

# Average Bandwidth and Delay for Reliable Multicast

Rajarshi Gupta and Jean Walrand  
University of California, Berkeley  
{guptar, wlr}@eecs.berkeley.edu

## 1 Introduction

The number of real-time applications that use multicasting [DC90] has been increasing rapidly over the past few years. These applications have tight delay requirements but tolerate packet errors. Other applications, such as software or news distribution, would benefit from reliable one-to-many multicasting (see e.g., [LP96, D+97]). The necessity therefore is for a protocol that ensures reliability for multicasting in a scalable and efficient manner. In this paper, we study the bandwidth utilization and delays involved in the correction of errors in one-to-many multicasting.

We begin with a discussion of a promising class of such strategies. In the scheme outlined in RMTP [LP96], certain hosts/routers act as specialized agents called *Designated Receivers (DR)* that have the purpose of accumulating nacks and maybe handle retransmissions. These DRs are spread throughout the tree, and every host is assigned a ‘parent DR’.

**Non-caching Scheme:** In this simpler DR scheme, the DRs act only as nack processors. When a host detects a lost packet, it sends a nack (possibly after waiting for a timeout). However, instead of sending the nack to the sender of the packet, the nack is sent to the designated DR for the host. The DR waits for the other hosts in the subtree to nack for the same packet and sends only a *single* nack to its parent DR. This process continues until the source receives a nack, and retransmits the packet. The advantage is that the sender only receives a few nacks, and the nack implosion is avoided.

**Caching Scheme:** In a more advanced system, the DRs cache the data that passes through them. Depending on the size and duration of the session, they may cache the entire session, or a portion thereof. If a DR receiving nacks does not have the packet, it acts as before — sending a single nack up to its parent. However, when it has the requested packet cached, the DR handles the retransmission itself, without propagating the nack upwards.

## 2 Average Bandwidth

### Notation

Before we present the derivations, certain notations need to be clarified. Note that we assume a perfectly uniform multicast tree with equally likely and independent losses on the links. We use the following notations:

$p$  = loss probability on each link;

$w$  = order of the tree, i.e., every node has  $w$  children;

$h$  = distance between DRs, i.e., every  $h^{th}$  level from the host consists of DRs;  
 $b$  = total depth of the tree in levels of DRs, i.e., actual depth of the tree is  $b \times h$ ;  
 $k \in [1, h]$  is a variable used to denote the distance of the host under consideration from its DR;  
 $i \in [0, b - 1]$  is a variable used to denote the position of the DR being considered;  
 Thus, when a host is designated by  $(i, k)$  its actual distance from the source is then  $ih + k$ .

## 2.1 Number of Transmissions Required to Transmit Packet to All Receivers $\alpha(H, w, p)$

As an important measure, we calculate the expected number of transmissions required to transmit a single packet to *all* receivers in the multicast tree. We utilize the models devised in [BMT94] and [NB96] but generalize their specialized tree to a general uniform tree with  $w$  children of each node. Every link has a loss probability of  $p$ , and the losses are independent.

Denote by  $T(n)$  the number of transmissions required for a packet to arrive at node  $n$  and all its children, given that the packet arrives at the parent of node  $n$  at each time.

Let  $F_n(i) = P(T(n) \leq i)$ , for  $i \geq 1$ . If we let  $s$  denote the source, then  $F_s(i) = P(T(s) \leq i)$ .

Let the hosts that are one level away from the source be called children  $c1$ , the hosts two levels away called  $c2$  and so on. Also, note that any host (source included) has exactly  $w$  children. Then,

$$F_s(i) = \prod_{c1 \in \text{child}(s)} F_{c1}(i) = (F_{c1}(i))^w$$

$$F_{c1}(i) = \sum_{j=0}^{i-1} \binom{i}{j} p^j (1-p)^{i-j} \prod_{c2 \in \text{child}(c1)} F_{c2}(i-j) = \sum_{j=0}^{i-1} \binom{i}{j} p^j (1-p)^{i-j} (F_{c2}(i-j))^w.$$

The recursion eventually finishes when it reaches the leaf nodes, where  $F_l(i) = 1 - p^i$ .

