

# Forwarding Group Multicast Protocol (FGMP) for Multihop, Mobile Wireless Networks

Ching-Chuan Chiang, Mario Gerla and Lixia Zhang  
Computer Science Department  
University of California, Los Angeles

## Abstract

In this paper we propose a new multicast protocol for multihop mobile wireless networks. Instead of forming multicast trees, a group of nodes in charge of forwarding multicast packets is designated according to members' requests. Multicast is then carried out via "scoped" flooding over such set of nodes. The forwarding group is periodically refreshed to handle topology/membership changes. Multicast using forwarding group takes advantage of wireless broadcast transmissions and reduces channel and storage overhead, thus improving the performance and scalability. The key innovation with respect to wired multicast schemes like DVMRP is the use of flags rather than upstream/downstream link state, making the protocol more robust to mobility. The dynamic reconfiguration capability makes this protocol particularly suitable for mobile networks. The performance of the proposed scheme is evaluated via simulation and is compared to that of DVMRP and global flooding.

**Keywords:** Wireless Network, Multihop, Multicast, Ad hoc, Clustering, FGMP

## 1 Introduction and Background

In this paper we introduce a novel multicast scheme for a mobile, multihop wireless network with no fixed infrastructure [1, 2]. Various multicast schemes have been proposed for such an environment. One scheme creates a per-source multicast tree for each sender source [3]. Packets are multicast on the tree using Reverse Path Forwarding (RPF) for duplicate detection. We will show that RPF is not very effective in high mobility environments. Another is using a shared tree spanning the members in the multicast group [4, 5]. Data sent to the shared tree are forwarded to all receiver members. For the shared tree multicast, it is necessary to maintain a "core" or Rendezvous Point (RP) for sender and receiver paths to meet. RP mobility may affect multicast efficiency. Some schemes use sets of RPs [6] to direct multicast routing and resource reservation. The mobile RPs tend to increase the overhead of RP selection and thus reduce multicast efficiency.

In this paper we propose a multicast protocol which requires minimal infrastructure (without RP) and is thus resilient to mobility. Yet it achieves good efficiency by exploiting the inherent broadcast property of the wireless medium. In essence, the protocol is a hybrid between flooding and shortest tree multicast. In a static network it does converge to per source multicast. For this reason, we will compare its performance to that of flooding and DVMRP [3], the latter being a popular per source tree implementation.

Section 2 describes DVMRP which is used in the internet (MBone) and extends it to mobile wireless networks. Section 3 describes the novel multicast protocol based on "forwarding group".

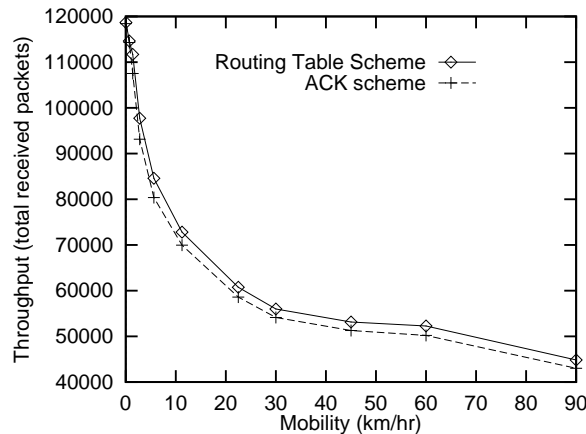


Figure 1: Throughput of different leaf node detection schemes

Section 4 reviews the protocol infrastructure and the simulation environment, and details the performance evaluation. Section 5 concludes the paper.

## 2 Distance Vector Multicast Routing Protocol (DVMRP) for a wireless environment

In DVMRP, each sender uses flooding to direct the multicast packets to all nodes within a specified range (defined by TTL). Multicast packets are selectively forwarded according to the reverse shortest path forwarding protocol [3]. Non-member leaf nodes and nodes without any downstream members send prune messages upstream to prune off branches to non-member nodes. After timeout, pruned branches become alive again and get flooded with messages. A new receiver member can also send a graft message to upstream nodes in order to speed up the connect process.

There are some problems in the use of DVMRP in mobile wireless networks. One problem is the leaf node detection. In a multihop mobile network, all nodes can function as routers; thus, there is no explicit subnet. IGMP is not suitable for this environment. In section 2.1, we explore two schemes to detect the leaf nodes and show their performance via simulation. Another problem is the data flooding overhead. Because upstream nodes may change or be disconnected due to node mobility, it is necessary to reflow by exploring pruned branches after timeout in order to reestablish the upstream information, reconnect lost members, or allow new members to join. In addition, reflowing is needed to confirm the existence of the sender source. A sender is declared non-existent after timeout. This periodical reflowing causes very large transmission overhead for the low bandwidth wireless channel. The other problem is the Reverse shortest Path Forwarding (RPF). Since DVMRP uses the RPF mechanism, packets are accepted only from the shortest path. If the shortest path changes and there are no packets to be forwarded in the new path, the node will be disconnected from the tree (since it will not accept packets from the old path). Section 2.2 presents a modified DVMRP scheme to reduce reflowing overhead and adapt to changing topology.

