MAC RELIABLE BROADCAST IN AD HOC NETWORKS

Ken Tang, Mario Gerla
Computer Science Department
University of California, Los Angeles
{ktang, gerla}@cs.ucla.edu

ABSTRACT

Traditional wireless ad hoc medium access control (MAC) protocols often utilize control frames such as Request-To-Send (RTS), Clear-To-Send (CTS) and Acknowledgement (ACK) to reliably delivery unicast data. However, little effort has been given to improve the reliable delivery of broadcast data. Often, broadcast data are transmitted blindly without any consideration of hidden terminals. In this paper, we proposed a novel MAC protocol, Broadcast Medium Window (BMW) that reliably delivers broadcast data.

INTRODUCTION

Multicasting in ad hoc networks has generated considerable interest in the wireless community. The common approach to the multicast problem is to resolve it at the network layer. Examples of such multicast protocols include AMRoute [2], ODMRP [10][11][12], AMRIS [15] and CAMP [5]. These protocols are utilized on top of the medium access control (MAC) layer, commonly IEEE 802.11 [3]. However, currently wireless random access MAC protocols, such as 802.11, do not support reliable multicasting of data since multicast protocols often rely on the link-level broadcasting of packets to achieve multicast. For example, 802.11 uses collision avoidance along with RTS/CTS/ACK control frames to transmit unicast packets in order to combat hidden terminals. For broadcast packets that are to be received by all neighbors of the source node, no control frames are used. Therefore, broadcast packets are sent blindly without consideration of hidden terminals and channel noise.

In this paper, we introduce a new MAC protocol, Broadcast Medium Window (BMW), which supports reliable multicast in ad hoc networks. We first start by introducing the BMW protocol. We then present the simulation parameters and results of applying BMW under ODMRP vs using 802.11. Finally, we end with our concluding remarks.

BROADCAST MEDIUM WINDOW (BMW)

The fundamental idea behind BMW is to reliably transmit each packet to each neighbor in a round robin fashion. However, since BMW exploits many of the same concepts of IEEE 802.11, a brief operational overview of 802.11 is in order.

IEEE 802.11 utilizes a collision avoidance scheme along with RTS/CTS/ACK control frames to transmit unicast packets. In 802.11, the Distributed Coordination Function (DCF) represents the basic access method that mobile nodes utilize to share the wireless channel. The scheme incorporates CSMA with Collision Avoidance (CSMA/CA) and acknowledgement (ACK). Optionally, the mobile nodes can make use of the virtual carrier sense mechanism that employs RTS/CTS exchange for channel reservation and fragmentation of packets in situations where the wireless channel experiences high bit error rate. CSMA/CA works as follows. A node wishing to transmit senses the channel. If the channel is free for a time equal to the DCF InterFrame Space (DIFS) interval, the node transmits. If the channel is busy, the node enters a state of collision avoidance and backs off from transmitting for a specified interval. In the collision avoidance state, the node sensing the channel busy will suspend its backoff timer, only resuming the backoff countdown when the channel is again sensed free for a DIFS period. A typical sequence of exchanges in 802.11 using the virtual carrier sensing mechanism involves the source node first sensing the channel using CSMA/CA. After CSMA/CA is executed, the source node transmits RTS, followed by the destination node responding with CTS, then with the source node sending the data frame and finally with the destination node confirming with an ACK to the source node. Any nodes receiving RTS, CTS or data frame that is not an intended destination will yield long enough for the source and destination nodes to complete the data exchange. For broadcast packets, IEEE 802.11 nodes simply execute collision avoidance and then transmit the data frame.

In BMW, each node is required to maintain three lists: a neighbor list (NEIGHBOR LIST), a list of transmitted

frames (SEND BUFFER) and a list of received sequence numbers (RECEIVER BUFFER). All nodes keep track of their neighbors upon reception of frames (RTS/CTS/DATA/ACK/HELLO). Upon receiving any type of frames, a node updates its NEIGHBOR LIST. Furthermore, the NEIGHBOR LIST is purged of the neighboring node if the neighboring node in the NEIGHBOR LIST has not been heard from for a specified amount of time. Each node also maintains a SEND BUFFER. The SEND BUFFER holds copies of the frames that were already transmitted but might be needed later for retransmission. A copy is removed from the SEND BUFFER after all neighbors have received it. The size of the SEND BUFFER should be at least as large as the maximum number of neighbors for any given node. Besides the SEND BUFFER, there is also a queue that stores packets that have not yet been transmitted. Finally, each node also maintains RECEIVER BUFFER. When a node receives a new frame, it records the frame's sequence number in RECEIVER BUFFER. When a source node transmits RTS to a destination node specifying a range of (from and to) sequence numbers, the destination node examines its RECEIVER BUFFER to determine whether it is missing any previous sequence numbers in the specified range. If so, the destination node replies with the missing sequence number in the CTS response.

