A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols

Sung-Ju Lee, William Su, Julian Hsu, Mario Gerla, and Rajive Bagrodia
Wireless Adaptive Mobility Laboratory
Computer Science Department
University of California
Los Angeles, CA 90095-1596
http://www.cs.ucla.edu/NRL/wireless

Abstract—**In this paper we investigate the performance of multicast routing protocols in wireless mobile ad hoc networks. An ad hoc network is composed of mobile nodes without the presence of a wired support infrastructure. In this environment, routing/multicasting protocols are faced with the challenge of producing multihop routes under host mobility and bandwidth constraints. In recent years, a number of new multicast protocols of different styles have been proposed for ad hoc networks. However, systematic performance evaluations and comparative analysis of these protocols in a common realistic environment has not yet been performed. In this study, we simulate a set of representative wireless ad hoc multicast protocols and evaluate them in various network scenarios. The relative strengths, weaknesses, and applicability of each multicast protocol to diverse situations are studied and discussed.**

I. Introduction

An ad hoc network [13], [15] is a dynamically reconfigurable wireless network with no fixed infrastructure or central administration. Due to the limited radio propagation range of wireless devices, routes are often "multihop." Applications such as disaster recovery, crowd control, search and rescue, and automated battlefields are typical examples of where ad hoc networks are deployed. Nodes in these networks move arbitrarily, thus network topology changes frequently and unpredictably. Moreover, bandwidth and battery power are limited. These constraints, in combination with the dynamic network topology make routing and multicasting in ad hoc networks extremely challenging.

In a typical ad hoc environment, network hosts work in groups to carry out a given task. Hence, multicast plays an important role in ad hoc networks. Multicast protocols used in static networks (e.g., Distance Vector Multicast Routing Protocol (DVMRP) [7], Multicast Open Shorted Path First (MOSPF) [23], Core Based Trees (CBT) [2], and Protocol Independent Multicast (PIM) [8]) do not perform well in ad hoc networks because multicast tree structures are fragile and must be readjusted as connectivity changes. Furthermore, multicast trees usually require a global routing substructure such as link state or distance vector. The frequent exchange of routing vectors or link state tables, triggered by continuous topology changes, yields excessive channel and processing overhead.

Various multicast protocols have been newly proposed to perform multicasting in ad hoc networks. However, no performance study between them has yet been performed. The comparative analysis of ad hoc *unicast* routing protocols has been reported in [4], [6], [14], [20], but this paper is the first to conduct a performance comparison study of ad hoc wireless *multicast* protocols in a realistic common simulation environment. By using a detailed simulator, we provide quantitative performance analysis of five protocols with different characteristics: AMRoute [3], ODMRP [17], [18], [19], AMRIS [29], CAMP [10], [11], [21], [22], and flooding. In addition to multicast routing protocols, we also implemented a detailed and realistic model of physical and medium access control layer protocols.

The five multicast routing protocols were simulated in diverse network scenarios. We studied the impact of mobility on performance by varying the speed of network hosts. We varied the number of data packet senders to emulate a variety of different multicast applications. One source to many receivers can correspond to battlefield data dissemination. Many sources to many receivers can correspond to search and rescue team communication. Different multicast group member sizes were simulated to investigate the impact on performance. Various traffic loads were also applied to study how traffic patterns influence multicast protocol performance. We apply metrics that show the "efficiency" in addition to the "effectiveness" of the protocols. Understanding the protocol's efficiency gives us the ability to study and discuss relative strengths, weaknesses, and applicability to various situations for each multicast routing protocol.

The rest of the paper is organized as follows. Section II presents an overview of the multicast protocols we simulate. The simulation environment and methodology are described in Section III, followed by simulation results in Section IV. Merits and shortcomings of each multicasting algorithm are discussed in Section V and concluding remarks are made in Section VI.

II. Multicast Protocols Review

In this section, we introduce the ad hoc wireless multicast protocols we have selected. Basic operation procedures and implementation choices are described.

A. Adhoc Multicast Routing (AMRoute)

AMRoute [3] is a tree based protocol. It creates a bidirectional shared multicast tree using unicast tunnels to provide connections between multicast group members. Each group has at least one logical core that is responsible for member and tree maintenance. Initially, each group member declares itself as

This work was funded in part by the Defense Advanced Research Projects Agency (DARPA) under contract DAAB07-97-C-D321, as a part of the Global Mobile Information Systems (GloMo) program.

TABLE I

PARAMETER VALUES FOR AMROUTE

Periodic JOIN-REQ interval	60 sec
Periodic JOIN-REQ interval when no group members are connected to the core	5 sec
Periodic TREE-CREATE interval	20 sec
TREE-CREATE timeout	40 sec
Core resolution algorithm	Highest ID

a core for its own group of size one. Each core periodically floods JOIN-REQS (using an expanding ring search) to discover other disjoint mesh segments for the group. When a member node receives a JOIN-REQ from a core of the same group but a different mesh segment, it replies with a JOIN-ACK and marks that node as a mesh neighbor. The node that receives a JOIN-ACK also marks the sender of the packet as its mesh neighbor. After the mesh creation, each core periodically transmits TREE-CREATE packets to mesh neighbors in order to build a shared tree. When a member node receives a non-duplicate TREE-CREATE from one of its mesh links, it forwards the packet to all other mesh links. If a duplicate TREE-CREATE is received, a TREE-CREATE-NAK is sent back along the incoming link. The node receiving a TREE-CREATE-NAK marks the link as mesh link instead of tree link. The nodes wishing to leave the group send the JOIN-NAK to the neighbors and do not forward any data packets for the group.

The key characteristic of AMRoute is its usage of virtual mesh links to establish the multicast tree. Therefore, as long as routes between tree members exist via mesh links, the tree need not be readjusted when network topology changes. Non-members do not forward data packets and need not support any multicast protocol. Thus, only the member nodes that form the tree incurs processing and storage overhead. AMRoute relies on an underlying unicast protocol to maintain connectivity among member nodes and any unicast protocol can be used. The major disadvantage of the protocol is that it suffers from temporary loops and creates non-optimal trees when mobility is present.

Table I shows the AMRoute parameter values used in our experiments. The implementation followed the specification in [3].

B. *On-Demand Multicast Routing Protocol (ODMRP)*

ODMRP [17], [18], [19] creates a mesh of nodes (the "forwarding group") which forward multicast packets via flooding (within the mesh), thus providing path redundancy. ODMRP is an on-demand protocol, thus it does not maintain route information permanently. It uses a soft state approach in group maintenance. Member nodes are refreshed as needed and do not send explicit leave messages.

