Mobile Reliable Multicast Support in IP Networks†

Chunhung Richard Lin and *Chang-Jai Chung*

Department of Computer Science and Information Engineering, National Chung Cheng University, TAIWAN
Email: chlin@acm.org

Abstract - We propose a reliable mobile multicast protocol, *RRBMoM* (Reliable Range Based Mobile Multicast), in IP networks using Mobile IP. RRBMoM is an extension of RBMoM [8]. It provides ACK-based reliability and sender-initialed loss recovery, and uses a tree-based hierarchical structure to alleviate the *acknowledgment-implosion* problem. A *multicast home agent* (MHA) acks as a representative of a set of mobile receivers which are roaming within its service range. Only the MHA's send acknowledgments to the sender, instead of all receivers sending their acknowledgments to the sender to prevent acknowledgment-implosion. In addition, the scheme of handoff loss recovery leads to improved throughput and end-to-end delay at the cost of additional backup nodes in the multicast tree. This cost can be dynamically controlled by adjusting a system parameter (i.e., *service range*). Performance results show that RRBMoM is efficient and adaptive to the dynamics of the system.

1. INTRODUCTION

Recent years have witnessed a tremendous increase in the use of the Internet for a large variety of applications including commerce, web access, software distribution, multimedia, and of course, data communication. Many of these applications require reliable data transmission from a sender to multiple receivers. Thus, reliable multicast is gaining popularity as a highly desirable feature of the future Internet. Providing mobile service to multicast protocols is made further difficult because many multicast protocols are inefficient when faced with frequent membership or location changes.

The basic idea of most reliable multicast protocols is to use the IP multicast infrastructure for routing, and add functionality at the end hosts, and possibly at the multicast routers, in order to support reliable multicast. Unfortunately, the provision of multicast services to mobile hosts proves to be a very challenging problem, for several reasons. First, even unicast routing for mobile hosts is a difficult problem, since the routing of datagrams to a mobile host changes whenever the mobile host changes location. Second, all existing multicast protocols (e.g., DVMRP, MOSPF, CBT, and PIM [12]) implicitly assume stationary hosts when configuring the multicast delivery tree. The IETF Mobile IP specification defines a method for routing packets to mobile hosts. It also defines two multicast support options, called remote subscription and bi-directional tunneled multicast. RBMoM [8] generalizes the above two protocols by using a parameter called *service range.* RBMoM intends to trade off between the shortest delivery path and the frequency of the multicast tree reconfiguration by controlling the service range. However, these three mobile multicast protocols do not provide reliability service.

Our goal is to provide an new protocol that guarantees reliable, sequenced delivery of multicast streams with support for mobility. Based on RBMoM, we provide the reliable service. In our protocol, RRBMoM, we use ACK-based reliability, hybrid sender-initialed and local server-initialed loss recovery, and end-to-end reliable delivery. In addition, we describe the RRBMoM protocol and address key design-related and implementation-related issues. The rest of the paper is organized as follows: Section II discusses the RRBMoM reliable mobile multicast protocol. Section III presents performance of RRBMoM. Finally, we conclude this paper in Section IV.

2. THE DESIGN OF RRBMoM

RRBMoM is a reliable multicast protocol used in an internetwork with mobile receivers. Since it is a reliable multicast protocol, it has to tackle the problem of loss detection and loss recovery. In RRBMoM, we use ACK-based mechanism to deal with loss detection and use the receiver-initialed retransmission to do with loss recovery. We describe this protocol detailedly by dividing the reliable multicast protocol into the following parts: (1) multicast tree management, (2) handoff scheme, (3) ACK mechanism, and (4) duplicate data request.

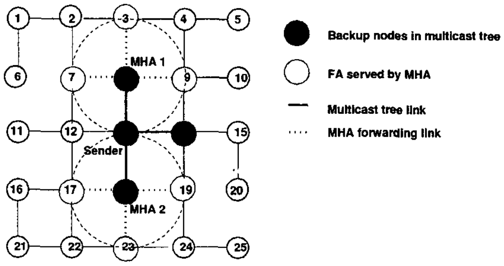

Figure 1: The service range concept

2.1. Multicast Tree Management

RBMoM [8] provides a mobile multicast solution, which uses a parameter called *service range* to generalize two mobile multicast approaches, *remote subscription* and *bi-directional tunneling*, proposed by Mobile IP. In order to provide the reliable multicast service in mobile networks, we will base on RBMoM for two reasons. First, it was designed to support multicast with mobility. That is, it can dynamically and efficiently maintain the multicast delivery tree according to the movement of mobile hosts. Second, for reliability, we need to have backup in some tree nodes for local recovery. How can we choose those nodes? RBMoM provides an easy way to deal with this problem.