Hence, the expected number of transmissions required to transmit a single packet to *all* the receivers in the multicast tree is denoted by

$$\alpha(H, w, p) = E[T(s)] = \sum_{i=0}^{\infty} (1 - F_s(i)).$$

## 2.2 Calculating the Caching Average Bandwidth $\rho_{avg}^C$

In our calculation for the average bandwidth per packet, we first calculate the bandwidth experienced at some DR (located  $i$  levels away from the source) and average this across all the DRs at all levels. Since the DRs themselves are hosts, and the tree is uniform with DRs located in every  $h^{th}$  level, averaging the bandwidth across all the levels of DRs gives us a true estimate of the average.

Let  $\rho^C(i)$  denote the expected bandwidth utilized per packet at a DR located  $i$  levels away from the source. The bandwidth utilized incorporates the sum of all the packets and nacks that pass through this DR, including both incoming and outgoing ones. We calculate  $\rho^C(i)$  as the sum of five quantities that we explain below:

$$\rho^C(i) = N_1 + N_2 + N_{3A} + N_{3B} + N_4.$$

Averaging this over all the levels of DRs,

$$\rho_{avg}^C = \frac{\sum_{i=0}^{b-1} \rho^C(i) \cdot w^{ih}}{\sum_{i=0}^{b-1} w^{ih}}.$$

**$N_1 =$  Number of copies of the packet that pass through the DR**

As soon as the DR gets one copy of the packet, it caches it. In case it sees other copies of the same packet, they are dropped and so need not be considered in our calculations. Thus  $N_1 = 1$

**$N_2 =$  Number of nacks the DR forwards**

For every transmitted copy of the packet that the DR sees a nack — it either needs to respond, or send a retransmission. Once it receives a copy of the packet, it caches it and responds to subsequent nacks. Furthermore, in the ideal case assumed, it forwards exactly *one* nack for every copy of the packet transmitted, until it gets some copy. So the number of nacks forwarded by the DR is exactly the number of drops experienced by the packet until it reaches the DR. Modeling the process required for the packet to reach the DR as a Markov Chain, we get

$$N_2 = \frac{i(1 - (1-p)^h)}{(1-p)^h}.$$

**$N_{3A} =$  Number of nacks seen by the DR until the packet reaches the DR**

Until the DR receives its first copy of the packet, *every* host below it do not have the packet either. Consequently, all the children hosts of the DR nack to it asking for the packet. All its children DRs too send one nack each for the packet. This goes on until the first copy is received (after  $N_2$  transmissions). Therefore,

$$N_{3A} = N_2 \cdot \frac{w^{h+1} - 1}{w - 1} = \frac{i(1 - (1-p)^h)}{(1-p)^h} \cdot \frac{w^{h+1} - 1}{w - 1}.$$

**$N_{3B} =$  Number of nacks seen by the DR after the packet reaches the DR**

We calculate this by considering a loss at level  $k$  downstream from the DR and evaluating its effects on the number of nacks generated. A loss at level  $k$  causes all the hosts below that point to nack, together with one nack each from all the DRs from that subtree.

$$\begin{aligned} N_{3B} &= \sum_{k=1}^h (\# \text{ links at level } k) \cdot (\text{lossprob at level } k \text{ link}) \cdot (\# \text{ nacks generated due to loss at level } k) \\ &\approx \frac{p}{w-1} \frac{w^{h+1} [h(1-p)(w-1) - 1]}{w(1-p) - 1} \quad \text{For } p \ll 1 \implies (1-p)^h \approx 1 - hp. \end{aligned}$$

**$N_4 =$  Number of packets the DR retransmits**

A DR has to keep retransmitting a packet until every host below it till the next level of DRs gets the packet. It suffices to calculate the number of transmissions required till the next level of DRs gets the packets, since this implies every host above it having got the packet too.

There are  $h$  levels till the next line of DRs, and the probability of losing any packet at any link is  $p$ . Thus,  $N_4 = \alpha(h, w, p)$ .

