of the subgroup. So, *Assumption 5* captures that all HRM schemes try to localize the recovery traffic, among members of the same subgroup and they achieve to do so more or less accurately.

Our model does not account for the ability of router-assisted proxies, such as [24, 21], to remember where the losses occurred and selectively retransmit only to those subtrees. The proxies in our model always multicast a lost packet to the entire subgroup.

3. PERFORMANCE ANALYSIS

In this section we explore two metrics, which are very commonly used in the performance analysis of Reliable Multicast:

1. the average number of ***transmissions*** by the source/proxy or ***E[M]*** and
2. the average number of links crossed by the above transmissions or ***packet-hops***

needed either for the source of Figure 1 to reliably deliver a single packet to the entire group, or for the proxy of Figure 2 to reliably deliver a packet to the entire subgroup.

3.1 The E[M] Measure

3.1.1 Meaning and related work

E[M] is an important measure, which first appeared in [1] and ever since has been used in most analytical work on Reliable Multicast [18, 20, 19, 22, 25, 12].

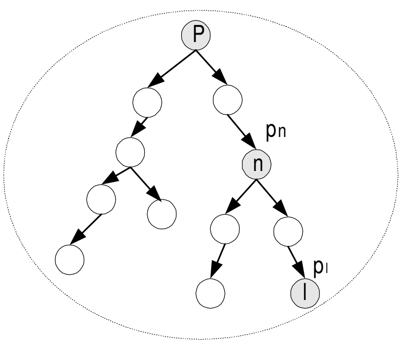

Figure 2. Inside a subgroup

E[M] is defined to be the expected number of times that a packet should be multicast (original transmission + retransmissions) by the source, until all group members receive it correctly, at least once. It depends on the shape of the tree and the link loss probabilities. Intuitively, the closer *E[M]* is to 1, the better, because *E[M]* close to 1, means that most of the group members received the packet correctly at the first transmission. Thus *E[M]* gives an idea about the bandwidth wasted in (re)transmissions and about the reliable transfer time. *1/E[M]* also provides a lower bound to the source processing rate, [11].

Let us consider a single subgroup with *L* receivers and the proxy *P* being the root shown in Figure 2. Our model states that (re)transmissions are multicast to the whole group and that the topology and loss rates of the tree are given. Let p_n be the probability of loss on the link leading to node n . We calculate the average number of times *E[M(P)]* a packet has to be multicast from the proxy *P* before all the nodes receive it at least once. One of them is the original transmission and the rest *E[R(P)]* are retransmissions due to losses. To find these expected values, [1] recursively calculated the CDF, first for leaf ($\underline{l}$), then for intermediate nodes (n) and finally for the proxy *P*. $F_n(i)$ = *Pr[all nodes from n and below got the packet at most in i attempts]*.

$$F_l(i) = \Pr[M(l) \le i] = 1 - p_l^i$$

$$F_n(i) = \sum_{u=0}^{i-1} \binom{i}{u} p_n^u (1-p_n)^{i-u} \prod_{c \in child(n)} Fc(i-u)$$

$$F_P(i) = \prod_{c \in child(P)} Fc(i)$$

$$E[M(P)] = \sum_{i=0}^{\infty} (1 - F_P(i))$$

$$E[R(P)] = E[M(P)] - 1$$

Figure 3. Calculation of E[M]

Although these formulas are general enough to capture any tree topology and any loss probabilities, they don't give any intuition and they are considered computationally intense, [19]. In practice, either simulation or approximations are used for the calculation of *E[M]*, [20].

3.1.2 Reduction Techniques

In this section, we present a way to find the ***equivalent link*** of any subtree. By equivalent link we mean a link with only one receiver that has the same loss behavior in terms of *E[M]* as the subtree it substitutes for. In other words, if a subtree is replaced by its equivalent link, the source will still have to transmit *E[M]* times on average. This procedure is by itself a way for computing *E[M]* as accurately as the general formula, [1]. In addition it provides insights that the general formula does not. Finally this technique can be used to make an algorithm for the selection of proxies.

Model for a single link:

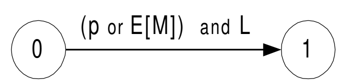

Figure 4. Model for a single link

A link may be a real link or it may summarize a part of the multicast tree. It has two characteristics:

- the loss probability ***p*** on the link or equivalently the number of transmissions from node 0 to reach node 1: *E[M]=1/(1-p).*
- and a weight ***L*** indicating the cost of retransmissions over this link. For example, L may be the number of links affected by these transmissions. Then the total bandwidth/ load" is *E[M]*L.*

Links in a row.

