Reliable Multicast in Ad Hoc Wireless Networks

Ching-Chuan Chiang
Department of Computer Science
Chung Cheng Institute of Technology
ccchiang@ccit.edu.tw

Jyh-Jean Wu
Department of Electronics Engineering
Chung Cheng Institute of Technology
g891302@ccit.edu.tw

Keywords

Reliable Multicast, SRM, FGMP, Ad Hoc Networks, Wireless Networks

Abstract

From applications involving human collaboration to critical network services like autonomous management and security, many target applications envisioned for multihop wireless networks[3] benefit from a network multicast. However, most network multicast services of today offer some form of "best-effort" delivery which lacks any end-to-end recovery mechanisms. While many ad hoc mechanisms have been developed over the years for adding end-to-end recovery to unreliable delivery, multicast poses several unique challenges related to both the size of the network and the number of multicast sessions and session participants; this problem deserves special attention in wireless networks with relatively scarce transmission resources. For these reasons, we conducted simulations with one such reliable multicast mechanism. Specifically, Scalable Reliable Multicast (SRM)[14] provides a framework for developing applications which both benefit from network multicast and incorporate end-to-end recovery mechanisms. In this paper, we have built upon the existing multicast simulation infrastructure, which is using FGMP protocol[3], to better understand the overall suitability of SRM in wireless environments. A unicast patching scheme is proposed to reduce the broadcast overhead of SRM and thus improves the performance.

1. Introduction

Wireline network multicast routing protocols (e.g., DVMRP[13], PIM[12], CBT[7,15], etc.) are based on the use of distribution trees for efficient delivery of multicast packets. In ad hoc wireless, mobile networks[2,10], however, the validity of tree structures is undermined by the broadcast nature of the channel and the continuously changing network connectivity. The use of trees in a rapidly reconfiguring environment requires frequent repairs of branches, and has two negative consequences: high channel and processor overhead, and high risk of packet loss during branch reconfiguration. To overcome tree topology limitations, we use the concept of "Forwarding Group", a set of nodes which are responsible for forwarding multicast data. The Forwarding Group infrastructure reduces storage overhead and requires a much looser connectivity among multicast members. It suffices that the mesh topology formed by multicast members and forwarding group nodes be connected (no islands). The reduction of channel and storage overhead and the relaxed connectivity make this protocol more scalable for large networks and more stable for mobile wireless networks. The Forwarding Group multicast protocol was first introduced in [3].

Adding reliability mechanisms to multicast poses several problems. First and foremost, an effective reliable multicast scheme must handle scaling of control traffic overhead in the face of one-to-many relationships. In order to support Internet size sessions, control mechanisms must not adversely scale with the size of the session. In the case of loss-recovery-based reliability, this design goal applies specially to the number of receivers. Considering TCP illustrates this idea. In the TCP protocol, receivers send explicit acknowledgments to the sender for each segment of data received. In the multicast domain, such a scheme scales poorly with the number of receivers. Sometimes generally referred to as acknowledgment "implosion", sessions with a large number of receivers each sending acknowledgments results in a clog of acknowledgments at the sender. Much as a data flow diverges in the network subsequent to initial transmission by the sender, effective reliable multicast which supports a large number of receivers must aggregate recovery initiated by those receivers.

In this paper, we have built upon the existing multicast simulation infrastructure, which uses FGMP protocol [3], to better understand the overall suitability of SRM in wireless environments. A revised version of SRM is proposed, which is using unicast patching scheme, and thus reduces the broadcast overhead and improves the performance.

Section 2 reviews the forwarding group multicast protocol (FGMP) and Scalable Reliable Multicast (SRM) protocol. Section 3 describes the FGMP+SRM protocol in details. Section 4 introduces the revised SRM using unicast patching scheme. Section 5 addresses the network infrastructure and simulation environment. Section 6 details the performance evaluation. Section 7 concludes the paper.