### 2.3 Calculating the non-Caching Average Bandwidth $\rho_{avg}^{NC}$

Similar to our calculation for  $\rho_{avg}^C$  (Section 2.2) we first calculate the bandwidth experienced at some DR (located  $i$  levels away from the source) and average this across all the DRs at all levels.

Let  $\rho^{NC}(i)$  denote the expected bandwidth utilized per packet at a DR located  $i$  levels away from the source. We calculate  $\rho^{NC}(i)$  as the sum of the same five quantities as Section 2.2. Thus  $\rho^{NC}(i) = N_1 + N_2 + N_{3A} + N_{3B} + N_4$ .

Again, averaging over all the levels of DRs,

$$\rho_{avg}^{NC} = \frac{\sum_{i=0}^{b-1} \rho^{NC}(i) \cdot w^{ih}}{\sum_{i=0}^{b-1} w^{ih}}.$$

**$N_1 =$  Number of copies of the packet that pass through the DR**

Since the DRs do not cache any packets, it is up to the host to keep retransmitting copies of the packets until *every* host in the multicast tree gets it. Furthermore, since all packets are multicast by the host, potentially every packet sent out is received by every host. There are  $bh$  layers in all and each link loses each packet with a probability  $p$ . Thus,

$$N_1 = \alpha(bh, w, p)(1-p)^{bh}.$$

**$N_2 =$  Number of nacks the DR forwards**

One nack is forwarded up by the DR for every copy of the packet transmitted, and this is carried on until every host below *this* DR receives the packet. Equivalently, we need to calculate how many transmissions are required till every one of the hosts in the last layer of the subtree rooted at this DR gets the packet.

Since all the packets sent do not actually reach the DR under consideration, we need to use a variation of the recursion scheme used in Section 2.1. Using the same notation, let  $n$  be this DR and let  $q$  be the probability that any packet sent by the host is lost on the path to DR  $n$ . Then  $q = 1 - (1-p)^{bh}$  and  $(b-i)h$  is the height of the subtree rooted at node  $n$ . Hence,

$$N_2 = E[T(n)] = \sum_{i=0}^{\infty} (1 - F_n(i))$$

$$F_n(i) = \sum_{u=0}^{i-1} \binom{i}{u} q^u (1-q)^{i-u} (F_c(i-u))^w.$$

The expression for  $F_c(i-u)$ , for a child node, is calculated as in Section 2.1.

**$N_{3A} =$  Number of nacks seen by the DR until the packet reaches the DR**

$N_{3A}$  for the non-caching case is identical to the  $N_{3A}$  term calculated for the caching case in Section 2.2.

$$N_{3A} = \frac{i(1 - (1-p)^h)}{(1-p)^h} \cdot \frac{w^{h+1} - 1}{w - 1}.$$

**$N_{3B} =$  Number of nacks seen by the DR after the packet reaches the DR**

Following the analysis for the  $N_{3B}$  for  $\rho^C$  (Section 2.2) we get the value for the number of nacks seen by the DR due to the hosts below it. However, in the non-Caching case, there may be nacks coming from below the next level of DRs due to packet losses further down. We incorporate the above phenomenon into our calculation by adding a term  $w^h p'$  where  $w^h$  is the number of children DRs and  $p'$  is the probability of seeing a nack due to a loss below the next level of DRs. Then,

$$\begin{aligned} N_{3B} &= \frac{p}{w-1} \frac{w^{h+1} [h(1-p)(w-1) - 1]}{w(1-p) - 1} + w^h p' \\ &= \frac{p}{w-1} \frac{w^{h+1} [h(1-p)(w-1) - 1]}{w(1-p) - 1} + w^h \left[ 1 - (1-p)^{\frac{w^{bh} - ih - h + 1 - 1}{w-1}} \right]. \end{aligned}$$

$N_4 = \text{Number of packets the DR retransmits}$

Since a non-Caching DR does not cache, it is incapable of retransmission. Hence,  $N_4 = 0$ .