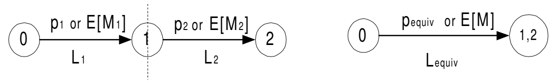

Figure 5. Links in a row

If links 0→1 and 1→2 links are separated, then each one of them is characterized by:

$$p_1, E[M_1] = 1/(1-p_1), L_1, Load_1 = L_1/(1-p_1)$$
$$p_2, E[M_2] = 1/(1-p_2), L_2, Load_2 = L_2/(1-p_2)$$

If both 1,2 are in the same subgroup, their equivalent is shown at the right of Figure 5.

$$p_{equiv} = 1-(1-p_1)(1-p_2)$$
$$F_0(i) = \Pr[M(0) \le i] = 1 - p_{equiv}^i$$
$$E[M] = \frac{1}{1-p_{equiv}} = \frac{1}{(1-p_1)(1-p_2)} = E[M_1] \cdot E[M_2]$$
$$Load = E[M] \cdot L_{equiv}$$

Similarly, for more than 2 links in a row, the equivalent link has: $p_{equiv}=1-(1-p_1)\ldots(1-p_n)$ or $E[M]=E[M_1]\ldots E[M_n]$ and $L<L_1+\ldots L_n$

Links in a star.

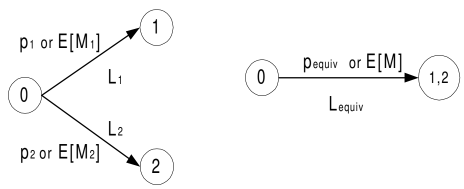

Figure 6. Links in a star

If links 0→1 and 1→2 are independent from each other, as shown at the left of Fig.6, then they have characteristics:

$$P_1,\ E[M_1]=1(1-p_1),\ L_1,\ Load_1=L_1/(1-p_1)$$

$$P_2,\ E[M_1]=1(1-p_2),\ L_2,\ Load_2=L_1/(1-p_2)$$

If the 2 links are in the same subgroup, then their equivalent link is shown at the right of Figure 6.

$$F_0(i) = \Pr[M(0) \le i] = (1-p_1^i)(1-p_2^i) = 1 - p_1^i - p_2^i + (p_1 p_2)^i$$
$$\frac{1}{1-p_{equiv}} = E[M] = \sum_{i=0}^{\infty}[1-F_0(i)] = \sum_{i=0}^{\infty} p_1^i + p_2^i - (p_1 p_2)^i = \frac{1}{1-p_1} + \frac{1}{1-p_2} - \frac{1}{1-p_1 p_2}$$

A transmission multicast by node 0 affects all links, therefore: $L_{equiv} = L1 + L2$ and $Load = E[M] \cdot L_{equiv}$. Note that $E[M]<E[M_1]+E[M_2]$ and the benefit of separating 0→1 from 0→2 into different subgroups is:

$$Load_1 + Load_2 - Load = \left(\frac{L_1}{1-p_2} - \frac{L_1}{1-p_1 p_2}\right) + \left(\frac{L_2}{1-p_1} - \frac{L_2}{1-p_1 p_2}\right)$$

Indeed, the first parenthesis describes the retransmissions destined for node 2 only, that unnecessarily bother node 1. Similarly, the 2nd parenthesis describes retransmissions useful only for node 1 that bother node 2.

For 3 links in a star, the equivalent link has $L = L_1 + L_2 + L_3$ and

$$E[M] = \frac{1}{1-p_{equiv}} = \frac{1}{1-p_1} + \frac{1}{1-p_2} + \frac{1}{1-p_3} - \frac{1}{1-p_1 p_2} - \frac{1}{1-p_2 p_3} - \frac{1}{1-p_3 p_1} + \frac{1}{1-p_1 p_2 p_3}$$

Similarly, for $n>3$ links, the rule for the equivalent link is as follows. All links are affected by all transmissions, so "take the sum of weights" to get the equivalent weight. "Take the union of errors on links *1,2..n*", to get the equivalent *E[M]*. Indeed, $1/(1-p_1)$ are the transmissions due to error on link 1, $1/(1-p_1 p_2)$ are the transmissions due to errors on both link#1 and link 2, , $1/(1-p_1) - 1/(1-p_1 p_2)$ are the retransmissions due to errors on link 1 and not on link 2, and similarly for the rest of the terms.