2. FGMP and SRM

2.1 Forwarding Group Multicast Protocol (FGMP)

The FGMP scheme (first introduced in [3]) is reviewed here for completeness. FGMP keeps track not of links but of groups of nodes which participate in multicast packet forwarding. Each multicast group, *G*, is associated a forwarding group, *FG*. Any node in *FG* is in charge of forwarding multicast packets of *G*. That is, when a forwarding node (a node in *FG*) receives a multicast packet, it will broadcast this packet if it is not a duplicate. All neighbors can hear it, but only neighbors that are in *FG* will first determine if it is a duplicate (based on a historical list of source sequence numbers) and then broadcast it in turn. Figure 1 shows an example of a multicast group containing three senders and three receivers. Three forwarding nodes take the responsibility to forward multicast packets. This scheme can be viewed as "limited scope" flooding. That is, flooding is contained within a properly selected forwarding set.

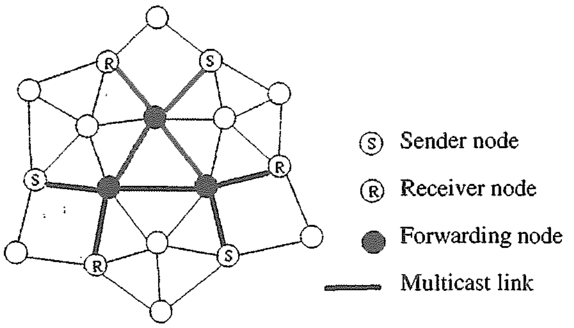


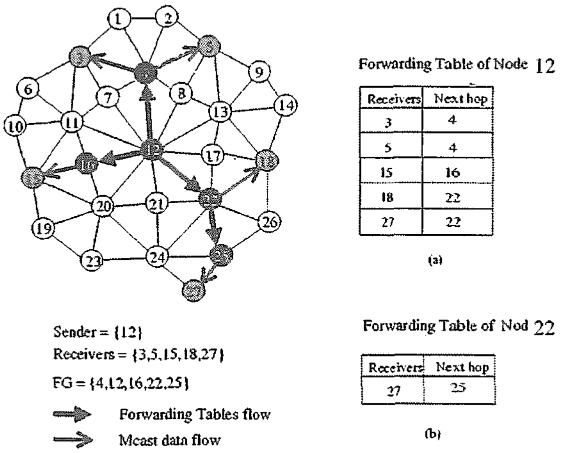
Figure 1. An example of FGMP

Only flag timer and historical source sequence number list are needed for each forwarding node. When the forwarding flag is set (as described in following subsections), each node in *FG* forwards non-duplicate data packets belonging to *G* until the timer expires. Storage overhead, a major problem in traditional multicast protocols, is minimal, thus improving the scalability. Timer is refreshed by the forwarding group updating protocol. Stale forwarding nodes are deleted from *FG* after timeout.

A key component of FGMP is the election and maintenance of the set *FG* of forwarding nodes. The size of *FG* should be as small as possible to reduce wireless channel overhead. Yet, the forwarding path from senders to receivers should be as short as possible to achieve high throughput.

FG **Maintenance**: In order to select the set *FG*, we require each source to periodically transmit control packets to all member destinations. In the process, all nodes along the shortest path from source to destination are "included" into *FG*. This procedure presumes that each source knows the member destinations. This is obtained via receiver advertising. Namely, each **receiver** periodically and globally floods its member information (join request) formatted as in table 1. TTL limits the scope of flooding. Each **sender** maintains a member table as shown in table 2. When a sender receives the join request from receiver members, it updates its member table.

Expired receiver entries will be deleted from the member table. Non-sender nodes simply forward the request packet. After updating the member table, the sender creates from it the forwarding table *FW* shown in table 3. Next hop on the shortest path to the receiver is obtained from preexisting routing tables. The forwarding table *FW* is broadcast by the sender to all neighbors; only neighbors listed in the next hop list (next hop neighbors) accept this forwarding table (although all neighbors can hear it). Each neighbor in the next hop list creates its forwarding table by extracting the entries where it is the next hop neighbor and again using the preexisting routing table to find the next hops, etc. After the *FW* table is built, it is then broadcast again to neighbors and so on, until all receivers are reached. Note that *FW* is discarded after use. The member table on the other hand is permanent. The forwarding table *FW* propagation mechanism essentially "activates" all the nodes on the source tree rooted at the sender. These nodes become part of the *FG*. At each step, nodes on the next hop neighbor list after receiving the forwarding table enable the forwarding flag and refresh the forwarding timer. Soft state dynamic reconfiguration[3] provides the ability to adapt to a changing topology.