### 3 Average Delay

#### 3.1 Calculating the average Caching Delay $\tau_{avg}^C$

For the caching case, the delay incurred in reaching a particular host consists of two parts — the delay incurred in reaching the DR for that host, and the the delay incurred while travelling from the DR to the host. So for a host located at height  $ih + k$ , we calculate the expected delay as the sum of two parts as calculated below:

$$E[\tau_{avg}^C(ih + k)] = E[\tau_i] + E[\tau_k].$$

$\tau_i = \text{the delay incurred in reaching the parent DR}$

While trying to calculate the delay, we let  $D$  denote the number of times the packet gets dropped. Each time the packet is dropped, retransmission starts at the last DR to have received it, but after waiting for a timeout  $T$ . Let there have been  $D$  drops on the way. Then,

$$E[\tau_i] = ih + \frac{i(1 - (1 - p)^h)}{(1 - p)^h} T. \quad (\text{Since } D = N_2 \text{ as calculated in Section 2.2})$$

$\tau_k = \text{the delay incurred in going from the parent DR to the host}$

When the packet is transmitted from the final DR to the host, there is no further caching involved, and the successful transmission must cross all the intermediate links. Again, each time there is a drop, a timeout worth of delay is incurred. Let  $d$  denote the number of drops and  $T$  the timeout. Probability that there is a drop  $= 1 - (1 - p)^k$ . This gives us

$$E[\tau_k] = E[E[\tau_k|D]] = \frac{2k + T}{(1 - p)^k} - k - T.$$

#### 3.2 Calculating the average non-Caching Delay $\tau_{avg}^{NC}$

$\tau_{1D}^{NC}(M)$  for One-Dimensional tree

Our assumption is of a one-dimensional multicast tree with every link having an independent loss probability of  $p$ . Whenever a packet is lost, a nack is generated by the highest host that is yet to get the packet. We are interested in the expected delay experienced by the host at level  $M$ , denoted by  $\tau_{1D}^{NC}(M)$  (the subscript ‘1D’ denotes one-dimensional).

Assuming a timeout  $T$  each time the packet gets dropped, and that there are  $D$  drops,

$$\begin{aligned} \tau_{1D}^{NC}(M) &\simeq DT + M + 2 \sum_{k=1}^D Tx_k \\ E[\tau_{1D}^{NC}(M)] &\simeq E[DT + M + 2E[(Tx_1 + \dots + Tx_D)|D]]. \end{aligned}$$

$Tx_k$  is the distance from which the nack is sent after  $k$  transmissions. This is the maximum height that is yet to receive the packet. So,  $E[Tx_k|D] \simeq \frac{1}{p^k} \ln(k)$ . Then,

$$E[\tau_{1D}^{NC}(M)] \approx TD^* + M + \frac{2}{p^F} \ln(D^*!).$$

where  $D^* = E[D] = \frac{1}{(1-p)^M} - 1$  and  $F = 1 - (1-p)^{M-1}$ .

### $\tau_{avg}^{NC}(M)$ for the General Tree

Using the same argument as in the one-dimensional case, and assuming  $D$  drops,

$$\tau_{avg}^{NC}(M) = DT + M + 2 \sum_{k=1}^D Tx_k.$$

However, in the case of the general tree, the  $Tx_j$ s depend on the exact location of the losses. This is because the retransmissions are triggered by the nack from the host closest to the source that is yet to get the packet, and so there is no simple relationship between the  $Tx_j$ s.

To solve the system, we can note that the system is a Markov process since the next state of the tree depends only on the current state. We can then apply existing algorithms to solve the system once it has been modeled. However, the state space for this problem is often unmanageably large, and we are working on an elegant solution to the Markov model.

### 3.3 Simulation Results

We used simulation studies of the system to verify the delay results obtained. In the simulation model, any packet was lost at any link with a loss probability  $p$ . In the caching model, each time a packet got lost, it was retransmitted by the last DR to have received it. For the non-caching model though, every retransmission began at the source. A counter kept track of the total number of steps taken by the packet to reach the destination. This value was averaged over 1000 packets to determine  $\tau^C$  and  $\tau^{NC}$ .