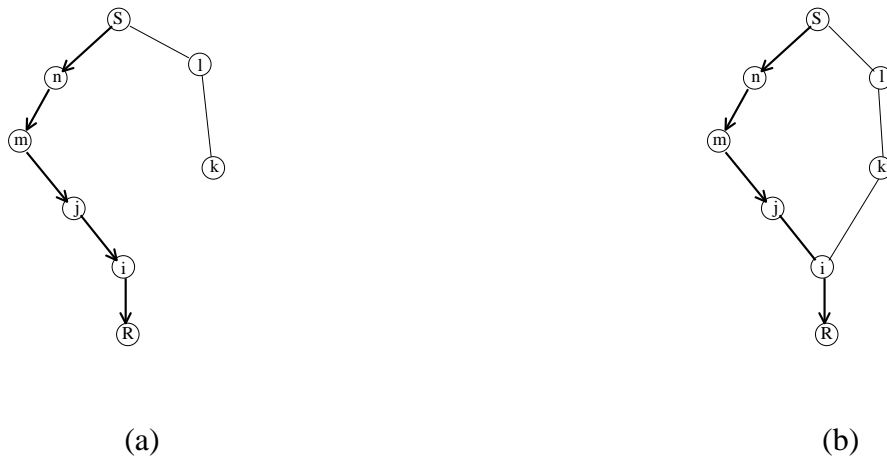


Figure 2: Example of upstream change

## 2.1 Leaf Nodes Detection

In DVMRP, leaf routers are responsible for sending prune messages to upstream routers when multicast packets arrive and there is no member on the leaf subnet. For the wireline networks, each router has explicit link information to decide which links are child links for a given source. Multicast packets are forwarded only to child links. However, it is not trivial for a wireless node to decide whether it is a leaf node. For multihop wireless networks, all nodes are routers and there is no explicit link interface information to determine the leaf status. Here we propose two schemes for detecting the leaf nodes for a source. One is using ACK to detect the leaf node. When a node  $N$  receives a multicast packet from source  $S$  via RPF, it transmits the packet and all neighbors will hear it. Neighbors which are on the reverse shortest path via  $N$  to  $S$  will accept the packet and send ACK to  $N$ . Node  $N$  is a leaf node if there is not any ACK coming back (from its neighbors) after a timeout interval  $T$ .  $T$  is setup to accommodate the round-trip delay for all neighbors. The overhead of ACK scheme is the ACK traffic and the delay for  $N$  to send back the prune message to its upstream node. Another scheme is to use neighbors' routing tables. Node  $N$  is able to find out if there exists any downstream neighbor for source  $S$  by checking its neighbors' routing tables (which are periodically exchanged). A neighbor is the downstream node if it has longer distance to  $S$  via  $N$  or if it advertises  $N$  as the next node to destination  $S$ . Node  $N$  is a leaf node if there is no downstream neighbor. Prune messages can be sent to upstream with lower delay than with the ACK scheme. However, the overhead of this scheme is the space required to store all neighbors' routing tables.

Figure 1 compares the performance of these two schemes using the simulator described in section 4.

## 2.2 Adaptive Reverse Shortest Path Forwarding

DVMRP routers set a timer for each pruned-off downstream and upstream link. Multicast packets are reflooded to pruned-off downstreams when the timer expires. Reflooding is necessary for the following reasons: (1) to pick up new members who do not have source information (i.e., upstream to source), (2) to update per-source tree information, and (3) to refresh source status. A router sends a prune message to its upstream if it is the leaf router and all its downstreams are pruned off. When a node receives a multicast packet with source  $S$ , it stores (or updates) the upstream link

and the timer for  $S$ . If a node not on the tree wants to join this group, it can send graft messages directed to the senders of this group. In a mobile environment, it is frequent to find nodes without upstreams (new coming nodes, disconnected nodes, etc.). Reflooding is necessary to pick up new members. For a node on the tree, it is possible that the multicast traffic stops due to upstream link change. For example, in figure 2 the multicast tree from source  $S$  to receiver member  $R$  is  $S \rightarrow n \rightarrow m \rightarrow j \rightarrow i \rightarrow R$ . When the topology changes from figure 2(a) to figure 2(b), node  $i$  will not accept packets from  $j$  but from  $k$  (new shortest path), however, there is no traffic coming from  $k$  because  $l$  and  $k$  are not forwarding any packets from  $S$  (they are pruned off branches). Reflooding will correct this situation establishing the new tree  $S \rightarrow l \rightarrow k \rightarrow i \rightarrow R$

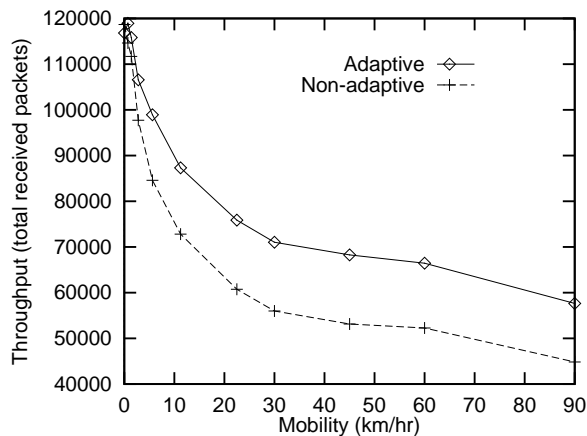


Figure 3: Throughput of adaptive vs. non-adaptive

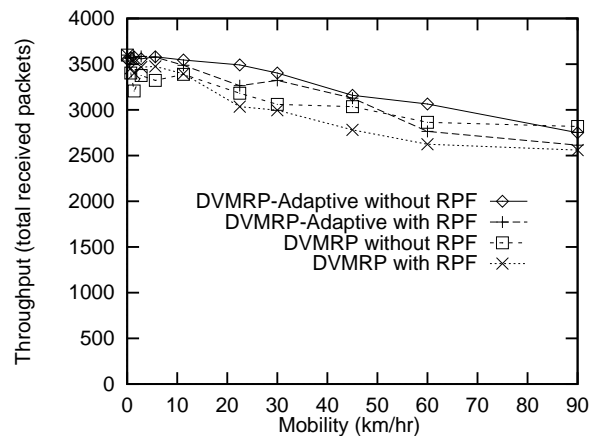


Figure 4: Throughput of DVMRP

Reducing the reflooding overhead is very important for the bandwidth-limited wireless channel. However, the mobile environment needs reflooding to reconfigure the multicast trees. We propose a variant of DVMRP called “adaptive” DVMRP, which reduces the reflooding frequency in a dynamic changing topology. Adaptive DVMRP monitors the routing information. If the shortest path to the source changes, the node sends a graft message to the new upstream and a prune message to the old upstream. For example in figure 2(b) node  $i$  detects the change and sends graft message to  $k$  and prune message to  $j$ . Reflooding for topology adjustment is no longer required. We still need reflooding however to pick up new members. In addition, status of the senders (live or expired) must be refreshed via reflooding. A scheme similar to “adaptive” DVMRP has been proposed in [7], but it provides only adaptive grafting, not pruning.