Forwarding Table of Node 12

Receivers	Next hop
3	4
5	4
15	16
18	22
27	22

(a)

Forwarding Table of Nod 22

Receivers	Next hop
27	25

(b)

Figure 2. Example of Forwarding table for FGMP-RA

Figure 2 shows an example of multicasting forwarding tables. Node 12 is the sender. Five nodes are forwarding nodes, *FG = {4,12,16,22,25}*, because they are in the next hop list. Only sender and internal nodes, in our case 12 and node 22, need to create a forwarding table (figure 2(a),(b)) and broadcast it. Forwarding nodes 4, 16, and 25 do not need to create their forwarding tables since they are "leaves".

Another way to advertise the membership is to let the senders flood sender information. Sender advertising is more efficient than receiver advertising if the number of

Mcast Group id	Receiver member id	Sequence #	TTL

Table 1. Format of join_request packet

Mcast Group id	
Refresh Timer	
Receiver member id	timer
...	...

Table 2. Format of member table at the sender members

Mcast Group id	
Receiver member id	Next hop
...	...

Table 3. Format of forwarding table *FW*

senders is less than the number of receivers. Most multicast applications belong to this category. Like in receiver advertising, senders periodically flood the sender information. Receivers will collect senders' status, then periodically broadcast "joining tables" to create and maintain the forwarding group *FG*.

Member table and forwarding table size pose a scaling limitation when the multicast group grows to hundreds or even thousands of nodes. A possible solution (which we are currently exploring) is to dynamically (and randomly) elect a small set of "core" nodes which lie on the path between senders and receivers. These core nodes advertise (at a fairly low frequency) their presence, i.e., their ID, to all nodes. Senders and receivers alike send short join messages to each of the core nodes, activating the *FG* flag in all the nodes encountered along the path. The scheme scales well in both storage and channel overhead. It does not however guarantee shortest paths between all senders and receivers. It also introduces the additional complexity of core node elections. We are evaluating some of these tradeoff in our current research.

2.2 Scalable Reliable Multicast (SRM)

SRM, as described in[14], is a framework for building multicast applications which incorporate reliability. In recent years, many proposals have emerged for supporting reliable multicast. The key contribution of SRM lies in its overall philosophy of supporting a minimal form of reliability. While, like most reliability mechanisms, SRM's notion of reliability includes the recovery of data lost during transit, it does not include guarantees about the ordering of packets commonly present in other mechanisms. Instead of providing a reliable ordered bit-stream like TCP, SRM places the requirement for providing unique names and ordering for data-units upon the application. In this way, SRM applies the principles for Application Level Framing (ALF)[5] to the reliable multicast.

While SRM differs from TCP in many ways, it remains similar in one respect. Much like TCP, the designers of the SRM framework took an end-to-end approach to control. In other words, recovery from losses occurs somewhere above the network-level. This allows SRM recovery to operate over a potentially large internetwork in which intermediate remain ignorant of SRM operations. This presents a contrast to some other mechanisms, for example, those presented in[6], which introduce reliability at or below the network-level.

In addition to its lightweight notion of reliability and end-to-end control, SRM differs from other reliability mechanisms in several other ways. Typically, reliability schemes include a protocol, i.e. a set of well-defined message formats and an accompanying state machine describing how to process these messages. Since SRM applies ALF principles, these don't explicitly exist in the SRM framework. Instead, the SRM framework consists of two main components. (1) The notion of session messages, and (2) the loss recovery algorithm. Subsequent sections describe both of these.