3.1.3 A simple example of reduction

Consider the following tree, [1], with the same probability of error on all links, $p = 0.03$. This is a reasonable p to consider according to measurements on MBONE found in [27, 6, 1, 18, 8]. First, let's find the equivalent link of the whole tree and the total load, without the use of any proxy. The reductions are done by successively applying the rules for the chain and the star cases.

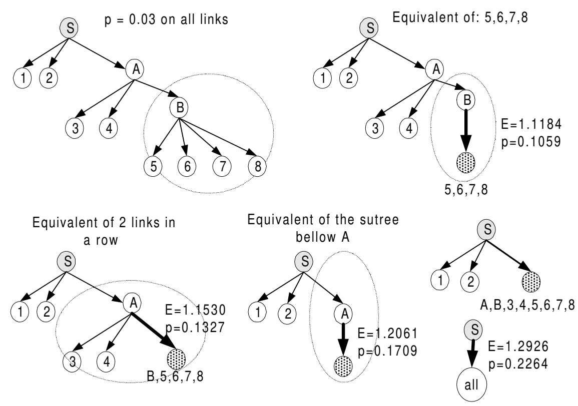

Figure 7. Example of reduction

E[M]=1.2926 means 29% more transmissions by the source than in the case that there is no loss.

3.1.4 A more realistic example: LGMP

In [8], simulation is used and a fixed number of packets are sent until all group members receive them. Hoffman et al. use the topology of Fig. 8, taken from the [27], to experiment with different sizes of subgroups and different hierarchies. The measure used by [LGMP] to capture the network load, is the total number of packets traveling in the network relative to the initial number of packets sent by the source. This is similar to E[M], the average number of transmissions from the source for one packet to be correctly transferred. We apply our reduction technique on the scenarios considered in [8] and find that our results agree with the simulation. Therefore, one could have chosen among the [8] hierarchies analytically.

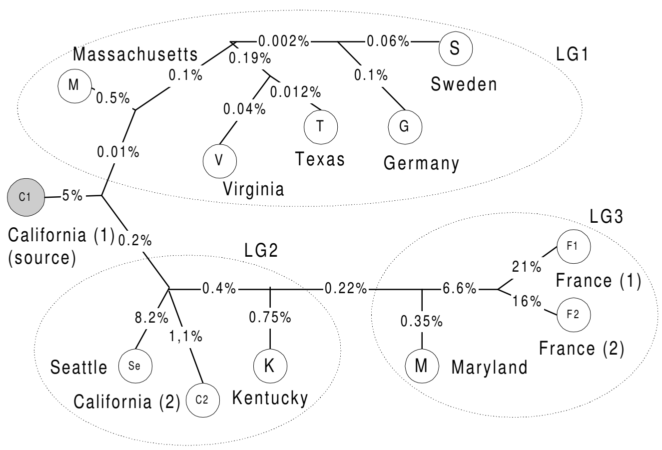

Figure 8. MBONE scenario used in [8]

Scenario 0: one flat multicast tree. All retransmissions are re-multicast from the source S. By applying our reduction techniques, we find that the equivalent probability of error is p=0.4113 and the expected number of transmissions is E[M]=1.6986. The simulated netload was 1.827.

Scenario Ib: subgrouping. In this scenario the participants are grouped in three subgroups as shown in Figure 9:

- LG1={Massachusetts,Virginia,Texas, Sweden, Germany},
- LG2={Seattle,California(2), Kentucky},
- LG3={Maryland, France(1), France(2)} .

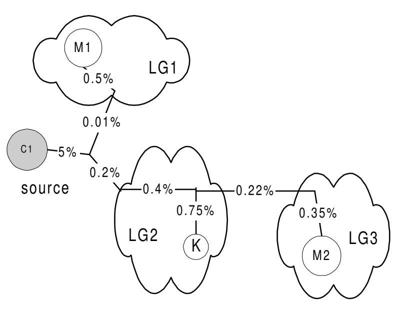

Figure 9. Subgrouping scenario Ib

Given the (Ib) subgrouping LGMP considers two possible hierarchies:

The **Ib-f scenario,** considers a flat hierarchy, i.e. all subgroups in Figure 9, first try to locally recover from loss and then request retransmissions directly from the source S. In this case, the source S "sees" 3 receivers, the best of each local subgroup: Massachusetts, connected through a link with loss (5% OR 0.01% OR 0.5%), Kentucky, connected through a link with loss (5% OR 0.2% OR 0.4% OR 0.75%) Maryland, through (5% OR 0.2% OR 0.4% OR 0.22% OR 0.35%). Each one of proxies M1, K, M2 sees only the other members of its local subgroup. In this case, we found analytically that E=1.0845 or p=0.0779 is seen by the sender. Simulation gave load = 1.1198.