---

† This work was supported by the National Science Council, TAIWAN, under Contract NSC-89-2213-E-194-007 "Design of Multicast Support for Mobile Networks".

0-7803-6283-7/00/$10.00 © 2000 IEEE

In RBMoM, it uses the idea of "multicast home agent" (MHA) and "service range". MHA is a multicast router which is in the multicast tree. Consider the example in Figure 1. If the service range of MHA is one (hop distance), the MHA_1 only forwards multicast datagrams from the sender to the mobile receivers which are roaming in the foreign networks 3, 7, and 9; Similarly, MHA_2 only datagrams to the foreign networks 17, 19 and 23. Simply by changing the service range, we can control the size of the multicast delivery tree. For example, when the service range is small, the size of the multicast tree is inclined to increase because each MHA serves fewer networks and more MHA's will be needed for all networks which contain mobile subscribers. In addition, a small service range will increase the overhead of the multicast tree maintenance. This is because MHA's will join or leave the multicast tree more frequently when subscribers are roaming. However, a small service range can provide a more optimal delivery path.

Our protocol is based on RBMoM for reliability support. Mobile receivers are grouped into local regions or domains and each domain is defined by the service range of each MHA. MHA is a special nodes in the multicast tree which is responsible for sending acknowledgments periodically to the sender, for processing acknowledgment from foreign agents (FA's) in its domain, and for retransmitting lost packets to the corresponding receivers. We temporarily backup the multicast datagrams in MHA. In order to reduce the backup nodes to save the cost, we can increase the service range of MHA. However, there will be more unicast transmission from MHA to the corresponding mobile receivers. This will result in the same problems of implementing multicast by multiple unicast (e.g., waste of bandwidth). Another drawback of large service range is that when the unicast datagram request was issued by a mobile receiver after handoff, it will take longer time to reply datagram to the mobile receiver (longer delivery path).

2.2. Handoff Scheme

Multicast datagrams may be lost in the duration of handoff. In order to support reliability, we need to redirect multicast data stream to the new foreign network after handoff, and more important we must perform the loss recovery after handoff.

On arriving at a foreign network, a mobile host (MH) locates the foreign agent (FA) and registers with it according to Mobile IP. Besides, it also reports the sequence number status to FA. If there has already existed at least one mobile participant of this multicast session in the foreign network, the FA forwards the sequence number status to its MHA. Then the MHA forwards it to the root of the multicast tree to enable the root to monitor the status of receivers. Otherwise, the FA directly forwards the status message to the root. Figure 2 shows the status message to be passed when handoff occurs. This information can be used to detect the data loss. Therefore, we can treat the status message as a repair request. On the way to the root, this status message may be received by a tree node which detects a loss and has a copy of the missing data. If this is the case, the tree node immediately sends the data to the mobile host and inform the root of this data delivery. Also, the sequence number continues being forwarded by the tree node to the root. In our solution, in order to move processing away from the sender, the effort of processing requests and sending retransmissions is shared among all tree nodes. Therefore, because missing data can be recovered by retransmissions from the tree nodes as opposed to retransmissions from the original sender, recovery latency is significantly reduced (because of the shorter delivery path), and the overall throughput is improved as well. Also, the amount of bandwidth consumed by retransmission thus can be reduced.

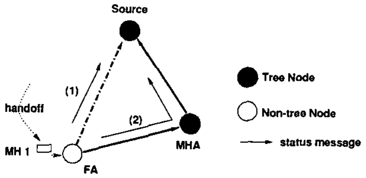

Figure 2: Status message passing when handoff occurs

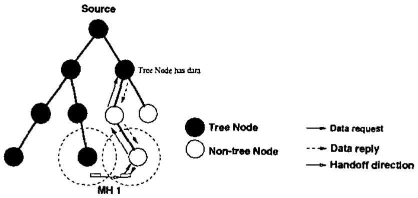

Figure 3: Loss recovery in handoff scheme

Consider the example shown in Figure 3. Once reaching a new foreign network, MH_1 will register with the new FA and report the sequence number status to it. In order to recover the lost packets in the duration of handoff, the FA will send a repair request (containing the sequence number) to the root in the shortest path. On the way to the root, the first tree node which gets this request replies the request by delivering the missing data to MH_1. If the FA is not served by any MHA, according to of the RBMoM protocol, it has to join the multicast group. Before FA completes the join operation and becomes a multicast group member, the tree node must continue forwarding multicast datagrams to the FA (FA can make this request to this tree node). Thus, there will not be any data loss for the mobile receiver after handoff.