2.2.1 Session Messages

Each participant in a multicast session periodically transmits session messages to the entire session. These messages allow the other participants to form a shared-state describing the session. Some values of interest include the sequence numbers of ordered data for the purpose of detecting loss as well as the topological distances used by the loss recovery algorithm described in the next section. However, the exact nature of the session message will depend upon the requirements of the application, it's methods of detecting loss, and it's method of deriving a distance metric to weight the timers of loss recover.

2.2.2 Loss Recovery Algorithm

SRM's loss recovery works in the following way. Upon detecting a loss, a receiving participant in the session schedules a request for repair (patch request) by setting a timer over a uniform interval weighted by the perceived distance to the source. This distance could be measured in a number of ways, but the white-board application described in[14] uses a synchronized time space maintained by the session messages. Upon sending the request, the receiver resets and multiplicatively increases this request-timer. The patch request is flooded to the network so that other participants in the session can receive it. Upon hearing this request, a participant in the multicast session who previously received the requested data schedules a response by setting a timer over a uniform interval weighted by the distance to the requesting receiver. By weighting the response in this way, participants closer to the requesting receiver will generally respond before those farther away, keeping the repair localized about the source of the request. If a responding participant does not hear a response from anyone else by the time its timer expires, it transmits its repair (patch) to the entire session (flooding).

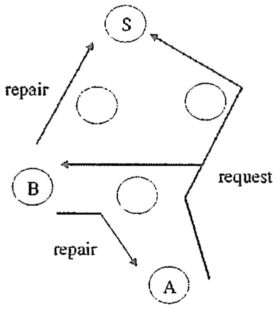


Figure 3. SRM Loss Recovery: upon hearing B's response to A's request, S does not respond.

On the other hand, if a responding participant hears a repair from some other participant while waiting to transmit its response, as in figure 3, it cancels its own response and does not send any repairs for the loss. Finally, if the receiver originally requesting a repair fails to recover by the time its request timer expires, it sends another request and again sets its request timer with a multiplicative back-off.

3. FGMP + SRM

To explore the reliable multicast and analyze SRM, we implement SRM into FGMP. Each multicast packet is marked with a sequence number and the sender ID so that receivers can detect the loss. A end-of-session message, which contains the sequence number of the last packet, needs to be sent to all receivers in order to recover lost packets due to the loss of the last packet. For the receiver advertising FGMP (FGMP-RA) scheme, the sender can send a reliable unicast end-of-session message to all receiver members since the receiver membership is stored in the sender's member table. For sender advertising FGMP (FGMP-SA), the end-of-session message can be piggyback on multicast packets or can be sent to all receivers by flooding. Packet loss is detected by the receiver and a timer is set for a patch request for the lost packet. Instead of using a gap which may consist of variable number of packets, we use a packet by packet recovery scheme. That is, each lost packet is set with a timer for the patch request and will be recovered by a patch.

Patch requests and patches are flooded with TTL specified to the entire network. Rather than recovering the whole gap a packet based recovery is to reduce the dominance and unfairness of wireless channel usage. Patch recovery timers are set according to the perceived delay which can be measured from the patch request.

4. Unicast patching for FGMP-RA

The SRM uses flooding algorithm for patch request and patches. This is necessary when the receiver does not have other members' information. However, flooding creates much overhead (especially for patches) and thus increases the latency of recovery. For FGMP-RA, the receiver advertises its information and this advertising message can be used to track multicast membership. Namely, each receiver keeps other members' information such as member IDs, distance, and delay. These information can be used to improve loss recovery. For example, based on the distance to members, a receiver can issue a patch request with proper TTL in order to avoid a large scope of flooding. Another potential is unicasting the patch requests and patches. We use the unicast patching here to improve the performance. The advantages of using unicast patching are: (a) unicast patching reduces the channel overhead incurred by flooding; and (b) upon receiving the patch request, the member which has received the requested packet can respond to the request immediately (without setting a timer since there is no possibility that other members might respond as well and need suppression to avoid duplicate responses), thus reducing the patch delay. The patch request is sent to a candidate member which is elected on the distance metric. If the candidate member cannot repair the request, it forwards the request to another member. The patch request carries a list of traversed members to avoid oscillation. Performance results are reported in the following section.