The **Scenario Ib-h,** considers the same subgroups of Fig.9, but organizes them into an hierarchy: LG1 and LG2 request lost packets from the source S, but LG3 requests lost packets from LG2. So, the source S "sees" only Massachusetts and Kentucky, Massachusetts and Maryland "see" only their local group. K, apart from its local group, "sees" also Maryland connected through loss (0.75% OR 0.22% OR 0.35%). In this case, each subgroup separately has the same load as in Ib-f, but the source has to deal only with two receivers, Kentucky and Massachusetts, and thus is has less load: E=1.0723 or p=0.0674 is seen by the source. Simulation showed load=1.059.

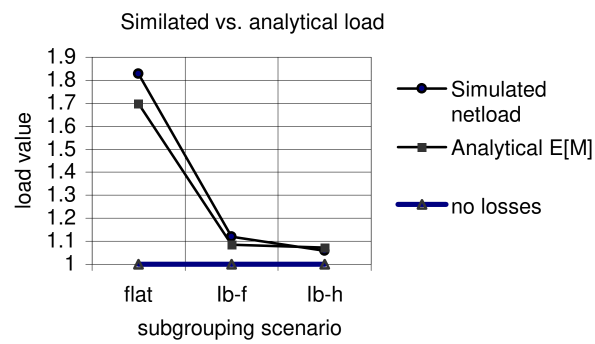

Figure 10. Analysis vs. Simulation

Figure 10 shows that the analytically found and the simulated loads are very close. Although the actual number of transmissions *M* is a random variable, the measured *M* should be close to the average *E[M]*, because the experiment is repeated many times (40000 original data packets). The small differences are due to the deviations of the model from the actual protocol. First, in our model there is exactly one retransmission per lost packet, while in practice there may be more, due to bad tuning of timeouts. This is why the analytical *E[M]* is below the simulated one. Also, the model assumes that only roots of subtrees send retransmissions while in LGMP, other members may do so. However, this doesn't cause a large discrepancy because in our calculations for source's *E[M]*, we took into account the best receivers as LGMP actually does. Finally, what appears as a single node in Figure 8, summarizes 20 nodes. Despite these deviations, our model captures the dominant part of *E[M]*.

3.2 The Packet-Hops Measure

3.2.1 Meaning and related work

Although *E[M]* gives an idea about the bandwidth usage, the exact amount depends on the topology (how many links are traversed by each retransmission). Packet-hops, defined as the number of links actually traversed by the *E[M]* transmissions, gives the exact amount of bandwidth used. It can be summed up over all independent subgroups to give the total bandwidth used. So, this measure characterizes the performance of the whole group, and it is therefore an appropriate objective to be minimized, unlike *E[M]*. Packet-hops can be easily counted by simulation. For example, [15] tracks the number of router forwards rather than the number of distinct packets generated by end-stations. However, it was not calculated analytically and apart from simulations, only coarse approximations [20,11] have been used so far.

3.2.2 Analytical calculation of average packet-hops

The usual approximation that is used for packet-hops is its upper bound:

$$E[Packet\_hops] \leq E[M]*(number \;\; of \;\; links) \quad (1)$$

However, not all multicast transmissions traverse all links, so this is only an upper bound, whose tightness depends on the form of the multicast tree and on how many links are affected by a loss. It is a tight bound for topologies close to a star and for small loss rates and a loose bound for topologies resembling to a chain and/or having large loss rates. In this section we calculate analytically the packet-hops for all cases without approximations.

First, for ***star topologies*** the number of links affected on average is exactly $E[M]*(number \;\; of \;\; links)$. Such topologies are based on the [27] measurement paper and they are widely used in the performance analysis of RM schemes, [20, 11].

We now calculate the average number of packet-hops for a ***chain topology.*** A multicast transmission from the source 0 reaches only the links until the point where the loss occurred. Then a new transmission starts again from node 0. In this case the number of links traversed, can be calculated as the average time to absorption by the last node L, starting from 0, at the following Markov chain of Figure11.