Figure 3 shows the improvement of adaptive DVMRP. Table 1 presents the reflooding periods (pruned-off branch timer) used in the experiment (Type 1, Load A).

### 2.3 Reverse path Forwarding limitations

For a mobile environment, RPF is not very efficient. To prove this, we evaluate DVMRP with and without RPF. Without RPF, duplicates are detected by packet id numbers (e.g., source and sequence number). The first packet delivered by flooding is accepted and the corresponding transmitter is marked as the upstream node. Figure 4 shows the throughput results (Type 1, Load B). RPF causes performance degradation in both versions of DVMRP for medium and high mobility.

Table 1: Reflooding period

Mobility (km/hr)	Reflooding period (ms)	
	non-Adaptive	Adaptive
0.02	20000	2000
0.70	4000	2000
1.41	2000	2000
2.81	1000	2000
5.62	800	2000
11.25	700	2000
22.50	600	2000
30.00	500	2000
45.00	400	2000
60.00	300	2000
90.00	200	2000

## 2.4 Protocol Overhead

To compare with our new protocol, we address the overhead of DVMRP in mobile wireless networks. There are two types of overhead: channel and storage. The channel overhead comes from redundant flooding. The frequency of flooding is dependent on the timers of pruned off branches. For each pruned off branch, there is a timer associated with it and multicast packets will be forwarded after the timeout. The channel overhead is worse for wireless networks than wireline networks since one timeout branch will cause packet flooding. The storage overhead corresponds to the storage space for tree status. For each sender source, each node needs to store information of upstream, downstreams, and some maintenance status. The information includes timers and status flags. If there are  $S$  senders for a group and  $D$  downstreams on average, the storage overhead of a node is  $S * (D + 1) * N$  bytes, where  $N$  is number of bytes for each link. In section 5.3.2 we evaluate the storage overhead in our simulation.

## 3 Forwarding Group Multicast Protocol (FGMP)

In a wireless broadcast channel, there is no notion of explicit link interface like in a wired point to point channel. Multicast forwarding is based on nodes (routers) which are going to accept multicast packets rather than on links on which multicast packets are forwarded. Traditional multicast protocols based on upstream and downstream links (like CBT [4, 8], PIM [5] and DVMRP) are not suitable here because creating and maintaining upstream and downstream link status in a wireless network is not efficient.

The proposed multicast protocol keeps track not of links but of groups of nodes which participate in multicast packets forwarding. To each multicast group  $G$  is associated a forwarding group,  $FG$ . Any node in  $FG$  is in charge of forwarding (broadcast) 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 and then broadcast it in turn. Figure 5 shows an example of a multicast group containing

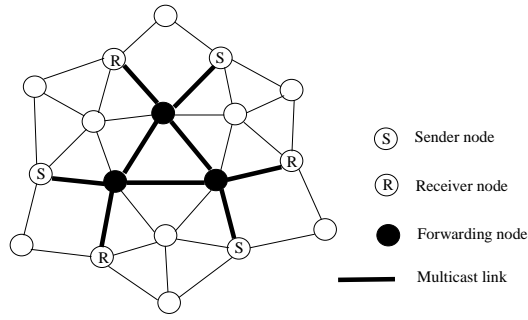


Figure 5: An example of FGMP

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. It is interesting to note that with proper selection of the forwarding group, the  $FG$  scheme can emulate any of the existing schemes. For example, to produce global flooding, the  $FG$  must include all nodes in the network. For CBT, the  $FG$  is restricted to the nodes on the shared tree except the leaf nodes. In DVMRP,  $FG$  includes all the non-leaf nodes on the source trees.

Only one flag and a timer are needed for each forwarding node. When the forwarding flag is set (as described in following subsections), each node in  $FG$  forwards data packets belonging to  $G$  until the timer expires. Regardless of member group size, one forwarding flag is enough to guarantee efficient delivery of multicast packets. Storage overhead, a major problem in traditional multicast protocols, is low, thus increasing the scalability. Timer is refreshed by the forwarding group updating protocol. Stale forwarding nodes are deleted from  $FG$  after timeout. The soft state approach of using a timer works well in dynamic changing environments.

The major problem of FGMP is how to elect and maintain the set  $FG$  of forwarding nodes. The size of  $FG$  should be as small as possible to reduce wireless channel overhead, and the forwarding path from senders to receivers should be as short as possible to get high throughput.

The key to efficient multicasting is a mechanism which allows the network to forward packets efficiently from senders to receivers without resorting to global flooding. Unless the senders have full membership information, some limited flooding is required to discover the members. For example, DVMRP periodically floods multicast packets to update the multicast information (upstream, downstreams, and source information). However periodic flooding of packets is not cost-efficient especially for wireless channel. We use a new scheme which has lower overhead (both in channel and storage) than DVMRP, thus improving the performance. To reduce the overhead, we propose to flood explicit sender or receiver membership (rather than data packets as in DVMRP). Namely, instead of flooding data packets like DVMRP, we only flood small size control messages and with less frequency. In DVMRP, flooding is needed to refresh senders’ status and reestablish multicast states (upstream and downstream) as explained in section 2. To adapt to the changing environment (mobility), the flooding frequency must be increased as mobility increases. Thus, high mobility needs high frequency flooding to maintain and reestablish the multicast status, but this will increase the channel overhead and degrade the performance. Our proposed membership advertising scheme only refreshes the membership. Channel overhead is much lower than DVMRP, making this scheme

effective in a mobile wireless environment.