In ODMRP, group membership and multicast routes are established and updated by the source on demand. Similar to on-demand unicast routing protocols, a request phase and a reply phase comprise the protocol. When multicast sources have data to send, but do not have routing or membership information, they flood a JOIN DATA packet. When a node receives a non-duplicate JOIN DATA, it stores the upstream node ID (i.e., backward learning) and rebroadcasts the packet. When the JOIN DATA packet reaches a multicast receiver, the receiver creates a JOIN TABLES and broadcasted to the neighbors. When a node receives a JOIN TABLE, it checks if the next node ID of one of the entries matches its own ID. If it does, the node realizes that it is on the path to the source and thus is part of the forwarding group. It then broadcasts its own JOIN TABLE built upon matched entries. The JOIN TABLE is thus propagated by each forwarding group member until it reaches the multicast source via the shortest path. This process constructs (or updates) the routes from sources to receivers and builds a mesh of nodes, the *forwarding group*. Multicast senders refresh the membership information and update the routes by sending JOIN DATA periodically.

TABLE II

PARAMETER VALUES FOR ODMRP

JOIN DATA refresh interval	3 sec
Acknowledgment timeout for JOIN TABLE	25 msec
Maximum JOIN TABLE retransmission	3

In networks where GPS (Global Positioning System) [16] is available, ODMRP can be made adaptive to node movements by utilizing mobility prediction [19]. By using location and mobility information supported by GPS, route expiration time can be estimated and receivers can select the path that will remain valid for the longest time. With the mobility prediction method, sources can reconstruct routes in anticipation of route breaks. This way, the protocol becomes more resilient to mobility. The price is, of course, the cost and additional weight of GPS. The details of mobility prediction and the procedure are described in [19].

The data transfer phase is identical for both versions. Nodes forward the data if they are forwarding nodes and the packet they receive is not a duplicate. Since all forwarding nodes relay data, redundant paths (when they exist) can help deliver data when the primary path becomes disconnected because of mobility. Another unique property of ODMRP is its unicast capability. Not only can ODMRP coexist with any unicast routing protocol, it can also operate very efficiently as unicast routing protocol. Thus, a network equipped with ODMRP does not require a separate unicast protocol.

The specification in [19] was used in our implementation. For consistency with comparison, we used the version *without* mobility prediction. ODMRP parameter values used are shown in Table II.

C. *Ad hoc Multicast Routing protocol utilizing Increasing id-numberS (AMRIS)*

AMRIS [29] establishes a shared tree for multicast data forwarding. Each node in the network is assigned a multicast session ID number. The ranking order of ID numbers is used to direct the flow of multicast data. Like ODMRP, AMRIS does not require a separate unicast routing protocol.

Initially, a special node called Sid broadcasts a NEW-SESSION packet. The NEW-SESSION includes the Sid's msm-id (multicast session member id). Neighbor nodes, upon receiving

TABLE III
PARAMETER VALUES FOR AMRIS

Periodic beacon interval	1 sec
Max allowed beacon losses	3
NEW SESSION lifetime	3 sec
Acknowledgment timeout for JOIN-REQ	2 sec
Random broadcast jitter time	50 msec

the packet, calculate their own msm-ids which are larger than the one specified in the packet. The msm-ids thus increase as they radiate from the Sid. The nodes rebroadcast the NEW-SESSION message with the msm-id replaced by their own msm-ids. Each node is required to broadcast beacons to its neighbors. The beacon message contains the node id, msm-id, membership status, registered parent and child's ids and their msm-ids, and partition id. A node can join a multicast session by sending a JOIN-REQ. This JOIN-REQ is unicasted to a potential parent node with a smaller msm-id than the node's msm-id. The node receiving the JOIN-REQ sends back a JOIN-ACK if it already is a member of the multicast session. Otherwise, it sends a JOIN-REQ.PASSIVE to its potential parent. If a node fails to receive a JOIN-ACK or receives a JOIN-NAK after sending a JOIN-REQ, it performs "Branch Reconstruction (BR)." The BR process is executed in an expanding ring search until the node succeeds in joining the multicast session.

AMRIS detects link disconnection by a beaconing mechanism. If no beacons are heard for a predefined interval of time, the node considers the neighbor to have moved out of radio range. If the former neighbor is a parent, the node must rejoin the tree by sending a JOIN-REQ to a new potential parent. If the node fails to join the session or no qualified neighbors exist, it performs the BR process.

Data forwarding in done by the nodes in the tree. Only the packets from the registered parent or registered child are forwarded. Hence, if the tree link breaks, the packets are lost until the tree is reconfigured.

Our AMRIS implementation followed the specification in [29]. The AMRIS parameter values are shown in Table III.

D. Core-Assisted Mesh Protocol (CAMP)

CAMP [10], [11], [21], [22] supports multicasting by creating a shared mesh structure. All nodes in the network maintain a set of tables with membership and routing information. Moreover, all member nodes maintain a set of caches that contain previously seen data packet information and unacknowledged membership requests. CAMP classifies nodes in the network as duplex or simplex members, or non-members. Duplex members are full members of the multicast mesh, while simplex members are used to create one-way connections between sender-only nodes and the rest of the multicast mesh. "Cores" are used to limit the flow of JOIN REQUEST packets.

CAMP consists of mesh creation and maintenance procedures. A node wishing to join a multicast mesh first consults a table to determine whether it has neighbors which are already members of the mesh. If so, the node announces its membership via a CAMP UPDATE. Otherwise, the node either propagates a JOIN REQUEST towards one of the multicast group "cores," or attempts to reach a member router by an expanding ring search of broadcast requests. Any duplex member of the node can respond with a JOIN ACK, which is propagated back to the source of the request.

TABLE IV
PARAMETER VALUES FOR CAMP

Number of cores in the network	1
Periodic beacon interval	3 sec
Periodic update interval	3 sec
Age out anchor timeout	45 sec
Heartbeat interval	15 sec
Request retransmission interval	9 sec
Max number of JOIN REQUEST retransmission	3

Periodically, a receiver node reviews its packet cache in order to determine whether it is receiving data packets from those neighbors which are on the reverse shortest path to the source. If not, the node sends either a HEARTBEAT or a PUSH JOIN message towards the source along the reverse shortest path. This process ensures that the mesh contains all such reverse shortest paths from all receivers to all senders. The nodes also periodically choose and refresh their selected "anchors" to the multicast mesh by broadcasting updates. These anchors are neighbor nodes which are required to re-broadcast any non-duplicate data packets they receive. A node is allowed to discontinue anchoring neighbor nodes which are not refreshing their connections. It can then leave the multicast mesh if it is not interested in the multicast session and is not required as anchor for any neighboring node.