In BMW, when a node has a packet to transmit, it first senses the channel and goes through a collision avoidance (CSMA/CA) phase similar to that of 802.11. Upon the completion of the collision avoidance phase when the channel becomes free, the node sends RTS to one of its neighbors, specifying what sequence numbers have already been sent and what the current sequence number is. This is accomplished by extracting the lowest sequence number from the SEND BUFFER and specifying it into the RTS frame along with the current sequence number the source node is expecting. Upon receiving the RTS, the intended neighbor examines its RECEIVER BUFFER and determines what sequence number it needs. If the node is missing a frame of a previous sequence number, the CTS response frame will reflect that. Likewise, if only the current sequence number is needed, the CTS response frame will reflect that as well. All other neighbors hearing the RTS will yield long enough for the CTS/DATA/ACK transmission. After the reception of the CTS, the source node then transmits the DATA (packet) that corresponds to the sequence number specified in the CTS frame. All other nodes hearing the CTS frame will yield long enough for the DATA/ACK transmission. Upon receiving the DATA, the destination node updates its RECEIVER BUFFER and replies with an ACK. All other neighboring nodes that received the DATA will also update their RECEIVER BUFFER. Upon receiving the ACK, if the DATA sent was not a current DATA but was instead obtained from the buffer, the source node continues its dialogue with the destination node with another RTS until the current DATA is sent from the queue. Here, the collision avoidance phase is skipped. Once the current DATA is transmitted and acknowledged, the source node then buffers the packet and chooses the next neighbor in its NEIGHBOR LIST and repeats the whole process over again.

The round robin process runs smoothly when there are always packets to send. However, when there are no packets left in the transmit queue, the round robin process will halt and the source will not know whether the next neighbor in the NEIGHBOR LIST received all the broadcast DATA correctly until there is a new packet to send. To prevent this, BMW sets a timer for transmitting to the next node in the NEIGHBOR LIST. If the queue is empty for the time equal to this timer, the next node in the NEIGHBOR LIST will be chosen and the round robin process continues. If all the neighbors are visited in the round robin process and the queue is still empty, the round robin process stops until there is a new packet to transmit.

To detect neighbors, BMW relies on either transmitting HELLO frames periodically or eavesdropping on existing MAC frames (RTS/CTS/DATA/ACK). To reduce the HELLO frame overhead, a node that has just transmitted a frame will not send a HELLO frame for that given time period. In the event that the nodes have absolutely no knowledge of any of their neighbors, transmissions by nodes are done through unreliable broadcasting (strict CSMA/CA) of the packets until the neighbors are detected.

In circumstances where reliable transmission is counterproductive (for example, when channel contention is extremely high or sources become extremely aggressive), BMW reverts back to the unreliable delivery of 802.11. That is, all packets are unreliably broadcasted. This is accomplished by keeping track of the number of retransmissions and queue overflow. If the number of retransmission to a particular neighbor reaches a certain threshold, the neighbor is cleared from the NEIGHBOR LIST. Thus, if all the neighbors are congested, the neighbor list will be purged of all neighbors and unreliable broadcasting will resume until neighbors are again detected. If the queue overflows, BMW will unreliably broadcasts packets until the queue size decreases to a certain threshold.

We illustrate the concept of BMW through an example. Let us assume that node 5 wants to transmit a broadcast packet in Figure 1. Node 5 first determines a neighbor, say node 1, and sends RTS with sequence numbers ranging from 0 to 0 since no DATA frames have yet been sent.