2.3. ACK Mechanism

The ACK mechanism is also an important issue in a reliable multicast protocol. For simplicity, we assume that there is no transmission error in wired links. The packet loss in the wired backbone is mainly because of the network congestion. The ACK message in our protocol is used to enable the sender to monitor the status of receivers. That is, ACK message reports the information of which mobile hosts have received the data.

Our protocol has been designed to alleviate the *ack-implosion* problem by using a tree-based hierarchical approach. The key idea is to group receivers into local regions and to use a MHA as a representative of the local region. Because every MHA is in the multicast tree, only the MHA's send their own status to their parents in the tree indicating which packets they have received and which packets they have not received. After collecting all downstream nodes status messages, a tree node processes these messages and then sends its own status message to its parent. Therefore, in the status message, a received packet means that all downstream nodes have received it. The FA's of the receivers in a local region send their status to the corresponding MHA. Note that a MHA does not consolidate status messages of the FA's in its local region, but uses these status messages to perform local retransmissions to the FA's, reducing end-to-end delay significantly. Thus the sender sees only the MHA's and a MHA sees only the FA's in its local region. Processing of status messages is distributed among the sender and the MHA's, thereby avoiding the ack-implosion problem.

In our protocol, each node in the multicast tree has to wait for three kinds of ACK messages. The first is from its children in the multicast tree, the second is from the MH's which are roaming in its subnetwork, and the third is from the handoff MH's. The first is necessary because a tree node has to confirm that all downstream nodes have received the multicast datagram. The second is required because a tree node has to know if all mobile hosts roaming in its subnetwork have also received the multicast datagram. However, the third is not always necessary. Only when a tree node is responsible for forwarding datagram to the handoff MH's as what we have mentioned, it needs to wait for this kind of ACK message.

We backup multicast datagrams in some nodes, called *backup nodes*, of the multicast tree for loss recovery (e.g. MHA). To reduce the backup cost, it is important that when we can remove a packet from a backup node. According to the ACK mechanism, a backup node cannot remove a packet until receiving two ACK's, one from its children and the other from the mobile hosts roaming in its subnetwork. If a backup node is serving a set of mobile hosts for handoff loss recovery, the ACK message from the handoff MH's is also necessary.

2.4. Data Recovery When DMSP Handoff

A DMSP handoff means that a FA re-selects its DMSP. A DMSP handoff may occur in two situations. One is that a new mobile host enters a network and its HA is more suitable to be the DMSP; the other is that all DMSPs' mobile hosts move away from the network. When a mobile host handoff occurs, its HA can learn the mobile host's new FA immediately by using RBMoM. But the old FA can not know the handoff until timeout. So, before the new DMSP is selected, none will serve the mobile hosts in the previous network. Multicast packets for mobile hosts will be lost during this period.

In RRBMoM, we slightly modify the DMSP selection in RBMoM. We use a three-way method to provide a reliable multicast service. At first, when a DMSP finds all its mobile hosts move away from the network. It will still continue forwarding data to the FA. The FA then triggers a DMSP selection after a period of time. When the FA receives a multicast packet from the new DMSP, it immediately issues a message to tell the old DMSP to stop forwarding datagrams to it. So, before the new DMSP is selected, the old DMSP will serve mobile hosts in the previous network.

2.5. Duplicate Repair Request

Duplicate repair request is also a problem in our reliable multicast protocol. This problem results from the fact that multiple mobile hosts enter into a foreign network after handoff and issue status messages to the root. The FA is responsible for forwarding these messages to the root. On the way to the root, these messages may be received by a backup node which still backups the lost packets. Then it will create multiple loss recovery tunnels (one for each status message) which terminate at the FA. The FA, thus, may receive duplicate lost packets. For example, MH_1 and MH_2 arrive at the same foreign network and issue their own status for loss recovery request to the FA. The FA checks the sequence number, and determines if it has to issue this for loss recovery to avoid to get duplicate copies of the lost data. The decision principle in the FA is described as follows. The FA must record the minimum sequence number, say N, of the mobile receivers in the foreign network. When receiving a new status message, the FA compares the sequence number with N. If the sequence number is lager than N, it need not issue a loss recovery request to the sender. This is because the FA has already sent a recovery request and is waiting for the missing packets for the new handoff MH. However, if the sequence number is less than N, then FA must issue the recovery request to get the lost packets for the new MH.