CAMP relies on an underlying unicast routing protocol which guarantees correct distances to all destinations within finite time. Routing protocols that are based on the Bellman-Ford algorithm cannot be used with CAMP, and CAMP needs to be extended in order to work with on-demand routing protocols.

Our implementation of CAMP followed the specification in [10]. Table IV shows the CAMP parameter values used in our simulation. Periodic beacon interval is 3 seconds, but the beacon is sent only when no packet has been transmitted during the beacon interval.

E. Protocol Summary

Table V summarizes key characteristics and properties of the protocols we simulated. Note that ODMRP requires periodic

TABLE V
SUMMARY OF PROTOCOLS

Protocols	**AMRoute**	**ODMRP**	**AMRIS**	**CAMP**	**Flood**
Configuration	Tree	Mesh	Tree	Mesh	Mesh
Loop-Free	No	Yes	Yes	Yes	Yes
Dependency on Unicast Protocol	Yes	No	No	Yes	No
Periodic Messaging	Yes	Yes	Yes	Yes	No
Control Packet Flood	Yes	Yes	Yes	No	No

messaging (JOIN DATA) only when sources have data packets to send.

III. SIMULATION MODEL AND METHODOLOGY

The simulator for evaluating multicasting protocols was implemented within the GloMoSim library [28]. The GloMoSim library is a scalable simulation environment for wireless network systems using the parallel discrete-event simulation capability provided by PARSEC [1]. Our simulation modeled a network of 50 mobile hosts placed randomly within a $1000m \times 1000m$ area. Radio propagation range for each node was 250 meters and channel capacity was 2 Mbits/sec. There were no network partitions throughout the simulation and the average number of neighbors for each node was 6.82. Each simulation executed for 600 seconds of simulation time. Multiple runs with different seed numbers were conducted for each scenario and collected data was averaged over those runs.

A. Channel and Radio Model

A free space propagation model [26] with a threshold cutoff was used in our experiments. In the free space model, the power of a signal attenuates as $1/d^2$ where d is the distance between radios. In addition to the free space channel model, we also implemented SIRCIM (Simulation of Indoor Radio Channel Impulse-response Models) [27] which considers multipath fading, shadowing, barriers, foliages, etc. SIRCIM is more accurate than the free space model, but we decided against using SIRCIM in our study because: (a) the complexity of SIRCIM increases simulation time by three orders of magnitude; (b) the accuracy of the channel model does not affect the relative ranking of the multicasting protocols evaluated in this study; and (c) SIRCIM must be "tuned" to the characteristics of the physical environment (e.g., furniture, partitions, etc.), thus requiring a much more specific scenario than we are assuming in our experiments.

In the radio model, we assumed the ability of a radio to lock onto a sufficiently strong signal in the presence of interfering signals, i.e., radio capture. If the capture ratio (the ratio of an arriving packet's signal strength over the sum of all colliding packets) [26] was greater than a predefined threshold value, the packet was received while all other interfering packets were dropped.

B. Medium Access Control Protocol

The IEEE 802.11 MAC with Distributed Coordination Function (DCF) [12] was used as the MAC protocol. DCF is the mode which allows mobiles to share the wireless channel in an ad hoc configuration. The specific access scheme is Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) with acknowledgments. Optionally, the nodes can make use of Request To Send/Clear To Send (RTS/CTS) channel reservation control frames for unicast, virtual carrier sense, and fragmentation of packets larger than a given threshold. By setting timers based upon the reservations in RTS/CTS packets, the virtual carrier sense augments the physical carrier sense in determining when mobile nodes perceive that the medium is busy. Fragmentation is useful in the presence of high bit error and loss rates, as it reduces the size of the data units that need to be retransmitted.

In our experiments, we employed RTS/CTS exclusively for unicast control packets directed to specific neighbors (e.g., replies). All other transmissions use CSMA/CA. We chose this configuration to minimize the frequency and deleterious effects of collisions over the wireless medium. We did not employ fragmentation because our data packets were small enough that the additional overhead would reduce overall network throughput.

C. Multicast Protocols

When implementing the multicast protocols, we followed the specifications of each protocol as defined in the published literature. We directly queried the protocol designers about details which were not specified in the publications (e.g., various timer values, core selection algorithm, etc.). ODMRP and AMRIS do not require underlying unicast protocol to operate, but AMRoute and CAMP do. While AMRoute can work with any protocol, the designers of CAMP specifically state that it can operate only with certain unicast protocols [10]. We have implemented one of those protocols, WRP [24], a distance-vector based unicast routing protocol developed by the same group which developed CAMP. For a fair comparison, WRP was used as the underlying unicast protocol also for AMRoute. The source code of each protocol was independently validated by two of the authors.

D. Traffic Pattern

A traffic generator was developed to simulate constant bit rate sources. The size of data payload was 512 bytes. The senders were chosen randomly among multicast members who in turn were chosen with uniform probability among 50 network hosts. The member nodes join the multicast session at the beginning of the simulation and remain as members throughout the simulation. Hence, the simulation experiments do not test/account for the overhead produced in the session leave process.

E. Metrics

We have used the following metrics in comparing protocol performance. Some of these metrics were suggested by the IETF MANET working group for routing/multicasting protocol evaluation [5].

- **Packet delivery ratio**: The ratio of the number of data packets actually delivered to the destinations versus the number of data packets supposed to be received. This number presents the effectiveness of a protocol.
- **Number of data packets transmitted per data packet delivered**: 'Data packets transmitted' is the count of every individual transmission of data by each node over the entire network. This count includes transmissions of packets that are eventually dropped and retransmitted by intermediate nodes. Note that in unicast protocols, this measure is always equal or greater than one. In multicast, since a single transmission can deliver data to multiple destinations, the measure may be less than one.
- **Number of control bytes transmitted per data bytes delivered**: Instead of using a measure of pure control overhead, we chose to use the ratio of control bytes transmitted to data bytes delivered to investigate how efficiently control packets are utilized in delivering data. Note that not only bytes of control packets (e.g., beacons, route updates, join requests, acknowledgments, etc.), but also bytes of data packet headers are included

in the number of control bytes transmitted. Accordingly, only the data payload bytes contribute to the data bytes delivered.

- **Number of control and data packets transmitted per data packet delivered**: This measure shows the efficiency in terms of channel access and is very important in ad hoc networks since link layer protocols are typically contention-based.

IV. Simulation Results

A. *Mobility Speed*

A.1 Scenarios

Each node moved constantly with the predefined speed. Moving directions of each node were selected randomly, and when nodes reached the simulation terrain boundary, they bounced back and continued to move. The node movement speed was varied from 0 km/hr to 72 km/hr. In the mobility experiment, 20 nodes are multicast members and 5 sources transmit packets at the rate of 2 pkt/sec each.