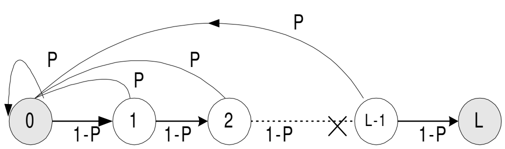

Figure 11. packet-hops for a uniform chain

Let E_i be the average time to absorption by state L, given that we are in state i. Being in state E_i means that the packet coming from the source was dropped just before the node *i.*

$$E_L=0,\; E_{L-1}=1+pE_0,\; E_{L-2}=1+pE_0+(1-p)E_{L-1},\ldots.$$

$$E_0=1+pE_0+(1-p)E_1$$

Therefore,

$$E_0 = \frac{1}{(1-p)^L} + \frac{1}{(1-p)^{L-1}} + \ldots + \frac{1}{(1-p)} = \frac{1}{p}\left\{\frac{1}{(1-p)^L} - 1\right\} \quad (2)$$

This procedure can also be applied to a non-uniform chain. It is difficult though for the degrees greater than 1, even for a simple 2-star:

$$E[\#packet\_hops] = \frac{1+2p}{1-p^2}\cdot 2$$

Finally, we calculate the average number of packet-hops for ***a general tree.*** We already know how to calculate the average number of transmissions $E[M]$. Let us call P_l the probability that the link l starting at node n is affected by a transmission, i.e. the probability that there is no error on the path from the source to node *n*:

$$P_l = \Pr\{a \;\; packet \; arrives \; at \;\; node \;\; n\} = \prod_{i \; on \; the \; path \; from \; S \; to \; n}^{i=n} (1-p_i) \quad (3)$$

Given the topology and the link loss probabilities, we can pre-calculate a table with entries (l, P_l) for all links *l*. P_l also equals the percentage of transmissions that affect the link *l*, i.e., $P_l \cdot M$ out of the M total transmissions affect this link. So the link rooted at node n is crossed on average $P_l \cdot E[M]$ times. Add over all links and we have the total number of average packet-hops $\sum_{over \; all \; links \;\; l} P_l \cdot E[M]\cdot$ Furthermore:

$$E[\#packet\_hops] = E\left[\sum_l P_l \cdot M\right] = \sum_l E[P_l \cdot M] = \sum_l P_l \cdot E[M] = E[M]\cdot \sum_l P_l$$

In conclusion, the average number of packet-hops is:

$$E[\#packet\_hops] = E[M]\cdot \sum_l P_l \quad (4)$$

We can reach the same result, more formally: Let l be the link starting at node n and $X_{il} = \begin{cases} 1, & w.p.P_l \\ 0, & w.p.(1-P_l) \end{cases}$ be the random variable indicating whether transmission i crosses link l or not. Then, the number of transmissions crossing link l is $X_l = \sum_{i=1}^{M} X_{il}$ and on average:

$$E[X_l] = E\left[\sum_{i=1}^{M} X_{il}\right] = E[M]\cdot E[X_{il}] = E[M]\cdot P_l$$

Add over all links $l = 1,2..L$ and get the same result as in (4).

Test 1. Let us verify the above formula in the special case of the uniform chain of Figure11. (2) and (4) give the same result.

$$P_n = \prod_{i=0}^{i=n-1}(1-p) = (1-p)^{n-1}, n = 1\ldots L-1, \qquad E[M] = \frac{1}{(1-p)^L}$$

$$E[\#packet\_hops] = E[M]\cdot \sum_{n=1}^{L-1} P_n = \frac{1}{(1-p)^L}\cdot \sum_{n=1}^{L-1}(1-p)^{n-1} =$$

$$\frac{1}{(1-p)^L}\cdot\frac{1-(1-p)^L}{1-(1-p)} = \frac{1}{(1-p)^L}\cdot\frac{1-(1-p)^L}{p} \Rightarrow$$

$$E[\#packet\_hops] = \frac{1}{p}\left\{\frac{1}{(1-p)^L} - 1\right\}$$

Test 2. In the 2-star case, the average number of packet-hops is $E[\#packet\_hops] = \frac{1+2p}{1-p^2}\cdot 2$, calculated either from (1), or (4) or as the average time to absorption.

4. PLACEMENT OF PROXIES

In the previous section, we showed how to compute *E[M]* in a bottom-up approach and how to find packet-hops accurately or using approximations. These two measures were calculated for a single group, where the only proxy is the source. In this section, we try to appropriately partitioning the multicast tree of Fig.1 into separate subgroups, each of them having its own proxy. We first discuss the special cases of uniform trees and chains. By "uniform" we mean that all links have the same loss probability *p*. Then, we try to optimally place proxies with respect to both *E[M]* and packet-hops.