3. PERFORMANCE EVALUATION

In our simulation, the network topology is randomly constructed at each time. Each multicast router also acts as a base station. Our topologies are based on 8×8 and 10×10 mesh networks. We use the Prim's algorithm to further determine the connectivity among multicast routers as was to be described in [8]. The distance between two nearby base stations is 100 meters and the power range is 75 meters.

In the beginning, we randomly place mobile hosts among the local networks. For simplicity, we assume there is only one multicast group in our simulation. The amount of the mobile participants of a multicast session may be 1, 20, 30, 60, or 120. Because wireless and wired channels have quite different bandwidth inherently, two system ticks are needed. We let the ratio of wireless tick to wired tick be 10. Here, we define a time unit to be a wireless tick which is the length of time to transmit a data packet through a wireless link. The data packet size is 4K bits. The bit error rate in the wireless links is from 50K to 10M (i.e., 1 error every x bits, on average). The mobile hosts in our simulation move uniformly in any direction at each time tick.

In our simulation, we randomly start to transmit data packets at any time. When a data packet is transmitted by the sender, it has to wait for the ACK to make sure if all subscribers have received the sent packet. The end-to-end delay of a packet means the amount of time units elapsed from the time of sending the packet to the time of receiving all ACK's. We randomly transmit 1 to 10 packets in each topology, and get the simulation results by averaging the results of 1000 different topologies. We do several experiments to assess the functionality and features of our protocol.

The first experiment is to evaluate the average end-to-end delay with different values of bit error rate (BER). The transmission error only occurs in the wireless links between the base station and the mobile hosts. The service range in this experiment is equal to one and the mobility of mobile hosts is 0.1. Figure 4 shows the result of our simulation. There are four curves for different numbers of multicast members ($m = 1, 30, 60, 120$). The curve of "$m = 1$" means the unicast. Observe that the average end-to-end delay increases as BER

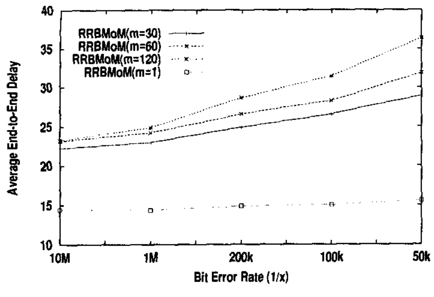

Figure 4: End-to-End Delay

increases. This is because there are more retransmissions to be needed for low channel quality. More mobile participants can further increase the probability of retransmission. Thus, the end-to-end transmission latency is increasing with BER. As was to be expected, in high channel quality, the end-to-end delay is not much affected by the amount of multicast participants. Observe that when BER = 10M, the delay is about 23 time ticks for the cases of 30, 60, and 120 participants. This is because of few retransmissions cost. Actually, for the same BER, we can find that the amount of multicast participants does not apparently affect the end-to-end delay. Moble hosts receive multicast datagram from its base station by broadcasting. A base station needs to re-broadcast data even if there is only one mobile host not getting the data. Thus, the transmission delay in a wireless network is affected by the number of mobile hosts roaming in the network. The end-to-end delay should be much affected by the network with more mobile hosts instead of the total number of mobile subscribers. In the case of low wireless channel quality, the delivery latency in the wired backbone is not so important to the end-to-end delay. In the simulation, we place the mobile hosts randomly in the networks and the topology changes from time to time. Therefore, the network with the maximal number of the MH's is also randomized. From the point of view, we can realize why the total number of subscribers does not affect the end-to-end delay so obviously.

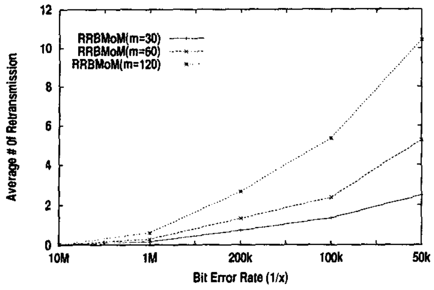

Figure 5: Number of retransmission

Figure 5 shows the result of the average number of retransmissions per data packet over the wireless links. As was to be expected, high BER causes more data retransmissions. In addition, the total number of mobile subscribers almost linearly affects the number of retransmissions.