A.2 Results and Analysis

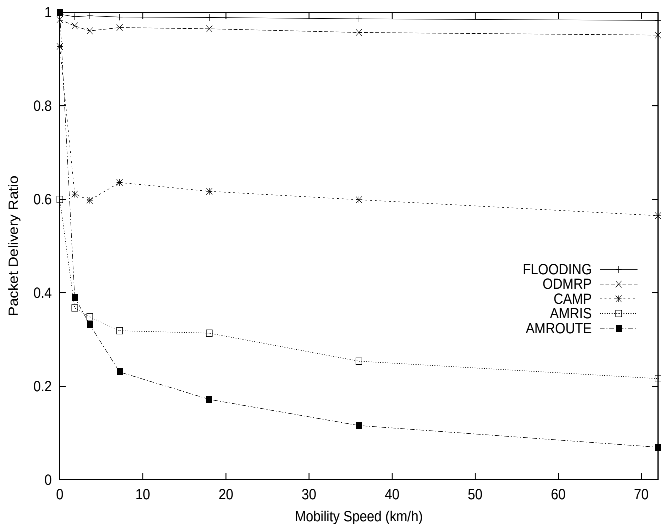

Fig. 1. Packet Delivery Ratio as a Function of Mobility Speed.

Fig. 1 illustrates the packet delivery ratio of the protocols under different speeds. ODMRP shows good performance even in highly dynamic situations. ODMRP provides redundant routes with a mesh topology and the chances of packet delivery to destinations remain high even when the primary routes are unavailable. The path redundancy enables ODMRP to suffer only minimal data loss and be robust to mobility. In fact, ODMRP was as effective as flooding in this experiment.

CAMP, which also uses a mesh topology, shows a better performance than protocols which use trees. However, CAMP exhibited poorer performance than we had expected, especially under mobility. A major reason CAMP was not as effective as ODMRP was that many packets headed to distant routers in the mesh were not delivered. In CAMP, since the paths to distant destinations have fewer redundant paths than those closer to the center of the mesh, they are more prone to occasional link breaks preventing a vital "anchoring" node from successfully receiving packets. Most of the successful packet transmissions occur in this mesh center, and require fewer data transmissions per delivery than transmissions to the mesh edges. In addition, in the presence of mobility and link breaks, WRP (which is the unicast protocol CAMP prefers to coexist with) can require a period of network re-convergence in regards to a subset of destinations. During this interval, this subset of destinations will be marked as unreachable by the loop-detection facilities. If the group core is a part of this subset of temporarily unreachable nodes, the multicast routing updates regarding mesh maintenance will be postponed, which also contributes to delays in mesh response to mobility.

AMRIS shows a poor delivery ratio compared to protocols that use mesh configuration. Since AMRIS builds a shared tree for data dissemination, there is only one path between member nodes. If a single tree link breaks because of node movements, packet collision, or congestion, destinations can not receive packets. AMRIS detects node movements and tree breaks by a beaconing mechanism. Nodes send beacons every second, and neighbors are considered to have moved away if 3 consecutive beacons are not received. Thus, in the best case, it takes 3 seconds after the link break for AMRIS to start tree readjustment. A number of packets can be lost during that period. There are possible solutions to this problem, but they all have respective drawbacks. If beacons are sent more often, that could increase packet collisions. If the number of allowed beacon losses is decremented, a node may attempt to find a new route when the link is not broken but beacons are lost due to collisions. Finding the optimal beacon interval and allowed number of beacon losses for AMRIS is beyond the scope of the paper and we used the values recommended by the AMRIS designers. The result that surprised us was for zero mobility. While other protocols showed data delivery ratio approaching unity, AMRIS delivered only 60% of data packets. Since each node sends beacons every second, there are a number of packets contending for the channel. The beacon size of AMRIS is relatively large compared to other protocols that send beacons (see [29]). Thus, the beacon traffic combined with the data traffic causes a large number of collisions leading to 40% drop. Under very light data traffic, AMRIS shows improved performance as will be shown in Fig. 8.

AMRoute was the least effective of the protocols with mobility. Although its delivery ratio is near perfect in no mobility, it fails to deliver a significant number of packets even at low mobility speeds. The delivery ratio steadily worsens as the mobility speed is increased. One of the reasons AMRoute performs so poorly is due to the formation of loops and the creation of sub-optimal trees when mobility is present (at 72 km/hr, the average hop count was nearly 8 while other protocols were below 4). Loops occur during the tree reconstruction phase when some nodes are forwarding data according to the stale tree and others according to the newly built tree. The existence of loops is critical in protocol performance because they cause serious congestion. At some instants, nodes had up to 13.75 packets dropped per second. The loss of packets due to buffer overflow has two consequences. First, if a data packet is dropped in the early stage of its multicast tree traversal, a large portion of tree members will not receive it. Second, if control packets (Tree-

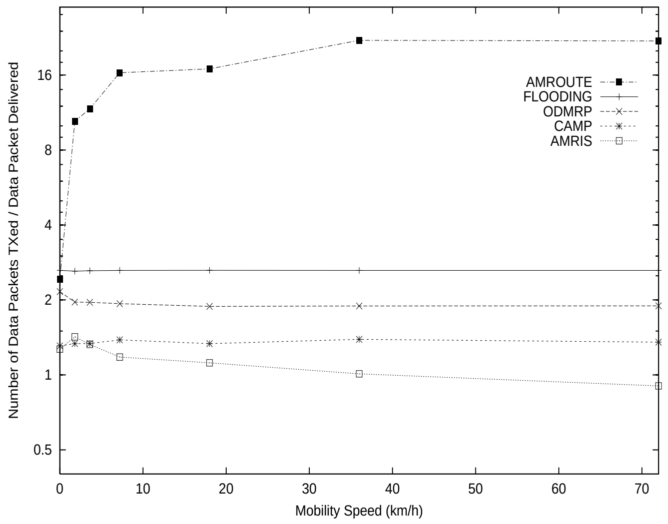

Fig. 2. Number of Data Packets Transmitted per Data Packet Delivered as a Function of Mobility Speed.

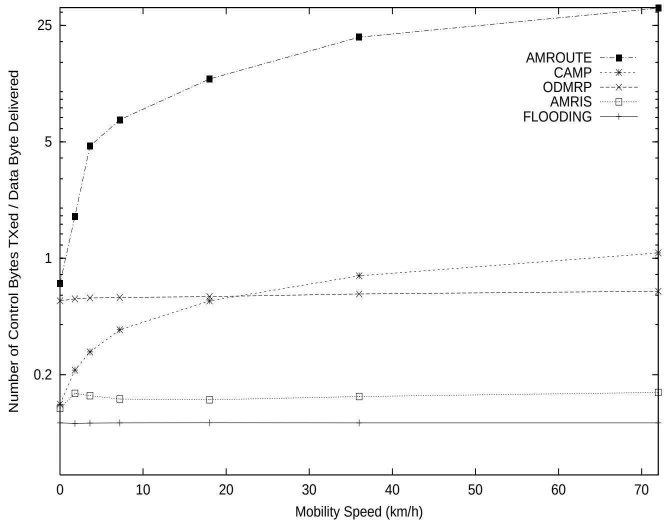

Fig. 3. Number of Control Bytes Transmitted per Data Byte Delivered as a Function of Mobility Speed.