5. Network Infrastructure and Simulation Environment

The infrastructure used in our experiments is a clustered multihop Infrastructure[2,10]. In our distributed clustering Algorithm[2], nodes are elected as clusterheads based on preferential criteria (e.g., lowest ID number, etc.). Neighbors are discovered with periodic Hello messages. All nodes within transmission range of a clusterhead belong to the same cluster, and can communicate directly with a clusterhead and (possibly) with each other. Nodes belonging to more than one cluster are called gateways. Gateways support communications between adjacent clusters[3,10].

Within each cluster, the MAC protocol provides for efficient transfer of packets between neighbors. For our experiments we have selected polling. Namely,the clusterhead polls the nodes to allocate the channel. Polling was chosen here for several reasons. First, polling is consistent with the IEEE 802.11 standard (Point Coordination Function)[1]. Secondly, polling gives priority to the clusterhead, which is desirable since only gateways and clusterheads can be *FG* members, and thus routes are forced to go through clusterheads.

For the sake of simplicity we assume that nodes (and in particular gateways) can receive on multiple codes simultaneously (e.g., using multiple receivers). This property does not enhance communications within a cluster, since all wireless nodes are tuned to the same code anyway.

Mobility (km/hr)	Soft state parameters (time interval in ms)			
	Member Advertising		FW Tab refresh	FG timeout
	refresh	Timeout		
0.02	400	960	200	560
0.70	400	960	200	560
1.41	400	960	200	560
2.81	400	960	200	560
5.62	400	960	200	560
11.25	400	960	200	560
22.50	400	960	160	480
30.00	400	960	120	400
45.00	400	960	80	320
60.00	400	960	60	280
90.00	400	960	40	240

Table 4. Soft state parameters

It does, however, permit conflict free communications with the gateways, and in particular conflict free multicast from clusterhead to gateways. Without the multiple code reception, the gateway must tune on different codes (of the adjacent clusters) and can receive correctly only if it is tuned to the transmitting clusterhead code. An example is offered in[4].

Nodes have a finite buffer. Packets are dropped when buffers overflow, or when there is no route to the intended destination. The latter occurs when the topology is disconnected or the route is not available. Packet drop, channel interference, noise, fading and mobility lead to packet loss, thus making the multicast protocol just described unreliable. End to end reliable delivery can be restored with SRM. SRM works at the transport application level and can be built directly on top of our multicast and exploits our cluster infrastructure (but not our multicast protocol).

A multihop, mobile wireless network simulator was developed using the parallel simulation language Maisie/PARSEC[11,16]. The simulator is very detailed in that it models all the control message exchanges at the cluster, MAC (e.g., polling) and network layer (Distance Vector routing tables and join/quit m-cast messages). Thus, the simulator enables us to monitor the traffic overhead of the protocols. The network consists of 100 mobile hosts roaming randomly at a predefined average speed in a 1000x1000 meter square. At each time tick, a random direction and step size are chosen. A reflecting boundary model is assumed. Radio transmission range is 120 meters. Free space propagation channel is assumed unless otherwise specified. Data rate is 2 Mb/s. Packet length is 10 kbit for data, 2 kbit for routing tables, and 500 bits for MAC control packets and multicast control tables. Thus, transmission time is 5 ms for data packet, 1 ms for routing table, and 0.25 ms for control packet. Buffer size at each node is 10 packets.

Routing tables and control messages have higher priority over data. Channel overhead (e.g., code acquisition time, preamble, etc.) is factored into packet length. Routing tables are updated every second. This low update rate is consistent with typical wired network operation and is adequate for a static network. As node mobility increases, however, the topology starts changing rather rapidly. In order to maintain accurate routing information, changes in local link status and new routing tables from neighbors trigger new updates. Other soft state parameters are listed in table 4.