Node 1, upon receiving the RTS frame, replies with sequence number 0 in the CTS frame. Nodes 2, 3, and 4, upon receiving the RTS frame, yield long enough for the CTS/DATA/ACK exchange between node 5 and 1. After receiving the CTS frame, node 5 transmits DATA with sequence number 0. Node 1, upon receiving DATA, updates its RECEIVER BUFFER and replies with an ACK. For illustration purposes, let's say node 2 did not receive the DATA (possibly due to interference from neighboring nodes) while node 3 and 4 received the DATA correctly. Thus, node 3 and 4 also update their RECEIVER BUFFER. Upon receiving the ACK, node 5 stores the DATA that was sent into the SEND BUFFER and then selects node 2 as its next neighbor to transmit to. After executing the collision avoidance phase, node 5 sends RTS with sequence number range 0 to 1. Upon receiving the RTS, node 2 examines its RECEIVER BUFFER and noticed that frame 0 has not yet been received. Node 2 then sends CTS requesting sequence number 0. Node 5, upon receiving the CTS, obtains the DATA with sequence number 0 from the buffer and transmits the DATA. Upon receiving the DATA, node 2 updates its RECEIVER BUFFER and responds with an ACK. Upon receiving the ACK, node 5 sends RTS again with sequence number range 0 to 1 since the most recent DATA has not yet been sent. Node 2, upon receiving the RTS, sends CTS with sequence number 1 after examining its RECEIVER BUFFER. Node 5, upon receiving the CTS, sends the DATA with sequence number 1. Node 2, upon receiving the DATA, replies with an ACK. Again, for illustration purposes, let's say nodes 1, 3 and 4 successfully receive the DATA and update their respective RECEIVER BUFFER. Node 5, upon receiving the ACK, buffers the DATA in SEND BUFFER and elects node 3 as its next neighbor. Following the collision avoidance phase, node 5 transmits RTS with sequence number range 0 to 2. Upon receiving the RTS, node 3 examines its received sequence number list and sends CTS requesting sequence number 2 (since 0 and 1 were successfully received previously). Node 5, upon receiving CTS, transmits DATA with sequence number 2. Node 3, upon receiving DATA, transmits ACK and updates its RECEIVER BUFFER. Node 5, upon receiving ACK, buffers the DATA in SEND BUFFER, selects node 4 as it's next neighbor to transmit to, and the process resumes.

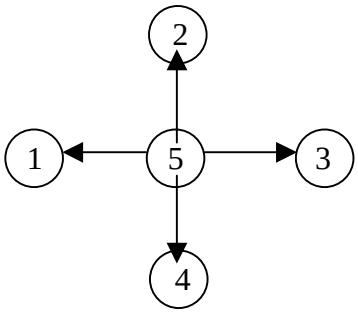


Figure 1. Node 5 broadcasting packets. Nodes 1, 2, 3 and 4 are within range of node 5 but not with each other.

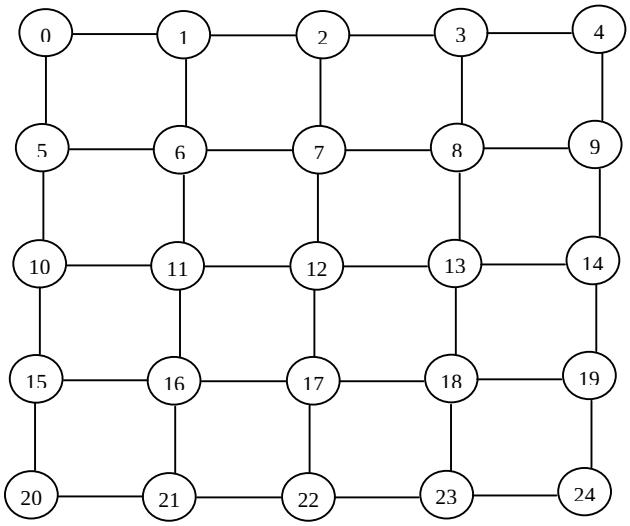
Note that at the start, nodes do not know who their neighbors are. Therefore, initial transmissions by nodes are done by broadcasting the packets until neighbors are detected.

SIMULATION PARAMETERS

We have simulated BMW using GloMoSim [16]. GloMoSim is a discrete event, parallel simulation environment implemented in PARSEC, PARallel Simulation Environment for Complex Systems. We use the On-Demand Multicast Routing Protocol (ODMRP) [10][11][12] to multicast packets from different sources to numerous destinations and with varying traffic rates. The nature of the ODMRP protocol heavily relies on the broadcast mechanism of the MAC layer to achieve multicast and therefore is a good multicast routing candidate to evaluate the multicast support feature of BMW. Moreover, the broadcast transmission in ODMRP is used for flooding control packets as well as data packets. Flooding is a very redundant type of transport mechanism: if a node fails to obtain a packet broadcast by neighbor A, there is an excellent chance it will get it again from neighbor B and so on. Nodes with just one neighbor (i.e. stubs) are at a disadvantage. However, stubs suffer no hidden terminal problems. In summary, strict reliable broadcast is not required for ODMRP but is likely to improve the performance significantly.