CREATE, JOIN-ACK, etc.) are dropped, the tree is not properly built or becomes segmented and data will not be delivered. Another reason for AMRoute ineffectiveness is its dependency on the underlying unicast protocol. AMRoute relies on the unicast protocol to set up bidirectional tunnels between group members for the multicast tree. However, as shown in [25], when mobility speed increases, the bidirectional link assumption in ad hoc networks becomes weak (i.e., a node can reach a neighboring node, but not necessarily vice versa). In our experiments, unidirectional "critical" links existed in AMRoute trees. Critical links are such that packets sent by the one end of the link are mostly received by the other end but not vice versa. A great number of packets are lost at these critical links. Since there are no alternate routes in the AMRoute shared tree (although AMRoute creates the mesh in order to build a tree, data is forwarded only by tree nodes), data delivery ratio is very low.

Fig. 2 shows the number of data transmissions per data delivery to destinations. AMRoute has the highest number of transmissions because of loops. We can observe that protocols using meshes (i.e., ODMRP and CAMP) transmit more data packets than AMRIS, which uses a tree. In fact, ODMRP transmits nearly as much data as flooding because it exploits multiple redundant routes for data delivery.

The control byte overhead per data byte delivered is shown in Fig 3. Remember that data packet header is included in control overhead. Flooding has no control packets. Hence, only the data header contributes to control overhead and this overhead does not increase with mobility. Other protocols generate increasing overhead as speed increases. AMRIS shows a low control overhead compared to other multicast schemes. The primary reason is that it transmitted less data packets (as seen in Fig. 2). CAMP shows a larger control overhead under high mobility than ODMRP because of its reliance on the unicast routing protocol WRP, which sends triggered updates. WRP suffers from exponential growth in control traffic overhead under increasing mobility. Moreover, CAMP piggybacks its own update

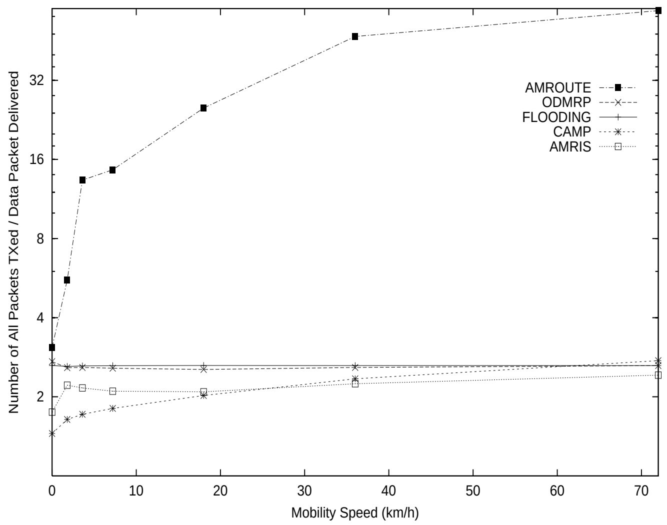

Fig. 4. Number of Total Packets Transmitted per Data Packet Delivered as a Function of Mobility Speed.

messages onto WRP updates and those packets play a role in overhead growth. In ODMRP, the control overhead remains relatively constant because no updates are triggered by mobility. JOIN DATA refresh interval was set constant to 3 seconds and hence no additional overhead is required as mobility increases. AMRoute has the highest ratio because of the data headers that caught in the loops. The high ratio is also due to the formation of inefficient trees. During the tree creation phase, an inefficient tree can be formed when the TREE-CREATE packets from distant mesh neighbors arrives earlier than packets from nearby nodes (e.g., due to network congestion, etc.). The non-optimal tree results in having longer hops between member nodes and increasing the number of data transmissions.

The number of all packets transmitted per data packet delivered is presented in Fig. 4. An interesting result is that CAMP

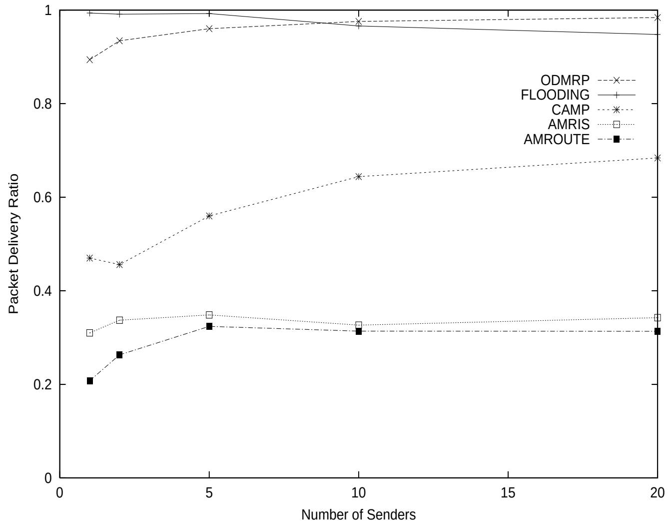

Fig. 5. Packet Delivery Ratio as a Function of Number of Senders.

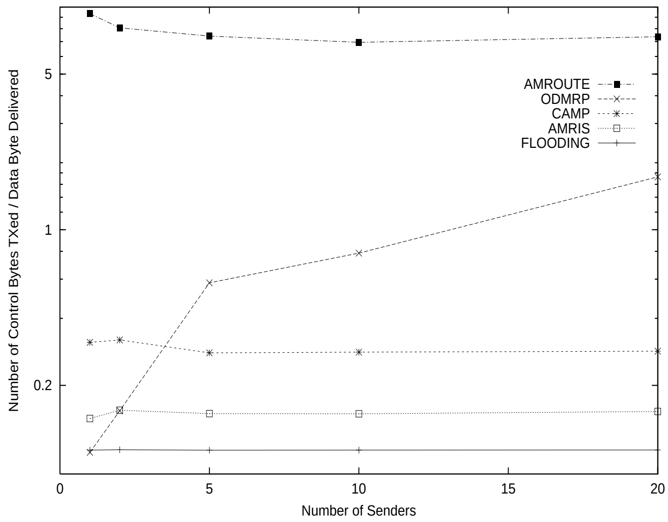

Fig. 6. Number of Control Bytes Transmitted per Data Byte Delivered as a Function of Number of Senders.

has a smaller number of transmissions than ODMRP. This result stems from two factors. First, ODMRP transmits more data packets on redundant paths than CAMP. Second, although CAMP has more control overhead bytes, the number of control packet transmissions is lower since CAMP updates are piggybacked onto WRP updates. Again, AMRIS has the smallest number of packet transmissions because it uses a tree and AMRoute has the highest value because of loops.