6. Performance Evaluation

The simulator described above is used to evaluate SRM. The multicast protocol Is FGMP-RA. The multicast membership configuration is one to many multicast. The sender *S* sends 400 multicast packets at the rate of *1/ !* = 100 *ms*. The end-of-session message containing the end of sequence number (400) is sent to all receivers by the way of reliable unicast. In addition to multicast, there is light background uniform unicast load (datagram) originating from each node at the rate of *1/ !* = 5 *sec*. The timer for a patch request and a patch has a uniform distribution in an interval based on RTT (Round Trip Time). The patch has the same packet size as data packet (10k bits) while the message size of the patch request is equal to the control message size(500 bits). Two sets of experiments are simulated: FGMP and FGMP+SRM. Performance measures are based on throughput, average delay, average hop, and maximal buffer. The patch overhead and performance are evaluated as well to explore the efficiency of SRM. Sets of experiments using unicast patching are developed as well to compare the performance with SRM. A 2-level mobility model is also used to evaluate the performance under more stable environments.

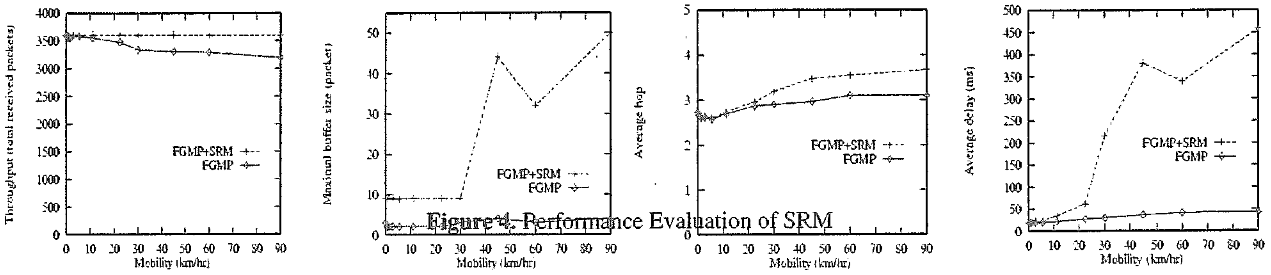


Figure 4. Performance Evaluation of SRM

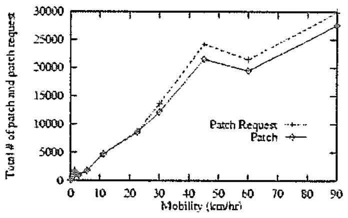


Figure 5. Patch O/H

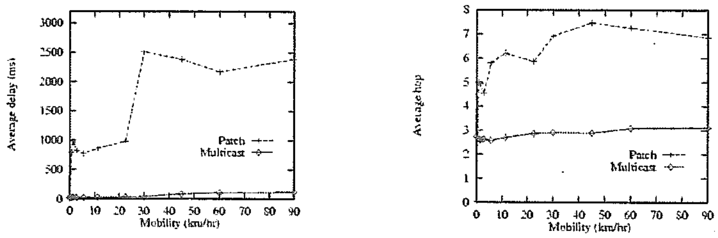


Figure 6. Performance Measure of Patch

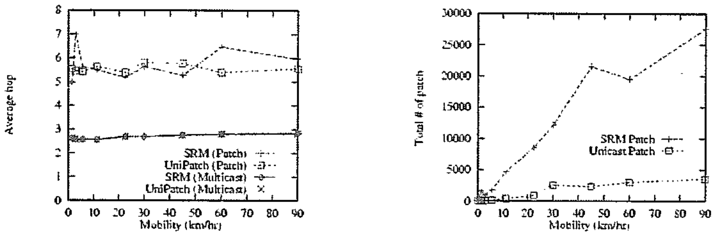


Figure 7. Patch Comparison of SRM vs. Unicast

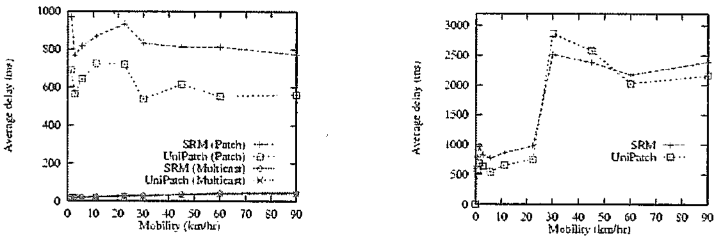


Figure 8. Patch Comparison of SRM vs. Unicast Patching (2-level Mobility)