We plotted the value of the average delay in reaching the last level as the depth of the tree grew from 5 to 25, for link loss probabilities of 0.001, 0.01, 0.05 and 0.1. Timeout was chosen as a constant value of 10 for all the simulations. In the graphs plotted here (Figures 1 and 2), the distance between adjacent levels of DRs was chosen as 5 (i.e.  $h = 5$ ). Simulations using  $h = 4$ ,  $h = 6$  and  $h = 10$  also yielded similar results.

This is to be compared with the theoretical values plotted on the same graphs (Figures 1 and 2). A comparison shows that the simulated values mirror the calculated ones quite well, the difference being of the order of 15%, and remaining fairly constant across all loss probabilities. As can be seen from the graphs, the simulation results are always *more* than the theoretical results. In the theoretical calculations, the delay experienced by each unsuccessful transmission of the packet is considered only upto the last DR it reaches, while the packet may actually travel a few more steps before being dropped at an internal host. These extra steps are also taken into account in the simulation model, accounting for the excess delay.

Further comparison of the delay graphs across the caching and the non-caching case (comparing Figures 1 and 2) shows the delay in the Non-Caching case to be considerably more than the caching case. Furthermore, as the height of the tree and the link loss probability increases, the delay in the Caching scheme increases linearly, while the Non-Caching delay increases in an exponential manner. Thus, in poor network conditions, and for large multicast trees, it becomes imperative to use Caching DRs in order to ensure efficient and reliable performance.

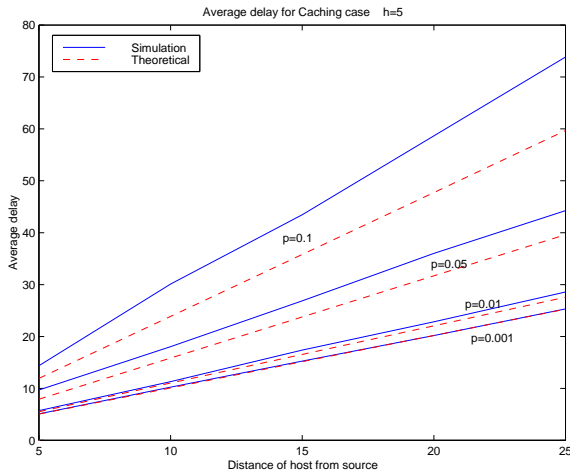


Figure 1: Caching Average Delay

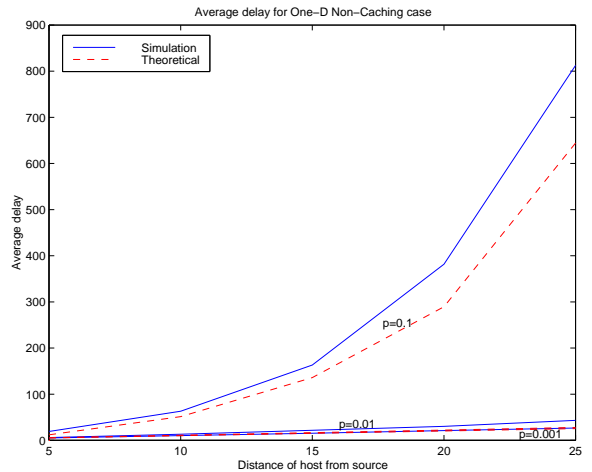


Figure 2: Non-Caching Average Delay

## 4 Conclusion

We have presented our analysis of the bandwidth and delay for a large class of Reliable Multicast protocols. We identified the class of protocols to analyze — a nack-based and tree-based reliability scheme which utilizes nack-accumulating and caching agents (called Designated Receivers) to handle retransmission.

Having modeled the multicast tree as an uniform tree with equal and independent link loss probabilities, we estimated the bandwidth utilized over a single link at a DR for transmitting a single packet to all the members of the multicast group. This expected value was then averaged over all possible DRs to yield the average bandwidth per packet per link. The average delay experienced by a host per packet was then evaluated for the same tree. The calculations were carried out for both the caching and the non-caching cases. Simulation results were also used to support the delay calculations.