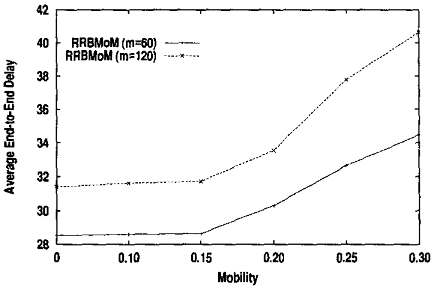

Figure 6: End-to-End Delay

In Figure 6, 7, and 8, we do some experiments for different mobility. The BER in these experiments is uniformly set to 100K. The value of service range is 1. and the size of the mesh network is 64 (8 × 8). In Figure 6, when the mobility increase, the end-to-end delay also increase. This is because high mobility increases the handoff rate. Therefore, MH's take longer time to receive the data because of the overhead of the handoff loss recovery. Observe that the curves for the two different numbers of subscribers (m = 60, 120) are very similar. Also, the difference between two curves is almost the same even when the mobility goes up. When the number of mobile subscribers is high, the size of multicast delivery tree is larger according to the RBMoM protocol. Therefore, the backup nodes will also increase and thus the time of retransmission is reduced. This results in the fact that the number of subscribers does not seriously affect the end-to-end delay at the same mobility (e.g., at mobility 0.2, end-to-end delays for m = 60 and 120 are 30 and 33 time ticks, respectively). But what are the overheads when more subscribers are in the system? Figure 7 and 8 show us the answer.

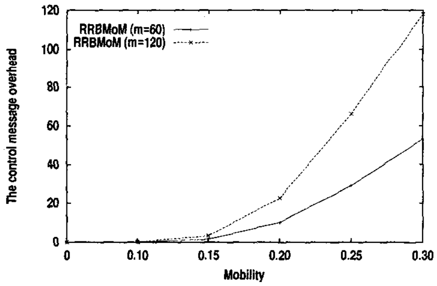

Figure 7: Control message overhead during handoff

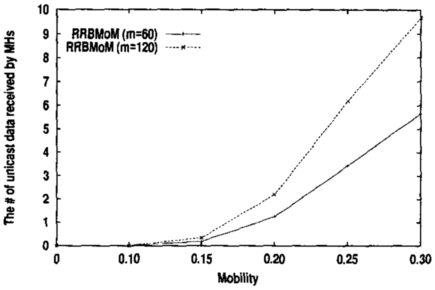

Figure 8: Number of MH's that receive unicast data after handoff

Figure 7 shows the total number of status messages for handoff loss recovery. The status message contains the information of the sender ID, sequence number, and the flag for if retransmission is needed. From the result, the number of subscribers "does" affect the amount status messages more and more obviously when the mobility goes up. This is because more subscribers in high mobility will increase the total number of handoff more quickly. Thus, the amount of status messages is quickly increased for loss recovery. Similarly, Figure 8 is like the Figure 7. In this experiment, we count how many receivers which get the datagram by unicasting instead of multicasting for each multicast datagram sent from the root. That is, the received datagram is from the backup node in the multicast tree by unicasting. This situation only occurs in the duration of handoff. Thus, the number of MH's which receive data from unicast is directly decided by the handoff rate. We can find that high mobility and a larger amount of subscribers will result in more handoff loss recovery operations to be performed.

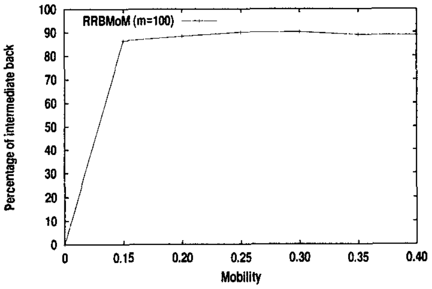

Figure 9: Percentage of intermediate back

In our protocol, since lost packets can be recovered by retransmissions from the tree nodes as opposed to retransmissions from the original sender, end-to-end latency is significantly reduced (because of the shorter delivery path), and the overall throughput is improved as well. Also, the amount of bandwidth consumed by retransmission thus can be reduced. Figure 9 shows the percentage of handoff retransmission requests which are replied by the backup node, instead of the root. From the figure, we can find that about 90% requests are replied by the tree nodes.