B. Number of Senders

B.1 Scenarios

In this experiment, the multicast group size is set constant at 20, node mobility speed is slow (1 m/s), and network traffic load is relatively light (10 pkt/sec). The number of multicast senders range in the set $\{1, 2, 5, 10, 20\}$. A single sender represents a class lecture scenario, while at the other extreme, 20 senders model a video conference situation.

B.2 Results and Analysis

The packet delivery ratio as a function of the number of multicast senders is shown in Fig. 5. As the number of sources increases, performance of flooding slightly degrades as more packets are lost by collision, congestion, and channel contention. ODMRP shows robustness to the number of sources. In fact, performance even improves with number of senders because of increasing number of forwarding nodes and thus better path redundancy. ODMRP limits the number of sources that can send JOIN DATA at the same time. Whenever a source needs to flood a JOIN DATA, it listens if any other source is flooding the packet. It proceeds to send the JOIN DATA only if no flooded packets are received within a certain period. Thus, the number of collisions decreases and the the protocol remains effective. Like ODMRP, CAMP shows improved performance with a larger number of senders due to the increase in the number of anchors that each node requires. Each member node requests every neighbor which is in the reverse shortest path to some source, to rebroadcast multicast update packets it receives initially. Hence increasing the number of sources increases the redundant paths in the mesh. AMRIS and AMRoute performance was unaffected by the number of senders because they use a shared tree for the multicast session.

Fig. 6 shows the control overhead per data byte delivered. Every protocol except ODMRP shows a constant value. While the other three multicast protocols form a shared mesh or tree, ODMRP builds per-source meshes. If the number of senders increases, more JOIN DATA packets are propagated and control overhead grows accordingly. We can speculate from this result that ODMRP in its present form may not be as efficient in networks where a large number of nodes (e.g., hundreds and thousands) are multicast sources.

C. Multicast Group Size

C.1 Scenarios

We varied the number of multicast members to investigate the scalability of the protocol. While fixing the number of senders at 5, mobility speed at 1 m/s, and network traffic rate at 10 pkt/sec, the multicast group size was varied from 5 to 40 members.

C.2 Results and Analysis

The routing effectiveness of protocols as a function of multicast group size is illustrated in Fig. 7. Flooding and ODMRP performance were not affected by the number of multicast members. CAMP, on the other hand, performs markedly better as the number of receivers increases. Since the mesh becomes massive with the growth of the members, more redundant routes are formed and that improves the performance. If only a small number of nodes join the multicast session, the mesh actually appears closer to a tree for distant nodes, and the performance is reflected in this graph. AMRIS also shows improvements with the member size growth, but they are less dramatic than CAMP because redundant routes are not established in AMRIS. AMRoute shows the complete opposite behavior. As the group size

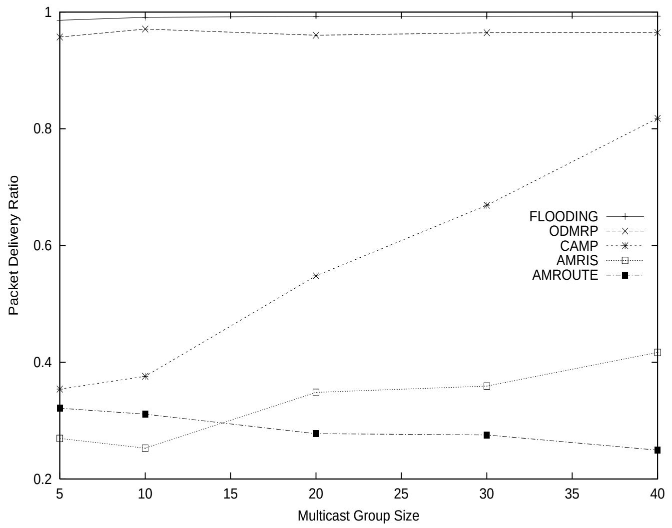

Fig. 7. Packet Delivery Ratio as a Function of Multicast Group Size.

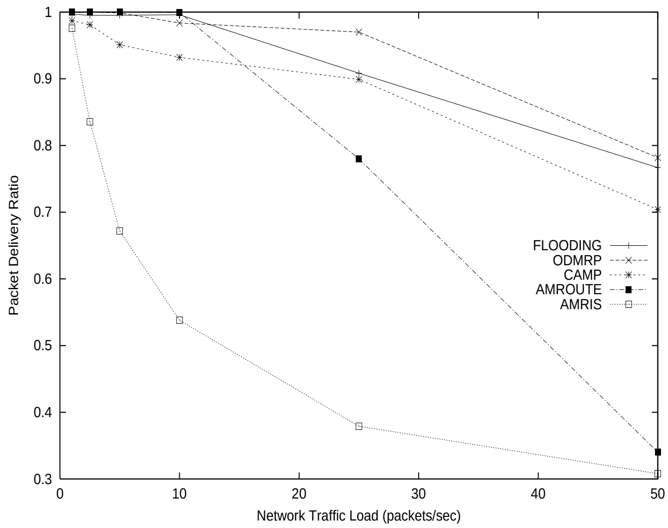

Fig. 8. Packet Delivery Ratio as a Function of Network Traffic Load.

increases, the delivery ratio actually drops. This behavior is due to the "critical" links that exist in the AMRoute multicast tree (critical links were described in section IV-A). As the group size increases, the number of tree links increases and the probability of sources being isolated in the tree by critical links increases as well.

D. Network Traffic Load

D.1 Scenarios

To study the impact of data traffic load on multicast protocols, we varied the load on the network. There were 5 senders and the multicast group size was 20. In this experiment, there was no node mobility. Therefore, the packet drops are only caused by buffer overflow, collision, and congestion. The network traffic loads used were between 1 pkt/sec and 50 pkt/sec.