In our simulations, we consider a grid topology with 25 nodes as shown in Figure 2 and a topology where 25 nodes are uniformly placed in a 1000m x 1000m area. Nodes are only within radio range of their immediate neighbors in the grid topology scenario while in the uniformly placed scenario, the radio range of each node is 300m. CBR (constant bit rate) sources with various packet transmission rates running on top of UDP are used for the application.

We utilize ODMRP to route multicast packets with varying number of multicast sources and members. For our experiments, we only consider one multicast group. Radios with no capture ability are modeled with a channel bandwidth of 2Mbps. Furthermore, the channel uses free-space with a threshold cutoff and the power of a signal attenuates as $1/d^2$ where d is the distance between two nodes. The simulation results are obtained from multiple simulation runs (each lasting 600 seconds) with varying seed numbers and are averaged out over the multiple runs. Each data packets are 512B.



25 Nodes

Figure 2. Grid topology. Nodes are only within range of their immediate neighbors.

We use the packet delivery ratio of ODMRP to determine the effectiveness of BMW and IEEE 802.11. The packet delivery ratio is the ratio of the number of data packets actually delivered to the destinations over the number of data packets that are supposed to be received. The packet delivery ratio measures the effectiveness of a multicast protocol [11].

SIMULATION RESULTS

In this section, we compare the performance of BMW and 802.11 by examining the packet delivery ratio of ODMRP.

We first examine a simple scenario consisting of 25 nodes placed in a grid format (Figure 2). Here, nodes 2, 12 and 22 are multicast sources while nodes 0, 4, 11, 13, 20 and 24 are multicast receivers. We vary the traffic rate of the multicast sources from 10ms to 500ms. This contrive scenario is useful to understand the circumstances of the experiments, such as the placement of multicast sources and members and where the intermediate forwarding groups are located. By understanding the scenario, we are able to more accurately understand the outcome. We will examine a more realistic scenario at the end of the section. The outcome of this contrive experiments is shown in Figure 3.

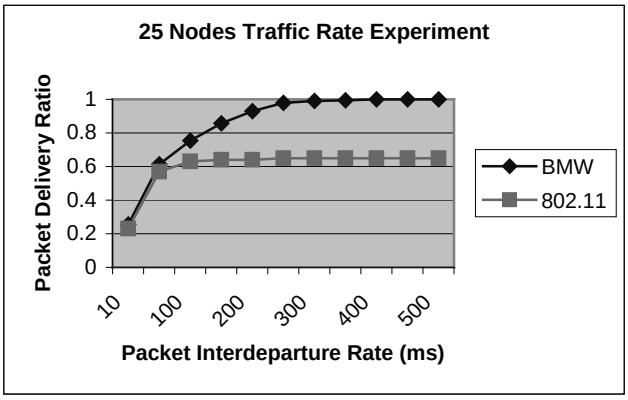


Figure 3. 25 nodes traffic rate experiment.

We observe that under high traffic rate (10ms), both 802.11 and BMW achieved the same packet delivery ratio (23%). With each source transmitting a packet every 10ms, BMW is not given enough time to combat packet loss. Since the rate of incoming packets is greater than the rate of outgoing packets, the network is overloaded and incoming packets will eventually overflow the MAC queue and get dropped. At 10ms, it would simply be better to not attempt reliable transmission. Thus, BMW simply reverts to 802.11 due to queue overflow and packet retransmissions caused by network overload and heavy contention among the intermediate nodes (or the forwarding groups) between the multicast sources and members. However, as the traffic rate slows down, BMW begins to dominate 802.11. The packet delivery ratio under BMW reaches 100% after 250ms. As we expected, the longer interdeparture time between packets allows BMW enough time to retransmit lost packets before new packets arrive. 802.11, on the other hand, is unable to combat packet loss, asymptotically achieving a packet delivery ratio of 66% as traffic rate decreases.

Next, we examine the same 25-node grid scenario. However, this time we vary the number of multicast sources from 1 to 16 while keeping the multicast members constant. Nodes 0, 4, 12, 20 and 24 are the fixed multicast members. The sources are chosen to be evenly spaced apart from one another. Each source transmits at an interdeparture rate of 500ms. Figure 4 presents the results.