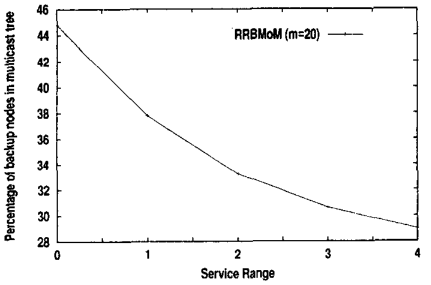

Figure 10: Percentage of backup nodes in the multicast tree

Finally, we compare the percentage of backup nodes in the multicast tree for different service ranges. In large service range, each MHA needs to serve more foreign networks. Therefore, for a given topology, less number of MHA's are needed to serve all networks. MHA's are backup nodes in the system for loss recovery. That is, we need less backup nodes for large service range as shown in Figure 10. To control the system backup cost, we can simply adjust the value of service range.

4. CONCLUSIONS

In this paper, we propose a new reliable mobile multicast protocol, RRBMoM. The RBMoM mobile multicast infrastructure provides an easy way to develop the reliability service. The use of MHA can group mobile participants into local regions or domains. The size of a domain is controlled by the service range of MHA. A MHA acts as a representative, which is responsible for sending acknowledgments periodically to the original sender, for processing acknowledgment from foreign agents (FA's) in its domain, and for retransmitting lost packets to the corresponding receivers. Since lost packets can be recovered by retransmissions from MHA's as opposed to retransmissions from the original sender, end-to-end latency is significantly reduced, and the overall throughput is improved as well. Also, the amount of bandwidth consumed by retransmission thus can be reduced. We temporarily backup the multicast datagrams in MHA's. Thus, a MHA can response the handoff loss recovery request from a distant FA, and retransmit the lost packets. To control the backup nodes in the system, we can simply adjust the service range. The use of a tree-based hierarchical ACK scheme can alleviate the acknowledgment-implosion problem. Simulation results demonstrate the performance characteristics of our protocol.

REFERENCES

[1] R. Aiello, E. Pagani, and G.P. Rossi, "Design of a reliable multicast protocol", *Proceedings of IEEE INFOCOM '93,* Mar. 1992, pp. 75-81.

[2] A. Ballardie, P. Francis, and J. Crowcroft, "Core Based Trees: An Architecture for Scalable Inter-Domain Multicast Routing", *Proceedings of ACM SIGCOMM '93,* August 1993.

[3] A.M. Costello and S.McCanne, "Search Party: Using Randomcast for Reliable Multicast with Local Recovery", *Proceedings of IEEE INFOCOM '99,* March 1999.

[4] S. Floyd, V. Jacobson, C.-G. Liu, S. McCanne, and L. Zhang, "A Reliable Multicast Framework for Light Weight Sessions and Application Framing,", *IEEE/ACM Transactions on Networking,* November 1996.

[5] T.G. Harrison, C.L. Williamson, W.L. Mackrell, and R.B. Bunt, "Mobile Multicast (MoM) Protocol: Multicast Support for Mobile Hosts", *Proceedings of ACM/IEEE MOBICOM '97,* September 1997.

[6] H. Holbrook, S. Singhal, and D. Cheriton, "Log-Based Receiver-Reliable Multicast for Distributed Interactive Simulation", *Proceeding of ACM SIGCOMM '95,* August, 1995.

[7] B.N. Levine and J.J. Garcia-Luna-Aceves, "A Comparison of Reliable Multicast Protocols", *Multimedia Systems (ACM/Springer),* August 1998.

[8] C.R. Lin, and K.-M. Wang, "Mobile Multicast Support in IP Networks", *Proceedings of IEEE INFOCOM 2000.*

[9] J.C.-H. Lin, S. Paul, K.K. Sabnani, and Bhattacharyya, "Reliable Multicast Transport Protocol (RMTP)", *IEEE Journal on Selected Areas in Communications,* April 1997, pp. 407-421.

[10] S. Pingali, D. Towsley, and J. Kurose, "A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols", *IEEE Journal on Selected Areas in Communications,* April 1991.

[11] B. Rajagopalan, "Reliability and scaling issues in multicast communication", *Proceedings of ACM SIGCOMM '92,* Sept. 1992, pp. 188-198.

[12] W. Stallings, *High-speed Networks: TCP/IP and ATM Design Principles,* Prentice-Hall, 1998.