In the proposed scheme, the decision to forward multicast packets depends on a forwarding flag. The forwarding flag is associated with a timer (“soft state”). When a node in  $FG$  learns of a receiver member (described in detail in following sections), it resets its forwarding timer. A node with enabled forwarding flag (i.e., timer has not expired) is responsible for forwarding the multicast packets for that group. Due to the inherent broadcast property of wireless transmission, it is not necessary to store the downstream link status as in DVMRP. When a node receives a multicast packet and the forwarding flag of that group is enabled, it just broadcasts it to its neighbors. Only neighbors with enabled forwarding flag will accept the packet. In DVMRP, every node needs to store the upstream and downstream link information for each sender, thus reducing the scalability. Here, instead of storing all downstream links (live or pruned) as in DVMRP, only the forwarding flag and timer are stored, thus reducing the storage overhead and increasing the flexibility and performance.

Table 2: Format of join request packet

Mcast Group id	Id	Sequence #	TTL
----------------	----	------------	-----

Table 3: Format of member table at the sender

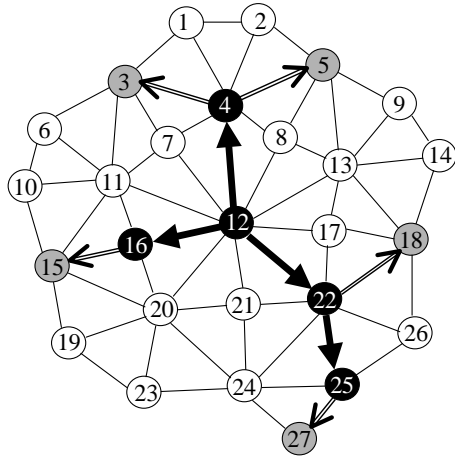
Mcast Group id
Refresh Timer
receiver member id    timer

Table 4: Format of forwarding table  $FW$

Mcast Group id
receiver member id    next hop

### 3.1 $FG$ Maintenance via Receiver Advertising (FGMP-RA)

One way to advertise the membership is to let each receiver periodically and globally flood its member information formatted as in table 2. When a sender receives the join request from receiver members, it updates the member table (format in table 3). Expired receiver entries will be deleted from the member table. The sender will broadcast multicast data packets only if the member table is not empty. After updating the member table, the sender creates from it the forwarding table  $FW$  shown in table 4. Next hop information 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



Sender = {12}  
 Receivers = {3,5,15,18,27}  
 FG = {4,12,16,22,25}

→ Forwarding Tables flow  
 ⇒ Mcast data flow

Forwarding Table of Node 12:

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

(a)

Forwarding Table of Node 22:

Receivers	Next hop
27	25

(b)

Figure 6: Example of Forwarding tables

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. The forwarding group is created and maintained via the forwarding table  $FW$  exchanges. 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 provides the ability to adapt to a changing topology.

Figure 6 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 6(a),(b)) and broadcast it. Forwarding nodes 4, 16, and 25 do not need to create their forwarding tables since they are “leaves”, i.e., all receiver members are immediate neighbors. It is critical to note that the forwarding tables are NOT STORED like routing tables. They are created and broadcast to the neighbors only when new forwarding tables arrive. When forwarding nodes receive new forwarding tables, their forwarding timers are refreshed; in absence of refreshes, the forwarding flag will automatically time out and the forwarding node is deleted from  $FG$ .

### 3.2 $FG$ Maintenance via Sender Advertising (FGMP-SA)

Another way to advertise the membership is to let senders flood sender information. Sender advertising is more efficient than receiver advertising if the number of senders is less than the number of receivers. Most multicast applications belong to this category. Like with receiver advertising, senders periodically flood the sender information. Receivers will collect senders’ status, then peri-



odically broadcast “joining tables” to create and maintain the forwarding group  $FG$ . The “joining table” has the same format as the “forwarding table” except that the joining table contains the sender ids while the forwarding table contains receiver ids. Forwarding flag and timer are set when a node receives the joining table. Forwarding group is maintained (soft state refresh) by the senders in receiver advertising scheme and by the receivers in sender advertising scheme. Both schemes achieve higher performance than DVMRP (due to the lower flooding overhead) and result in low storage overhead.

### 3.3 Protocol Overhead

The membership flooding overhead is much smaller than DVMRP because of the smaller size of control messages and longer flooding intervals as we explained in section 3. The storage overhead of membership advertising schemes is also greatly reduced. For example in DVMRP, if there are  $S$  senders,  $R$  receivers, and  $D$  downstream links on average for a group, every node needs  $S*(D+1)*B$  bytes, where  $B$  is number of bytes for each link, to store upstream and downstreams status. However, for FGMP, each forwarding node (not all nodes) only needs one flag and one timer for a group. In sender advertising scheme, a receiver needs an additional storage of  $S * B'$  bytes for membership table, where  $B'$  is number of bytes for entry in joining table.

## 4 Network Infrastructure & Simulation Environment

### 4.1 Multihop Network Infrastructure

The  $FG$  multicast protocol presented so far requires the availability of routing tables, but is otherwise independent of lower layer protocols. The overall performance does, however, depend on the protocol infrastructure. For this reason, we have developed and simulated in detail a specific multihop wireless infrastructure to be used in our experiments.