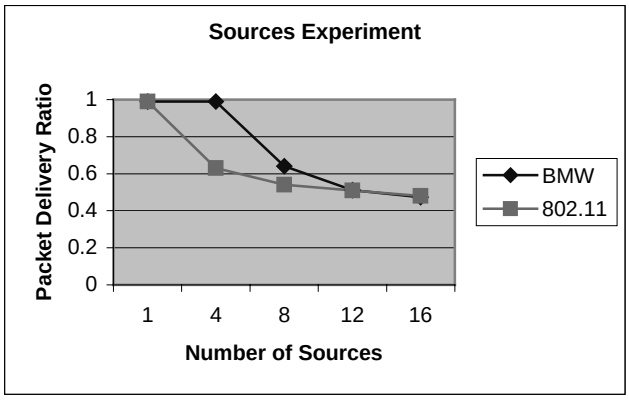


Figure 4. Sources experiment.

From Figure 4, we observe that with one multicast source, both BMW and 802.11 achieved complete reliable delivery due to the lack of channel contention in the network. However, as we increase the number of sources, both protocols start to falter. BMW is still able to combat packet loss with 4 sources but performance worsens when more sources are introduced into the network. Increasing the number of sources has the same effect as increasing the traffic rate. From the traffic rate experiments, we learn that BMW degrades to 802.11 when traffic rate is high. Thus, we would also expect BMW to eventually achieve the same packet delivery ratio as that of 802.11 as we increase the number of sources. This event occurs when there are more than 12 sources.

Next, we study the performance of BMW and 802.11 as we vary the number of multicast members from 1 to 16 in a 25-node grid topology. Nodes 0, 12 and 24 are multicast sources while the multicast members are chosen to be evenly spaced apart between one another. Sources transmit at an interdeparture rate of 500ms. The results are given in Figure 5.

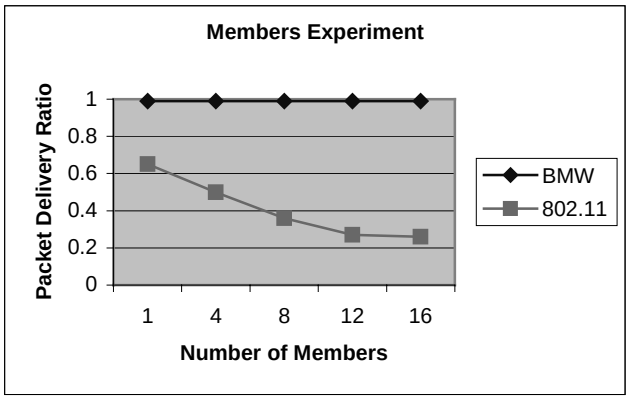


Figure 5. Members experiment.

As we can see from Figure 5, BMW is robust to the number of multicast members in the network. The performance of 802.11, on the other hand, decays, reaching an asymptotic packet delivery ratio of 25% as the number of multicast members grows. The effect of increasing the number of multicast members is that data packets must now be delivered to new members. This is not a concern under BMW since BMW already attempts to reliably deliver every packet to all neighboring nodes. On the other hand, increasing the number of multicast members troubles 802.11 since data packets must now reach new members that might not normally be reached.

Finally, we consider a more realistic ad hoc scenario. To this extent, we uniformly placed 25 nodes in a 1000m x 1000m area and randomly select 5 multicast sources and 5 multicast members. We vary the packet interdeparture time (traffic rate) of each source from 50ms to 500ms. Figure 6 depicts the result.

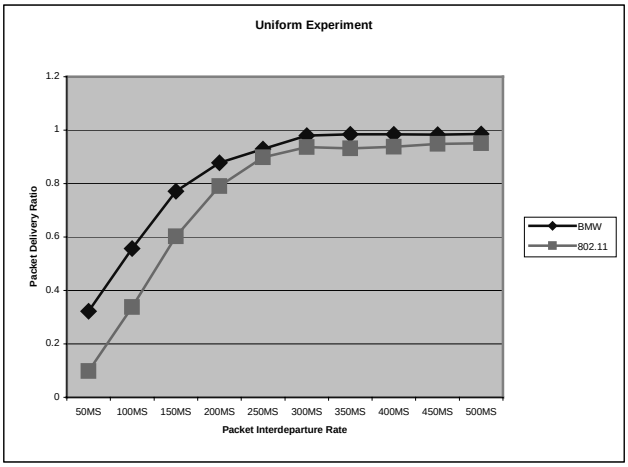


Figure 6. Uniform experiment.