6.1 Performance Evaluation of SRM

The information carried in the packet header includes sender ID, sequence number, hop count, and origin time stamp. The hop count is increased by one when the packet is forwarded. Upon receiving multicast packets, the receiver members compute the throughput (received packets excluding duplicates), hop count, and delay. Accumulated results are collected to analyze the average values (hop count and delay). Figure 4 compares the results of FGMP and FGMP+SRM.

Throughput: The throughput after recovering the loss is 3600 since there are nine receiver members (Type-1) and 400 packets sent by the sender. As expected, SRM is able to recover all packet losses since the patch requests are flooded to all multicast members and any of members can send the requested patches by flooding (if it is not cancelled by other patches). Without SRM, packet loss is increased with mobility. Since the multicast traffic load is very light and there is no packet dropping due to the buffer overflow. Packets are lost because the forwarding group cannot catch up the moving nodes and thus some packets are not forwarded correctly.

Average Delay: The packet delay is measured at receiver members by subtracting the origin time, which is carried in the packet header, from the reception time. For SRM, the patch carries the origin time from the multicast sender rather than from the node which originates the patch. From figure 4 we note that the average delay of SRM is much higher. This is due to the delay (timer) of issuing the patch requests and the patches and the channel competition for flooding the patches.

Average Hops: Like the average delay, the hop count is computed from the multicast packet rather than counting from the patch source. Thus, the hop count of a patch, which is originated from member *P* and is received at receiver *R*, is the path length from multicast sender *S* to receiver *R* via member *P*. As expected, the average hops increase after including the patches.

Maximal Buffer: The maximal buffer size required is measured during the experiment. We assume that there are unlimited buffers available and the maximal buffer size is recorded. SRM needs larger buffers at higher mobility to store the patches flooded from multicast members in order to recover the packet loss.

6.2 Patch analysis

To understand the SRM messages (patch and patch request), we further explore the patch behavior. Figure 5 shows the total number of patches and patch requests transmitted during the experiment. The patches and patch requests, which are identified by the sender ID and sequence number, are suppressed if they are duplicates. The number of patches are smaller than the one of patch requests but the patch size is larger than the request size, and thus the patch has much larger overhead than the patch request. Figure 6 compares the multicast packets (packets which are not lost) with the patch packets (recovered packets). Average delay and hop count is larger for patch packets than for multicast packets.

6.3 Performance of Unicast Patches

SRM repairs packet losses by sending patch requests to all members which respond the requests if the requested packets have been received. Timers are set in order to avoid duplicate requests and responses. The patch requests and responses are flooded to the networks in order to reach the member and to suppress the duplicates. However, flooding increases the channel overhead and thus reduces the efficiency. From figure 5 we note that the patch overhead is very large at high mobility. Figure 7 shows the results of unicast patching. In general, without flooding patches, unicast patching reduces much of the patch overhead and gains better response. The delay spike in the middle of the mobility spectrum occurs for both schemes and is worse for unicast patching. This is due to the temporary disconnection of network topology. When the receiver is disconnected from other members, unicast patching has a longer recovery period than flooding. To measure the performance under a more stable environment, A 2-level mobility model are involved, where the clusterheads are slow nodes and move at 1.41 km/hr, thus providing a highly connective clustering. Figure 8 shows the average delay of patches and non-lost multicast packets. Unicast patching achieves less delay as expected.