D.2 Results and Analysis

Packet delivery ratios for various traffic loads are shown in Fig. 8. AMRIS was the most sensitive to traffic load. AMRIS delivers a high percentage of data packets in extremely light load (i.e., less than 5 pkt/sec). As the load increases however, the ratio drops rapidly. As explained in section IV-A, the transmission and the size of beacons resulted in numerous packet collisions. AMRoute performance is nearly perfect when the packet rate is relatively low, but it drops rather quickly when the traffic load is increased. The degradation is caused by buffer overflow at the members in the tree and at the mesh nodes that connect the tree members. CAMP performance is also affected by traffic load. As the load increases, the number of collisions and packet losses increase. When important control packets are dropped, anchor construction can be delayed and data packets can fail to reach all the anchors. The degradation follows a pattern similar to flooding and ODMRP, indicating a common behavior in mesh based data delivery. Flooding shows worse delivery ratios than ODMRP as load grows. Since every data packet is flooded, the number of collisions and buffer overflows grows with the load. ODMRP is also affected by load, but the packet loss rate is less severe than flooding because the number of data packet transmissions is less than flooding. Although ODMRP shows the same patters of behaviors as CAMP, it gives a better delivery rate because it has less control overhead and suffers less buffer overflows than CAMP.

V. Discussion

In previous sections, we have studied the effectiveness and efficiency of several multicast protocols. In this section, we summarize the merits and shortcomings of protocols and derive suggestions for improvements. We also explain why our results differ from previous works by other researchers for some of the protocols. Finally, we share some of the lessons we have learned while conducting the study.

A. Protocol Analysis

AMRoute showed some promise in its simplicity and scalability in the number of senders. However, the presence of unidirectional "critical" links prevented reliable data delivery. The problem became worse as mobility was increased. Other drawbacks of AMRoute were the existence of loops and inefficient formation of trees. A possible improvement for AMRoute is to take reachability information (i.e., packets sent to a neighbor/packets received from that neighbor) into account when selecting tree links. Using this method, the impact of unidirectional critical links can be reduced. In addition, introducing adaptivity into the protocol (e.g., periodic TREE-CREATE interval) can build more optimal trees. Most importantly, a loop prevention mechanism must be utilized for AMRoute to be efficient.

ODMRP performed well in most of our experiments. Providing redundant paths by the formation of mesh configuration made the protocol robust to mobility. The protocol did not yield excessive overhead in high mobility scenarios because no control packets are triggered by link breaks. However, when there are a large number of multicast senders, the protocol may suffer from excessive control overhead. Enhancements to make the protocol more scalable to large member groups must be developed.

AMRIS performance was very sensitive to mobility and traffic load. The main reasons for the poor performance were the number of transmissions and the size of beacons. As shown in section IV-D, beacons caused a number of packet collisions even when nodes are stationary. In more dense networks, the performance may become worse. We believe AMRIS can be improved by using a beaconing mechanism similar to CAMP. If the beacon is sent only when no packet has been transmitted in given interval, the number of beacon transmissions can be reduced while still delivering node information to neighboring nodes. In addition, the selection of Sid can affect the shape of the tree and possibly its performance. The research into the Sid selection algorithm along with beaconing methods will help improve AMRIS.

CAMP has good control traffic scalability for increasing multicast group size. Since JOIN REQUESTS only propagate until they reach a mesh member, CAMP does not incur exponential growth of multicast updates as the number of nodes and group members increase. However, it is dependent upon the unicast routing protocol for behaviors regarding network convergence and control traffic growth in the presence of mobility. WRP's response to link breaks is not immediate, and can incorrectly deduce a link break in the presence of high network load. CAMP may perform better if it is modified so as to operate with an on-demand routing protocol. As shown in [4], [6], [14], [20], on-demand protocols performed favorably in terms of control packet overhead and response to mobility. If CAMP were able to leverage these advantages, it should dramatically improve its packet delivery ratio and control overhead.

B. Related Works

As of October 1999, only CAMP and ODMRP designers have performed simulation study of their protocols. AMRoute and AMRIS performance evaluation have not been published. In simulation works reported in [10], [21], [22], the results are quite different from the results we have obtained in our experiments. In [10], [21], [22], a simplified simulator was used. A perfect channel was assumed and radio propagation was not considered. FAMA [9] was used as the medium access control protocol, which is different from IEEE 802.11 [12], the emerging standard MAC protocol for wireless LAN, that we used in our simulation. Only a small portion of network hosts had mobility (5 out of 30 or 15 out of 30) in their study. The critical nodes for CAMP performance (e.g., core, senders), however, remained stationary. All the nodes in [10], [21], [22] were multicast session members, which is not realistic in typical multicast applications. The network traffic load was extremely light (4 packets/sec). Information on data size, radio propagation range, or simulation terrain range were not given. Thus, the results in [10], [21], [22] are somewhat limited. In any way, they cannot be directly compared to the results from this paper.

C. Lessons Learned

While implementing and evaluating multicast protocols, we have learned a great deal and would like to share our experience with researchers who design and implement ad hoc wireless multicast protocols. In our study, the mesh protocols performed significantly better than the tree protocols in mobile scenarios. In trees, when routes are invalidated due to node movements, the packets must be buffered or dropped until the tree is reconfigured. On the other hand, redundant routes in the mesh provide alternate routes for data delivery in the face of mobility and link breaks. Data packets can still reach the destinations while the primary route is being reconstructed.

Using detailed lower layer (i.e., link layer and physical layer) implementations in the network simulator along with programmable mobility patterns highlighted differences in the protocol tolerance to various wireless network conditions. We strongly recommend fellow researchers to use publicly available simulators which are validated by frequent use and which permit replication of the experiments.

VI. CONCLUSIONS

We have conducted a performance evaluation of five multicast protocols that have been proposed for ad hoc networks. The channel, radio, IEEE 802.11 MAC protocol, and multicast protocols (AMRoute, ODMRP, AMRIS, CAMP, and flooding) have been carefully implemented. The detailed simulator has enabled us to perform fair and accurate comparisons of the multicast protocols under a realistic wireless environment, for a broad range of parameters including mobility, number of senders, multicast group size, and traffic load.

A general conclusion is that, in a mobile scenario, mesh-based protocols outperformed tree-based protocols. The availability of alternate routes provided robustness to mobility. AMRoute performed well under no mobility, but it suffered from loops and inefficient trees even for low mobility. AMRIS was effective in a light traffic environment with no mobility, but its performance was susceptible to traffic load and mobility. CAMP showed better performance when compared to tree protocols, but with mobility, excessive control overhead caused congestion and collisions that resulted in performance degradation. ODMRP was very effective and efficient in most of our simulation scenarios. However, the protocol showed a trend of rapidly increasing overhead as the number of senders increased.

We experimented with scenarios which we thought were the most representation of ad hoc wireless network applications. However, we did not cover every possible situation. While the results of this paper can provide guidelines, the final selection of a multicast protocol should take into account other considerations which cannot be evaluated via simulation alone.