The infrastructure is a clustered multihop infrastructure [9, 1]. The aggregation of nodes into clusters under clusterhead control provides a convenient framework for the development of efficient protocols both at the MAC layer (e.g., code separation among clusters, channel access, bandwidth allocation) and at the network layer (e.g., hierarchical routing) [1]. At the MAC layer, the main objective of clustering is efficient use of the medium, while at the network layer the hierarchical routing induced by clustering provides scalability and robustness to mobility. In our distributed clustering algorithm, nodes are elected as clusterheads based on preferential criteria (e.g., lowest ID number, etc.). All nodes within transmission range of a clusterhead belong to the same cluster. That is, all nodes in a cluster 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. In a mobile network, an important criterion in cluster algorithm design is stability. Frequent clusterhead changes adversely affect the performance of other protocols such as scheduling and resource allocation. We use the Least Clusterhead Change (LCC) clustering algorithm [9], which minimizes clusterhead changes. One may also introduce the “preferential” condition that clusterheads are chosen among “slow moving” nodes. This way, the stability of hierarchical routing (and, consequently, of the forwarding group) is greatly improved. Beside clusterhead election, additional procedures are required to manage clusters. For example, if spreading codes are used, the nodes must agree on a common control code for initialization and for reconfiguration [1]; and on the

selection of orthogonal codes in adjacent clusters, etc. Specific solutions are reported in [10].

Cluster maintenance protocols run continuously in the background in order to adjust to node movements and dynamically reconfigure the cluster structure accordingly. Average convergence time of the clusterhead election algorithm is  $O(1)$ , that is, it does not depend on network size  $N$  and thus scales well [1]. In fact, clusters are reconfigured as quickly as links are added/deleted. This property implies that the convergence of the routing algorithm (which operates above clustering) is not “slowed down” by the presence of clusters.

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) [11]. Secondly, polling gives priority to the clusterhead, which is desirable since routes are forced to go through clusterheads in the 2-level cluster routing. Third, polling permits easy support of real time connections (which can be scheduled at periodic intervals by the clusterhead). Fourth, in our experiments each cluster has on average six neighbors (which is the optimal value in a uniform multihop architecture [12]); thus polling latency is not a critical concern. For larger cluster size the polling scheme can be replaced by a polling/random access scheme, to reduce latency. This is accomplished by defining a two phase cycle. In the first phase, the clusterhead transmits its packets and polls nodes with real time connections; in the second phase, backlogged nodes access the channel using the CSMA/CA protocol [13]. This latter scheme is also consistent with IEEE 802.11 (Distributed Coordination Function). In this paper we only consider polling for simplicity.

For the sake of simplicity we also 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. 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. For example, in [14] we assume that each gateway takes turns in tuning to the codes of the adjacent clusters for a fraction of time dependent on the traffic pattern. Reliable MAC layer broadcast can be restored by transmitting a separate unicast message to each gateway (via polling), or by performing repeated broadcasts with staggered ACKs after each transmission. We are currently evaluating these and other schemes for reliable MAC layer multicast. For the purpose of this study, however, the overhead penalty of any of the above remedies will affect equally the multicast schemes we are comparing in this paper (i.e., FGMP, DVMRP, flooding) without changing the ranking. Thus, for simplicity, we have opted for the multiple code reception.

Nodes have a finite buffer. Packets are dropped when buffers overflow, or when there is no route to the intended destination, because the topology is disconnected or the routing tables have not yet been updated. End to end reliable delivery can be implemented with other means such as Scalable Reliable Multicast [15, 16]. SRM works at the transport application level and can be built directly on top of our multicast scheme. Another reliable wireless multicast scheme is reported in [16]. That scheme works at the network layer and exploits our cluster infrastructure (but not our multicast protocol). To reduce packet drop rate due to congestion, access control and internal congestion control procedures (e.g., back pressure) can be built on top of the MAC layer.

As mentioned earlier, we use a hierarchical routing protocol. Namely, packets travel from source to destination through an alternation of clusterheads and gateways [9]. The actual routing protocol used in our experiment is an extension to the wireless environment of the DSDV (Destination-

Table 5: Multicast Traffic Load

Member Configuration	Traffic Load	$1/\lambda$	Total packets simulated
Type 1	Load A (heavy load)	25 ms	unlimited
	Load B (light load)	50 ms	400
Type 2	Load C	250 ms	unlimited

Sequenced Distance-Vector) scheme originally developed for wireline networks [17]. DSDV is a “distance vector” type algorithm with the same complexity as Bellman-Ford or RIP, but with better protection against loops.

## 4.2 Simulation Environment

A multihop, mobile wireless network simulator was developed using the parallel simulation language Maisie [18, 19]. The simulator is very detailed in that it models all the control message exchanges at the MAC layer (e.g., polling) and network layer (DSDV routing tables and join/quit m-cast messages). Thus, the simulator enables us to monitor the traffic O/H of the protocols. The network consists of 100 mobile hosts roaming randomly in all directions at a predefined average speed in a 1000x1000 meter square. 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. Furthermore, changes in link status and routing table trigger new updates.

Two multicast membership configurations are evaluated. Type 1 configuration consists of one sender and 9 receiver members. Type 2 configuration has 10 members and each of them is both sender and receiver. Type 1 traffic pattern corresponds to a broadcast service such as video on demand, while type 2 corresponds to audio/video conferencing. Packet interarrival times are exponentially distributed with mean  $1/\lambda$ . Table 5 shows the traffic load used for type 1 and type 2. In addition to multicast, there is light background uniform unicast load originating from each node at the rate of  $1/\lambda = 5sec$ .

Total simulation time for each experiment is  $4 \times 10^6$  simulation ticks. One simulation tick corresponds to  $50 \mu s$ . Thus, each run represents 200 seconds of simulated time.