From Figure 6, BMW outperforms 802.11 under all packet interdeparture rates, with the difference in performance showing the most under high traffic rates. At an interdeparture rate of 50ms, BMW is able to achieve three times the productivity of 802.11. However, as the traffic rates decrease, the packet delivery ratio of BMW and 802.11 starts to converge, with BMW exhibiting slightly higher packet delivery ratio. Still, BMW is still able to outperform 802.11 under a more realistic ad hoc scenario than the grid topology.

CONCLUSION

In this paper, we have presented a novel MAC protocol, BMW, which reliably delivers broadcast data under low to medium network load. Under such conditions, BMW is given ample time to retransmit lost packets. However, when the network load is high, attempting to recover from packet loss is counterproductive. In this case, it's better to use the ALOHA approach of 802.11 instead. Also, attempting to reliably deliver data to all known neighbors may not be the best approach to facilitate multicast routing protocols. An alternative tactic is to only reliably deliver data to neighbors of importance to the underlying multicast routing protocols. For example, in ODMRP, only nodes that are part of the multicast members or forwarding group serve a purpose. Other nodes can be disregarded. The advantage of this method is that we eliminate unnecessary efforts to reliably delivery data to "useless" neighbors and will therefore improve the network performance. The drawback would be that the MAC protocol would have to be tailored for each multicast protocol since different multicast protocols employ different schemes.

We are currently experimenting with BMW under other various scenarios to better understand its impact. Further work is also undergoing in incorporating the BMW concept into the ad hoc multicast routing protocols themselves. By doing so, the multicast routing protocol will be able to provide reliable delivery independent of the MAC protocol used.

REFERENCES

[1] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LAN's," ACM SIGCOMM, 1994.

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

[3] Editors of IEEE 802.11, Wireless LAN Medium Access Control (MAC and Physical Layer (PHY) specifications, Draft Standard IEEE 802.11, 1997.

[4] C. Fullmer and J.J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Access (FAMA) for packet radio networks," Computer Communication Review, vol. 25, (no. 4), (ACM SIGCOMM '95, Cambridge, MA, USA, 28 Aug.-1 Sept. 1995.) ACM, Oct. 1995.

[5] 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.

[6] J. Haartsen, M. Naghshineh, J. Inouye, O.J. Joeressen, and W. Allen, "Bluetooth: Vision, Goals, and Architecture," ACM SIGMOBILE Mobile Computing and Communications Review, vol. 2, no. 4, Oct. 1998, pp. 38-45.

[7] Anthony Joseph, B. R. Badrinath, and Randy Katz, "A Case for Services over Cascaded Networks," First ACM/IEEE International Conference on Wireless and Mobile Multimedia (WoWMoM'98), Dallas, Texas, October 30, 1998.

[8] John Jubin and Janet D. Tornow, "The DARPA Packet Radio Network Protocols," Proceedings of the IEEE, Jan. 1987.

[9] P. Karn, "MACA – A New Channel Access Method for Packet Radio," in ARRL/CRRL Amateur radio 9th Computer Networking Conference, ARRL, 1990.

[10] S.-J. Lee, M. Gerla, and C.-C. Chiang, "On-Demand Multicast Routing Protocol," Proceedings of IEEE WCNC'99, New Orleans, LA, Sep. 1999, pp. 1298-1302.

[11] S.-J. Lee, W. Su, J. Hsu, M. Gerla, and R. Bagrodia, "A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols," Proceedings of IEEE INFOCOM2000, Tel Aviv, Israel, Mar. 2000. Proceedings of IEEE INFOCOM2000, Tel Aviv, Israel, Mar. 2000.

[12] S.-J. Lee, W. Su, and M. Gerla, Internet Draft, draft-ietf-manet-odmrp-02.txt, Jan. 2000.

[13] K.J. Negus, J. Waters, J. Tourrilhes, C. Romans, J. Lansford, and S. Hui, "HomeRF and SWAP: Wireless Networking for the Connected Home," ACM SIGMOBILE Mobile Computing and Communications Review, vol. 2, no. 4, Oct. 1998, pp. 28-37.

[14] Andrew S. Tanenbaum, "Computer Networks: Third Edition," Prentice Hall PTR, New Jersey, 1996.

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

[16] X. Zeng, R. Bagrodia and M. Gerla, "GloMoSim: a Library for the Parallel Simulation of Large-scale Wireless Networks," PADS, 1998.