ACKNOWLEDGMENTS

We thank Rajesh Talpade of Bellcore, Chun Wei Wu of University of Singapore, and Ewerton Madruga of University of California, Santa Cruz for answering our questions about protocols details of AMRoute, AMRIS, and CAMP. We also thank Evan Tsang, Yung-Szu Tu, and Brent Murata of University of California, Los Angeles for helping the simulation process. We are also grateful of Ronn Ritke and James Stepanek of University of California, Los Angeles for helpful comments and suggestions.

REFERENCES

[1] R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, and H.Y. Song, "PARSEC: A Parallel Simulation Environment for Complex Systems," *IEEE Computer*, vol. 31, no. 10, Oct. 1998, pp.77-85.

[2] T. Ballardie, P. Francis, and J. Crowcroft, "Core Based Trees (CBT) - An Architecture for Scalable Inter-Domain Multicast Routing," In *Proceedings of ACM SIGCOMM'93*, San Francisco, CA, Oct. 1993, pp. 85-95.

[3] E. Bommaiah, M. Liu, A. McAuley, and R. Talpade, "AMRoute: Adhoc Multicast Routing Protocol," *Internet-Draft*, draft-talpade-manet-amroute-00.txt, Aug. 1998, Work in progress.

[4] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," In *Proceedings of ACM/IEEE MOBICOM'98*, Dallas, TX, Oct. 1998, pp. 85-97.

[5] M.S. Corson and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," *Request For Comments 2501*, Internet Engineering Task Force, Jan. 1999.

[6] S.R. Das, R. Castaneda, J. Yan, and R. Sengupta, "Comparative Performance Evaluation of Routing Protocols for Mobile, Ad hoc Networks," In *Proceedings of IEEE IC3N'98*, Lafayette, LA, Oct. 1998, pp. 153-161.

[7] S.E. Deering and D.R. Cheriton, "Multicast Routing in Datagram Internetworks and Extended LANs," *ACM Transactions on Computer Systems*, vol. 8, no. 2, May 1990, pp. 85-110.

[8] S. Deering, D.L. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu, and L. Wei, "The PIM Architecture for Wide-Area Multicast Routing," *IEEE/ACM Transactions on Networking*, vol. 4, no. 2, Apr. 1996, pp. 153-162.

[9] C.L. Fullmer and J.J. Garcia-Luna-Aceves, "Solutions to Hidden Terminal Problems in Wireless Networks," In *Proceedings of ACM SIGCOMM'97*, Cannes, France, Sep. 1997, pp. 39-49.

[10] J.J. Garcia-Luna-Aceves and E.L. Madruga, "The Core-Assisted Mesh Protocol," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, Aug. 1999, pp. 1380-1394.

[11] J.J. Garcia-Luna-Aceves and E.L. Madruga, "A Multicast Routing Protocol for Ad-Hoc Networks," In *Proceedings of IEEE INFOCOM'99*, New York, NY, Mar. 1999, pp. 784-792.

[12] IEEE Computer Society LAN MAN Standards Committee, *Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specification*, IEEE Std 802.11-1997. The Institute of Electrical and Electronics Engineers, New York, NY, 1997.

[13] Internet Engineering Task Force (IETF) Mobile Ad Hoc Networks (MANET) Working Group Charter. http://www.ietf.org/html.charters/manet-charter.html.

[14] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, "Scenario-Based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks," In *Proceedings of ACM/IEEE MOBICOM'99*, Seattle, WA, Aug. 1999, pp. 195-206.

[15] J. Jubin and J.D. Tornow, "The DARPA Packet Radio Network Protocols," *Proceedings of the IEEE*, vol. 75, no. 1, Jan. 1987, pp. 21-32.

[16] E.D. Kaplan (Editor), *Understanding the GPS: Principles and Applications*, Artech House, Boston, MA, Feb. 1996.

[17] S.-J. Lee, M. Gerla, and C.-C. Chiang, "On-Demand Multicast Routing Protocol," In *Proceedings of IEEE WCNC'99*, New Orleans, LA, Sep. 1999, pp. 1298-1304.

[18] S.-J. Lee, W. Su, and M. Gerla, "Ad hoc Wireless Multicast with Mobility Prediction," In *Proceedings of IEEE ICCCN'99*, Boston, MA, Oct. 1999, pp. 4-9.

[19] S.-J. Lee, W. Su, and M. Gerla, "On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks," *Internet-Draft*, draft-ietf-manet-odmrp-01.txt, Jun. 1999, Work in progress.

[20] S.-J. Lee, M. Gerla, and C.-K. Toh, "A Simulation Study of Table-Driven and On-Demand Routing Protocols for Mobile Ad-Hoc Networks," *IEEE Network*, vol. 13, no. 4, Jul. 1999, pp. 48-54.

[21] E.L. Madruga and J.J. Garcia-Luna-Aceves, "Multicasting Along Meshes in Ad-Hoc Networks," In *Proceedings of IEEE ICC'99*, Vancouver, Canada, Jun. 1999, pp. 314-318.

[22] E.L. Madruga and J.J. Garcia-Luna-Aceves, "Scalable Multicasting: The Core Assisted Mesh Protocol," To appear in *ACM/Baltzer Mobile Networks and Applications*, Special Issue on Management of Mobility, 1999.

[23] J. Moy, "Multicast Routing Extensions for OSPF," *Communications of the ACM*, vol. 37, no. 8, Aug. 1994, pp. 61-66, 114.

[24] S. Murthy and J.J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks," *ACM/Baltzer Mobile Networks and Applications*, vol. 1, no. 2, Oct. 1996, pp. 183-197.

[25] R. Prakash, "Unidirectional Links Prove Costly in Wireless Ad-Hoc Networks," In *ACM DIAL M'99 Workshop*, Seattle, WA, Aug. 1999, pp. 15-22.

[26] T.S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, Upper Saddle River, NJ, Oct. 1995.

[27] T.S. Rappaport, S.Y. Seidel, and K. Takamizawa, "Statistical Channel Impulse Response Models for Factory and Open Plan Building Radio Communication System Design," *IEEE Transactions on Communications*, vol. COM-39, no. 5, May 1991, pp. 794-807.

[28] UCLA Computer Science Department Parallel Computing Laboratory and Wireless Adaptive Mobility Laboratory, *GloMoSim: A Scalable Simulation Environment for Wireless and Wired Network Systems*. http://pcl.cs.ucla.edu/projects/domains/glomosim.html

[29] C.W. Wu, Y.C. Tay, and C.-K. Toh, "Ad hoc Multicast Routing protocol utilizing Increasing id-numberS (AMRIS) Functional Specification," *Internet-Draft*, draft-ietf-manet-amris-spec-00.txt, Nov. 1998, Work in progress.