7. Conclusion

Wireless communication provides an efficient and economical means for frequent roamers to communicate. The benefits of wireless networks are mobility, easy and rapid installation, and ubiquitous transmissions. The Multihop infrastructure allows rapid deployment and dynamic reconfiguration; it provides the feasible networking solution for a very dynamic environment such as battlefield communications and disaster recovery operations. Multicasting is very important in wireless networks because it reduces the channel overhead incurred by redundant and duplicate transmissions. In this paper we explore reliable multicast which provides end-to-end recovery mechanisms. The Forwarding Group Multicast Protocol (FGMP), which is using forwarding nodes instead of trees, and the Scalable Reliable Multicast (SRM), which is based on application level framing, have been implemented into our protocols and infrastructure via simulations, thus providing a reliable solution for wireless multicasting. A unicast patching scheme has been proposed to reduce the overhead of flooding, thus achieving better performance. Simulation results show that unicast patching is more suitable for high mobility wireless networks.

References

[1] B.P. Crow, I.Widjaja, J.G Kim, and P.Sakai. Investigation of the ieee 802.11 medium access control (MAC). In *IEEE INFOCOM*, 1997.

[2] C.-C. Chiang, H.-K. Wu, W.Liu, and M.Gerla. Routing in clustered multihop, mobile wireless networks with fading channel. In *The IEEE Singapore International Conference on Networks*, pages 197--211, 1997.

[3] C.-C. Chiang, M.Gerla, and L.Zhang. Forwarding group multicast protocol (FGMP) for multihop, mobile wireless networks. *Special Issue of Cluster Computing: the Journal of Networks, Software Tools and Applications*, 1(2), 1998.

[4] C.R. Lin and M.Gerla. Maca/pr: An asynchronous multimedia multihop wireless network. In *IEEE INFOCOM*, 1997.

[5] D.Clark and D.Tennenhouse. Architectural considerations for a new generation of protocols. In *ACM SIGCOMM*, pages 201--208, 1990.

[6] E.Pagani and G.P. Rossi. Reliable broadcast in mobile multihop packets networks. In *ACM MOBICOM*, pages 34--42, 1997.

[7] K.L. Calvert and E.W. Zegura. Core selection methods for multicast routing. Technical report, GIT-CC-95/15, 1995.

[8] M.G Ching-ChuanChiang and L.Zhang. Adaptive shared tree multicast in mobile wireless networks. In *IEEE 1998 Global Telecommunications Conference (GLOBECOM'98)*, pages 1817--1822, 1998.

[9] M.Gerla and C.-C. Chiang. Shared tree multicast with RP relocation in mobile wireless networks. Technical report, UCLA-CSD, Jan. 1998.

[10] M.Gerla and J.T.-C. Tsai. Multicluster, mobile, multimedia radio network. *ACM/Baltzer Journal of Wireless Networks*, 1(3):255--265, 1995.

[11] R.Bagrodia and W.Liao.Maisie: A language for the design of efficient discrete-event simulations. IEEE Transactions on Software Engineering, 20(4):225--238, 1994.

[12] S.Deering, D.Estrin, D.Farinacci, V.Jacobson, C.-G Liu, and L.Wei. The PIM architecture for wide-area

multicast routing. *IEEE/ACM Transactions on Networking*, 4(2):153--162, April 1996.

[13] S.E. Deering and D.R. Cheriton. Multicast routing in datagram internetworks and extended LANs. *ACM Transactions on Computer Systems*, 8(2):85--111, May 1990.

[14] S.Floyd, V.Jacobson, S.McCanne, C.-G. Liu, and L.Zhang. A reliable multicast framework for light-weight sessions and application level framing. In *ACM SIGCOMM*, pages 342--356, 1995.

[15] T.Ballardie, P.Francis, and J.Crowcroft. Core based trees (CBT) an architecture for scalable inter-domain multicast routing. In *ACM SIGCOMM*, pages 85--95, 1993.

[16] W.W. Liu, C.-C. Chiang, H.-K. Wu, V.Jha, M.Gerla, and R.Bagrodia. Parallel simulation environment for mobile wireless networks. In *1996 Winter Simulation Conference Proceedings (WSC'96)*, pages 650--612, 1996.