## 5 Performance Evaluation

### 5.1 Throughput & Efficiency

To evaluate multicast performance, we measure the “throughput” at the receivers. The “throughput” is defined as total multicast packets received at all receiver members excluding duplicated packets. As previously explained, not all packets will be delivered because of buffer overflow and dropping.

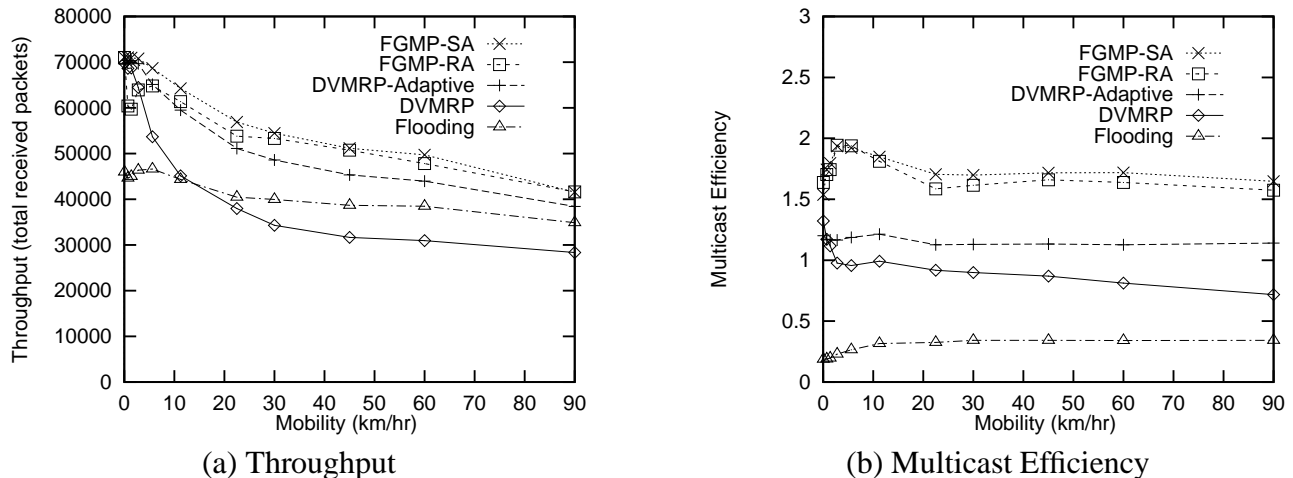


Figure 7: Performance of Flooding, DVMRP, and FGMP (Type 1, Load A)

Throughput performance is affected by two factors: (a) the temporary loss of a multicast route because of mobility, and; (b) the line O/H caused by control messages. Line O/H indirectly causes congestion and buffer overflow. Thus, net throughput is a good cumulative measure of multicast performance.

In a wireless channel, the multicast protocol can take advantage of MAC layer broadcast/multicast. For example, in our proposed scheme, a packet transmitted by a clusterhead is simultaneously received (and accepted) over the wireless channel by several gateways. The efficient wireless multicast scheme makes good use of the MAC layer broadcast facility. It is thus appropriate to evaluate and compare multicast schemes using the performance measure “Multicast Efficiency” defined as:

$$ME = \frac{\text{total number of multicast receptions (along the path as well as at destinations)}}{\text{total number of multicast transmissions (at each node)}}$$

To measure the multicast efficiency, we accumulate the hop count of every multicast packet at each destination (i.e., receiver member). Then,  $ME = \frac{H}{MTX}$ , where  $H$  is the total number of hops accumulated by multicast packets, and  $MTX$  is the total number of packet transmissions. For multicast transmissions, typically  $ME \geq 1$ . If  $ME < 1$ , that means some multicast transmissions are redundant like in the global flooding case. Unicast transmission with possible retransmission will yield a “Transmission Efficiency” less than or equal to 1.

Figure 7(a) compares the throughput of FGMP, DVMRP and flooding. We note that for zero mobility (static network) FGMP and DVMRP achieve the same throughput. Flooding yields lower throughput because of redundant transmissions. As mobility increases, DVMRP performance drops very fast, even below flooding. FGMP and adaptive DVMRP performs quite well. Flooding is almost insensitive to mobility, as expected. Figure 7(b) shows the “Multicast Efficiency” of the various schemes. Global flooding is the least efficient because of the large number of redundant transmissions. FGMP makes the best use of MAC broadcasting. In all, FGMP shows the best performance among all schemes under study. There is little difference between sender and receiver version of FGMP.

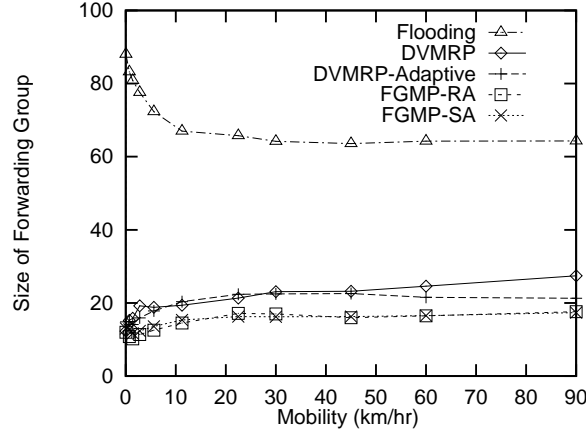


Figure 8: Forwarding Group Size comparison

## 5.2 Forwarding Group Size

The size of the forwarding group  $FG$  should be kept small to minimize the channel overhead. Large  $FG$  size will also degrade throughput efficiency as well, due to increased wireless channel competition. Figure 8 shows the average size of forwarding group for various protocols. The size is measured every 20 ms and averaged over the entire experiment. Flooding has largest  $FG$  size since all clusterheads and gateways are included in  $FG$ . FGMP has the smallest  $FG$  size because of better “scoped” flooding.  $FG$  size of DVMRP increases during reflooding and returns back to normal after pruning, thus ranging between FGMP and Flooding.