4.1 Special Cases

4.1.1 Uniform tree

Although the formulas for *E[M], E[R]* and *packet-hops* are quite hard to calculate in the general case, it has been proved in [18], that there is a simple and tight approximation if the loss probability is the same, *p*, for all *L* links of the tree: $E[R(S)] \cong pL$. This holds for trees of any shape, as long as $pL \prec 1$. The bandwidth, wasted in retransmissions is now at most $pL \cdot L = p \cdot L^2$.

Consider first a single subgroup *i*, of size L_i with proxy P_i, shown in Fig.2. The $E[M(P_i)]$ transmissions from P_i, are multicast to the entire subgroup, so they cross at most L_i links. $L_i \cdot E[R(P_i)]$ is an upper bound to the wasted bandwidth for a packet transmitted by proxy P_i to correctly reach the entire subgroup-*i*. Consider now the whole multicast tree, shown in Fig.1, partitioned into separate subtrees. $\sum_{subgroup-i} L_i \cdot E[R(P_i)]$ is the bandwidth per packet wasted in retransmissions. This last one will be our objective to minimize.

Proposition 1. The best way to place 2 proxies in a uniform tree, is by assigning half nodes to each subgroup.

Proof. If the only proxy is the source, we waste $p \cdot L^2$ bandwidth in retransmissions. If we put one more proxy, we partition the multicast group in 2 independent subtrees, the 1st one with L_1 links rooted at the source and the 2nd with L_2 links rooted at the 2nd proxy. Now $p \cdot L_1{}^2$ for retransmissions are wasted in the 1st subgroup and $p \cdot L_2{}^2$ in the 2nd. The benefit from having two proxies instead of the source is: $pL^2 - \left(pL_1{}^2 + pL_2{}^2\right) = p(L_1 + L_2)^2 - \left(pL_1{}^2 + pL_2{}^2\right) = 2pL_1L_2$. To achieve max benefit:

$$\left.\begin{array}{r}\max 2pL_1L_2 \\ s.t. L_1 + L_2 = L.\end{array}\right\} \Rightarrow L_1 = L_2 = \frac{L}{2}, \max = \frac{pL^2}{2}$$

Proposition 2. The best way to place n proxies in a uniform tree, assigns *L/n* nodes to each subgroup.

Proof. The benefit of having n proxies instead of just the source alone is $pL^2 - \left(pL_1{}^2 + pL_2{}^2 + \ldots + pL_n{}^2\right)$. Using Lagrange multipliers we find the max benefit:

$$\left.\begin{array}{l}\max\left\{pL^2 - \sum_{i=1}^{n} pL_i^2\right\} \\ s.t. \sum_{i=1}^{n} L_i = L.\end{array}\right\} \Rightarrow L_1 = \ldots = L_n = \frac{L}{n}, \max = \frac{(n-1)pL^2}{n}$$

4.1.2 Chains

The chain is an extreme case of a tree. It is easy to deal with it analytically and it gives insights about paths on the multicast tree. All nodes of a chain correctly receive a packet, at least once, if and only if the last node received it. A packet is correctly received by the whole group with probability of success $(1-p)^L$.

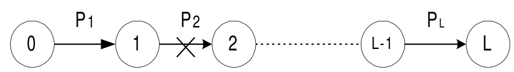

Figure 12. a chain topology

The results for the uniform tree hold also for the special case of uniform chain. It takes $E[M] = 1/(1-p)^L$ transmissions on average. An upper bound to the packet-hops is: *Upper bound* $= A(p,L) = L \cdot E[M] = L/(1-p)^L$

Because of the simplicity of the topology, we are able to use the more accurate measure of packet-hops, discussed in section 3.2: *Avg packet-hops* = $B(p,L) = \left\{1/(1-p)^L - 1\right\}/p$

By placing one proxy at node k, we partition the chain into 2 subgroups and we do better than before in terms of both measures. For example, for measure A:

$$A_1(p,L,k) = A(p,k) + A(p,L-k) = \frac{k}{(1-p)^k} + \frac{L-k}{(1-p)^{L-k}}$$
$$\prec \frac{L}{(1-p)^L} = A(p,L)$$

The optimal node to place one proxy in the uniform chain, $k = L/2$ with respect to both measures, is the middle node. We can prove this (i) by maximizing both benefits over the position of proxy k or (ii) by the symmetry between the *(k,L-k)* and *(L-k,k)* partitionings.