The chief contribution of this paper is to handle a large class of Reliable Multicast protocols and analyze the cost involved in implementing such methods — in terms of the excess bandwidth used by the retransmission packets and the extra delay incurred by the hosts. This is of critical importance to reliable multicast applications which need to function under various limitations on bandwidth and delay. Given the bandwidth and delay constraints imposed by the application, we can use the results presented here to evaluate the utility of using this class of solutions for a multicast session requiring reliability. This analysis shows the trade-offs involved in ensuring reliability over multicast, and quantifies the “costs” that have to be incurred to achieve this.

The complete paper, including all the derivations for the results presented here, may be downloaded from <http://www.path.berkeley.edu/~guptar/Gupta2706.ps.gz>.

## References

- [BMT94] P. Bhagwat, P. P. Mishra, and S. K. Tripathi, "Effect of Topology on Performance of Reliable Multicast Communication", *Proceedings of INFOCOM'94*, vol. 2, pp. 602-609, Toronto, Ontario, Canada, June 1994.
- [DC90] S. Deering and D. Cheriton, "Multicast Routing in Datagrams Internetworks and Extended LANs," *ACM Transactions on Computer Systems (TOCS)*, vol. 8, no. 2, pp. 85-110, May, 1990.
- [D+97] B. DeCleene, S. Bhattacharya, T. Friedman, M. Keaton, J. Kurose, D. Rubenstein and D. Towsley, "Reliable Multicast Framework (RMF): A White Paper", available as <http://www.tascnets.com/mist/RMF/RMFWP.ps>.
- [F+95] S. Floyd, V. Jacobson, C. Liu, S. McCanne and L. Zhang, "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing, Scalable Reliable Multicast (SRM)", *ACM SIGCOMM '95*, available as [ftp://ftp.ee.lbl.gov/papers/srm\\_sigcomm.ps](ftp://ftp.ee.lbl.gov/papers/srm_sigcomm.ps).
- [Gro96] M. Grossglauser, "Optimal Deterministic Timeouts for Reliable Scalable Multicast", *Proc. IEEE INFOCOM '96*, San Francisco, California, March 1996.
- [LG97] B. N. Levine and J. J. Garcia-Luna-Aceves, "A Comparison of Reliable Multicast Protocols", *ACM Multimedia Systems*, 1998.
- [LG97a] B. N. Levine and J. J. Garcia-Luna-Aceves, "Improving Internet Multicast with Routing Labels", *IEEE International Conference on Network Protocols (ICNP-97)*, October, 1997.
- [LP96] J. C. Lin and S. Paul, "RMTP: A Reliable Multicast Transport Protocol", *IEEE INFOCOM '96*, March 1996, pp.1414-1424.
- [MJV96] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven Layered Multicast", *ACM SIGCOMM*, Stanford, CA, pp. 117-130, August 1996.
- [NB96] J. Nonnenmacher and E.W.Biersack, "Reliable Multicast: Where to use Forward Error Correction", *Proc. 5th Workshop on Protocols for High Speed Networks*, pp.134-148, Sophia Antolis, France, Oct.1996, available as <http://www/eurocom.fr/~nonnen/mypages/FECgain.ps.gz>.
- [PP98] C. Papadopoulos and G. Parulkar, "An Error Control Scheme for Large-Scale Multicast Applications", *Proc. IEEE Infocomm '98*, San Francisco, California, August 1998.
- [RV97] L.Rizzo and L.Vicisano, "A Reliable Multicast data Distribution Protocol based on software FEC techniques", *Proc. of The Fourth IEEE Workshop on the Architecture and Implementation of High Performance Communication Systems (HPCS'97)*, Sani Beach, Chalkidiki, Greece, June 23-25, 1997.
- [TK98] D.Towsley and J.Kurose, "A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols", to appear in *IEEE Journal on Selected Areas in Communications*.
- [YKT96] M.Yajnik, J.Kurose and D.Towsley, "Packet Loss Correlation in the Mbone Multicast Network", *IEEE Global Internet Conf.*, pp.94-99, London, Nov. 1996.