## 5.3 Channel & Storage Overhead

### 5.3.1 Channel Overhead

Two types of channel overhead are evaluated. One is the redundant transmission of multicast data; the other is the transmission of multicast control messages. Redundant transmissions include unnecessary flooding and duplicates. Multicast efficiency,  $ME$ , defined in section 5.1 is the appropriate measure for channel overhead caused by redundant transmissions. The more redundancy, the smaller  $ME$  is. The results in figure 7(b) confirm that Flooding and DVMRP have a high percentage of redundant data transmission.

To compute control message overhead, we measure the total number of multicast control bits  $MC_{O/H}$  transmitted during the experiment and divide it by the “network” transmission capacity  $TC$  (expressed in bits). In our simulation environment, total transmission capacity  $TC = B * C * T$  bits, where  $B$  is channel bandwidth,  $C$  is average number of clusters, and  $T$  is total simulation time. For example, in our runs,  $TC$  is 8000 Mbit ( $B = 2$  Mbps,  $C = 20$ ,  $T = 200$ ). Figure 9 shows  $\frac{MC_{O/H}}{TC}$ , i.e., percentage capacity used for multicast control messages. Only FGMP and DVMRP are reported. Flooding has no control overhead. We note that the FGMP sender (SA) version outperforms receiver (RA) version in Type 1 application (video broadcast, one sender, nine receivers) as expected. The two versions show identical O/H in the videoconference application. Control overhead is rather insensitive to speed. Also it increases with the number of senders (Type 2). In all, percentage O/H is less than 4% for FGMP, thus quite manageable.

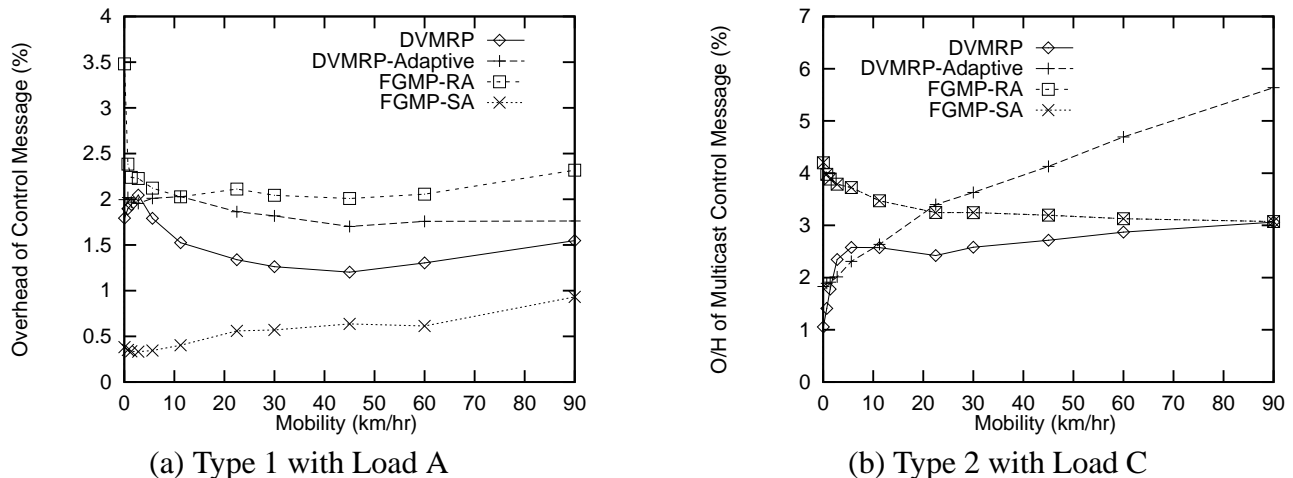


Figure 9: Control Channel Overhead

### 5.3.2 Storage Overhead

To evaluate the storage overhead, we count the storage required for multicast control (upstream, downstream, flag, and timers) excluding buffers for data. The measure is performed every 20 ms and averaged over the experiment duration. For FGMP, average size of forwarding group,  $|FG|$ , is measured, and therefore the average storage overhead  $SO/H_{FGMP}$  is given by  $|FG| * B_{FGMP}$ , where  $B_{FGMP}$  is the storage in bytes needed for forwarding flag and timer. For DVMRP, for each sender  $S_i$ , there is a sender-rooted tree. Average number of leaf nodes,  $Leaf_i$ , and non-leaf nodes,  $NonLeaf_i$  is counted. Average number of downstream nodes (pruned and live),  $Down_i$ , is also measured for a non-leaf node. The average storage overhead is given by:

$$SO/H_{DVMRP} = \left( \sum_i (Leaf_i + NonLeaf_i * (Down_i + 1)) \right) * B_{DVMRP}$$

where  $B_{DVMRP}$  is the storage in bytes required to store the multicast link status (timer, state, etc.). Figure 10 shows the storage O/H per node  $\frac{SO/H}{N}$ , where  $N = 100$ , and  $B_{FGMP} = B_{DVMRP} = 2$ . Flooding protocol is not shown in figure 10 since it incurs no storage overhead. The storage economy of FGMP is evident here.

### 5.4 Light Load Traffic Performance

Figure 11 shows the performance of various multicast schemes for Type 1 configuration in light load (Load B). In light load, packets are dropped when a route is not available, i.e., the multicast infrastructure (e.g.,  $FG$  or multicast tree) has not kept up with node movements. Best performance is offered by global flooding, which in light load does not suffer throughput degradation because of redundant packets, and at the same time will reach all members which are connected to the network. In global flooding, packet loss is caused exclusively by the temporary topology disconnection of some of the members. This problem is emphasized by mobility. We note that FGMP performance is very close to flooding. This indicates that in FGMP, most of the packet loss at high mobility is due to topology disconnections; only a small fraction of the loss is due to the fact that the forwarding group does not keep up with member movements. DVMRP performance is again much inferior to FGMP.