Note1. Benefits, under both measures, are convex in *k*, so a proxy near the middle performs "almost" optimally. The sensitivity of the optimal choice with *k* with *p, L* will be important when we will make the algorithm.

Note2. The best way to place n proxies in a uniform chain, with respect to both measures, is to equally spread them every L/n nodes. As we put more and more proxies, the case degenerates to hop-by-hop acknowledgement and

$$A^*_n(p,L), B^*_n(p,L) \xrightarrow[n \to L]{} L/(1-p)$$

Similar results hold for non uniform chains. The best node to put one proxy is the middle one, where "middle" means the node K which equally divides the probabilities of success at its left and at its right: $(1-p_1)..(1-p_K) \approx (1-p_{K+1})..(1-p_L)$

4.2 The General Case

4.2.1 Bounding E[M] per subgroup

In section 3.1, we gave a bottom-up way to compute the performance of a subtree. Now, we use this bottom-up approach to partition the tree into subtrees with desirable specified *E[M]* each. It is important to bound E[M] for the source in particular, because small E[M] means that the source can transmit at faster rates. The idea is that as we start from the leaves and proceed upstream, we accumulate more and more nodes in the equivalent link which becomes "heavier and heavier" in terms of *E[M]*. When this equivalent link becomes "heavy" enough and *E[M]* exceeds the desired bound, we decide to put a proxy there and we cut the equivalent link from tree. We continue with the rest of the tree until we are left only with one equivalent link from the source.

As an example, let us consider the tree of Figure 7 and try to place proxies in a bottom-up way as shown in Figure 13.

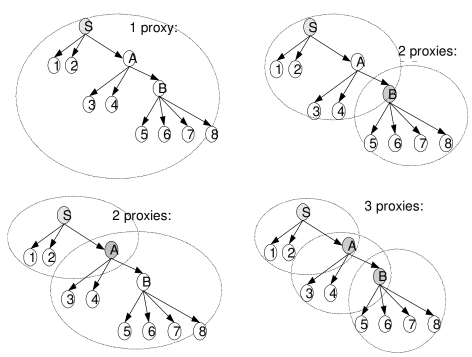

Figure 13. Example of partitioning

It seems that in terms of *E[M],* one additional proxy is enough to relieve the source. In terms of *packet-hops,* it seems that adding the capability of subcasting to the branch where the loss occurred is more important than adding more proxies without this capability. The maximum saving in packet-hops is 20.24%, with proxies subcast able on all S,A,B. Based on given constraints, one could choose among the possible subgroupings of Table 1.

Table 1. Comparison of possible partitionings

#Proxies	E[M]	Packet-Hops	Packet-hops (+subcast)
1(S)	S: 1.2926	12.926	12.0091
2(S,A)	S: 1.0901, A:1.2061	11.6773	10.8988
3(S,A,B)	S,A:1.0901 B:1.1184	11.0142	10.3093

4.2.2 Minimizing packet-hops

Unlike *E[M], packet-hops* can be added across separate subgroups, and it is an appropriate global objective to minimize. So, a second problem is how to place a fixed number of proxies in order to minimize the bandwidth used (packet-hops) across the whole tree.

It is not obvious how to reduce this problem to a well-known location problem, [4], as it might look at first. The K-means problem, a well-studied location problem for which good algorithms are known, is probably the closest. Indeed, independent unicast (re)transmissions from the source to the receivers, can be easily modeled as a location problem, similarly to the placement of web proxies in [14]. However a multicast connection cannot be modeled in a straightforward way because the cost from each node to the proxy does not depend only on the path from a member to the proxy, but also on the loss behavior of the rest of the group. This happens because multicast transmissions destined to lossy members bother the rest of the group.

We gave the optimal solution in the special cases of uniform trees and chains. In [17] which followed this early work, we develop a dynamic programming algorithm which places a fixed number of proxies on a multicast tree, in a way that the resulting packet-hops across the whole tree, is within a chosen precision of the minimum. At each step, our dynamic program examines a subtree and takes into account both its own loss behavior and the effect of the rest of the tree on it. In the same work we give a simpler algorithm that achieves the optimal placement for unicast retransmissions and we discuss extensions to include both bandwidth and delay constraints, variable number of proxies and ability to choose among multicast and unicast retransmissions.