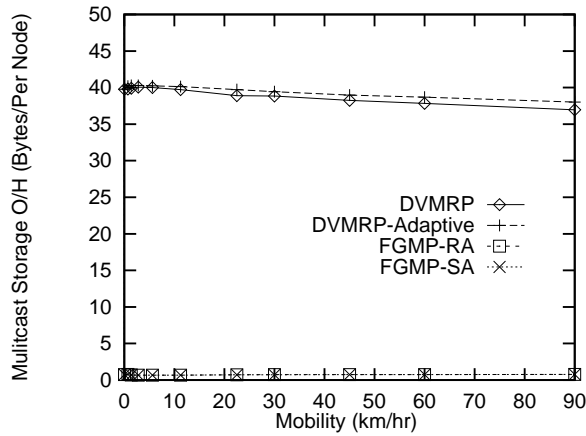


Figure 10: Multicast Storage O/H (Type 2 configuration)

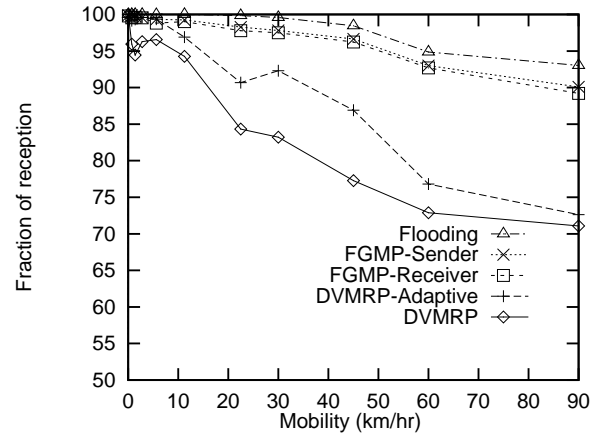


Figure 11: Performance in Light Load Multicast Traffic (mean = 50 ms)

## 6 Conclusion

Forwarding Group Multicast Protocol (FGMP) provides a simple and efficient way for multicasting in multihop, mobile wireless networks. The notion of “forwarding node” is much more suitable to the wireless broadcast channel than the conventional notion of upstream/downstream links. This leads to a more prompt adjustment to topology changes and to a reduction of redundant transmissions, resulting in higher throughput and multicast efficiency. Furthermore, storage overhead savings extend the scalability to large multicast group nodes.

## References

- [1] M. Gerla and J. T. Tsai, Multicluster, mobile, multimedia radio network, in: *ACM Journal on Wireless Networks*, 1(1995)255–265.
- [2] C. R. Lin and M. Gerla, A distributed architecture for multimedia in dynamic wireless networks, in: *IEEE GLOBECOM '95*, 1995, pp. 1468–1472.
- [3] S. E. Deering and D. R. Cheriton, Multicast Routing in Datagram Internetworks and Extended LANs, in: *ACM Transactions on Computer Systems*, 1990, pp. 85–111.
- [4] T. Ballardie, P. Francis and J. Crowcroft, Core Based Trees (CBT) An Architecture for Scalable Inter-Domain Multicast Routing, in: *ACM SIGCOMM '93*, 1993, pp. 85–95.
- [5] S. Deering, D. Estrin, D. Farinacci and V. Jacobson, C.-G. Liu and L. Wei, The PIM Architecture for Wide-Area Multicast Routing, in: *IEEE/ACM Transactions on Networking*, 4(1996)153–162.
- [6] M. S. Corson and S. G. Batsell, A Reservation-Based Multicast (RBM) Routing Protocol for Mobile Networks: Initial Route Construction phase, in: *ACM/Baltzer Wireless Networks*, 1995, pp. 427–450.

- [7] S. E. Deering, Multicast Routing in a Datagram Internetwork, in: *Dissertation, Stanford University*, 1991.
- [8] K. L. Calvert and E. W. Zegura, Core Selection Methods for Multicast Routing, in: *Technical Report, Georgia Institute of Technology (GIT-CC-95/15)*, 1995.
- [9] 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*, 1997, pp. 197–211.
- [10] M. Gerla and C.-C. Chiang, Multicast routing in multihop, mobile wireless networks, in: *Technical Report, Computer Science Department, University of California at Los Angeles*, 1998.
- [11] B. P. Crow, I. Widjaja, J. G. Kim and P. Sakai, Investigation of the IEEE 802.11 Medium Access Control (MAC), in: *IEEE INFOCOM '97*, 1997.
- [12] L. Kleinrock and J. Silvester, Optimum transmission radii for packet radio networks of why six is a magic number, in: *Nat. Telecom. Conf.*, 1978.
- [13] P. Karn, MACA: a new channel access method for packet radio, in: *IEEE Computer Networks Conference*, 1990.
- [14] C. R. Lin and M. Gerla, MACA/PR: An Asynchronous Multimedia Multihop Wireless Network, in: *IEEE INFOCOM '97*, 1997.
- [15] 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 '95*, 1995, pp. 342–356.
- [16] E. Pagani and G. P. Rossi, Reliable Broadcast in Mobile Multihop Packets Networks, in: *ACM MOBICOM '97*, 1997, pp. 34–42.
- [17] C. E. Perkins and P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, in: *ACM SIGCOMM '94*, 1994, pp. 234–244.
- [18] R. Bagrodia and W. Liao, Maisie: A Language for the Design of Efficient Discrete-Event Simulations, in: *IEEE Trans. on Software Engineering*, 1994, pp. 225–238.
- [19] 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)*, 1996, pp. 650–612.