5. SUMMARY AND FUTURE WORK

In this paper we presented a model for Hierarchical Reliable Multicast and we discussed how realistic it is. We studied in detail two performance metrics, the average number of transmissions and the average number of packet-hops, for reliable transfer. For the first one, we gave some reductions techniques and examples comparing to known simulation results. We also computed the second one analytically, while so far it was only approximated or simulated. Finally, we gave some insights on the optimal partitioning of a tree with respect to these measures. The general solution will be given in future work [17]. We are also currently considering extensions to our model to include feedback and other realistic RM mechanisms.

6. ACKNOWLEDGMENTS

Thanks to Jose Miguel Pulido for his comments and for suggesting comparison of analytical and simulation results.

7. REFERENCES

[1] P.Bhagwat, P.Mishra, S.Tripathi "Effect of Topology on Performance of Reliable Multicast Communication", INFOCOM'94.

[2] B.Briscoe, P.Bangall, "Taxonomy of Communication Requirements for Large-scale Multicast Applications" Internet draft.

[3] Y. Chawathe, S. McCanne, E. Brewer, "RMX: Reliable Multicast in Heterogeneous Networks", INFOCOM '00.

[4] M.Daskin, "Network and Discrete Location Theory", J.Wiley 1995

[5] S.Floyd, V.Jacobson, C.Liu, S.McCanne, and L.Zhang, "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing", IEEE/ACM Transactions on Networking, December 1997.

[6] M.Handley, "An examination of MBONE Performance", Jan. 1997.

[7] M.Hoffman, "Enabling Group Communication in Global Networks", Global Networking '97.

[8] M.Hoffman, "Impact of Virtual Group Structure on multicast performance".

[9] M.Hoffman, J.Nonnenmacher, e.a, "A Taxonomy of Feedback for multicast", Internet draft, June 1999.

[10] H. Holbrook, S. Singhal and D. Cheriton, "Log-Based Receiver-Reliable Multicast for Distributed Interactive Simulation", ACM SIGCOMM `95.

[11] S.Kasera, J.Kurose, D.Towsley, "A comparison of server-based and receiver-based local recovery approaches for scalable reliable multicast"

[12] B.N. Levine and J.J. Garcia-Luna-Aceves, "A Comparison of Reliable Multicast Protocols", in ACM Multimedia Systems Journal, August 1998

[13] B.Levine, S.Paul, J.J. Garcia-Luna-Aceves. "Organizing multicast receivers deterministically by packet-loss correlation"

[14] B.Li, M.Golin, G.Italiano, X.Deng, K.Sohraby, "On the optimal placement of web proxies in the Internet", INFOCOM '99

[15] D.Li, D.Cheriton, "OTERS (On-Tree Efficient Recovery using subcasting): A Reliable Multicast Protocol", ICNP '98.

[16] J.Lin ,S.Paul, "RMTP: A Reliable Multicast Transport Protocol", INFOCOM' 96.

[17] A.Markopoulou, S.Guha, "Optimal placement of proxies for Hierarchical Reliable Multicast", submitted to NFOCOM'01.

[18] J.Nonnenmacher, E.Biersack, "Performance Modeling of Reliable Multicast Transmission", INFOCOM '97.

[19] J. Nonnenmacher, E. Biersack, D.Towsley, "Parity-Based Loss Recovery for Reliable Multicast Transmission", Technical Report 97-17, UMASS, March 1997.

[20] J. Nonnenmacher, M. Lacher, M. Jung, E. Biersack, G. Carle, "How bad is Reliable Multicast without Local Recovery?", INFOCOM ' 98

[21] C. Papadopoulos, Parulkar, Varghese, "An Error Control Scheme for Large-Scale Multicast Applications", INFOCOM '98.

[22] D.Towsley, J.Kurose, S.Pingali, "A Comparison of sender and receiver-initiated reliable multicast protocols", JSAC, vol15, no3, April 1997.

[23] S.Ratnasamy and S.McCanne, "Inference of Multicast Routing Trees and Bottleneck Bandwidths using, End-to-end Measurements", INFOCOM'99

[24] T.Speakman, "Pretty Good Multicast (PGM): Transport Protocol Specification", Internet Draft 1998

[25] M.Yamamoto, D.Towsley, J.Kurose, H.Ikeda, "A delay analysis of sender and receiver initiated reliable multicast protocols"

[26] R.Yavatkar, J.Griffioen, M.Sudan, "A Reliable Dissemination Protocol for Interactive Collaborative Applications", ACM Multimedia 95.

[27] M.Yajnik, J.Kurose, D.Towsley, "Packet Loss Correlation in the MBONE", IEEE Global Internet Conference 1996.