Random Access MAC for Efficient Broadcast Support in Ad Hoc NetworkS

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

Abstract **- Wireless communications are becoming an important part of our everyday lifestyle. One major area that will have an enormous impact on the performance of wireless ad hoc networks is the medium access control (MAC) layer. Current random access MAC protocols for ad hoc networks support reliable unicast but not reliable broadcast. In this paper, we proposed a random access MAC protocol, Broadcast Support Multiple Access (BSMA), which improves broadcast reliability in ad hoc networks.**

I. INTRODUCTION

Wireless technology has rapidly expanded in recent years and will continue to grow in years to come. Initiatives such as Bluetooth [4], Iceberg [5] and HomeRF [11] are making wireless networking a reality. Such technologies allow for wireless communications between numerous devices, such as the PCs, PDAs, cellular and cordless phones, pagers and other appliances, to be practical and seamless.

Many random access protocols exist in ad hoc networks that facilitate wireless ad hoc communication. Some of these protocols include Carrier Sense Multiple Access (CSMA) [6], Multiple Access with Collision Avoidance (MACA) [7], MACAW [1], Floor Acquisition Multiple Access (FAMA) [3] and IEEE 802.11 [2]. However, none of these protocols are designed to support the reliable broadcasting of data. For example, 802.11 uses collision avoidance along with RTS/CTS/ACK control frames to transmit unicast packets in order to combat hidden and exposed 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 and exposed terminals and channel noise.

II. BROADCAST SUPPORT MULTIPLE ACCESS (BSMA)

In this paper, we propose a new protocol, Broadcast Support Multiple Access (BSMA), which facilitates the transmission of broadcast packets and therefore supports multicast protocols such as ODMRP [8][10] as they heavily rely on broadcasting packets to achieve multicast. Our protocol utilizes some of the features of 802.11. Therefore, an 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, and if it is free for a time equal to the DCF InterFrame Space (DIFS), 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 ending with the destination node confirming with an ACK to the source node. Any nodes receiving RTS, CTS or data frame that does not belong to it will yield long enough for the source and destination nodes to complete the data exchange. For broadcast packets, 802.11 nodes simply execute collision avoidance and then transmit the data frame.

Fig. 1 outlines the steps of the BSMA protocol.

1. Collision avoidance phase.
2. Source sends RTS to all neighbors and sets timer to WAIT_FOR_CTS.
3. Neighbors of source, upon receiving RTS, send CTS if not in YIELD state and set timer to WAIT_FOR_DATA.
4. If source receives CTS, send DATA and set timer to WAIT_FOR_NAK. Else, if no CTS and WAIT_FOR_CTS timer expires, back off and go to step 1. Nodes that are not involved in the broadcast exchange, upon receiving CTS, sets their state to

YIELD and set their timer long enough to allow for the broadcast exchange to complete.

5. Neighbors send NAK if WAIT_FOR_DATA timer expires and DATA has not been received.
6. If source receives NAK before WAIT_FOR_NAK timer expires, back off and go to step 1. Else, if no NAK and WAIT_FOR_NAK timer expires, the broadcast is complete. Go to step 1 and get ready to transmit new DATA.

Fig. 1. BSMA protocol steps assuming radio with DS capture feature.

The protocol assumes the radio has DS (direct sequence) capture ability. That is, a radio with capture ability has the capability to lock onto a sufficiently strong signal in the presence of other interfering, less powerful signals. If the ratio of the arriving packet's signal strength over the sum of all colliding packets is larger than the threshold value, the packet is successfully received while the other colliding packets are dropped. Else, nothing is received and no collision is detected.

BSMA incorporates the collision avoidance and RTS/CTS control frames of 802.11 and relies on negative acknowledgements (NAKs) to deliver broadcast packets. Before broadcasting a packet, the source node must first complete the collision avoidance phase. Once the collision avoidance phase is accomplished, the source node broadcasts RTS to its neighbors and sets the WAIT_FOR_CTS timer. Upon receiving the RTS, each neighbor transmits CTS if it is not in YIELD state (due to the reception of remote frames) and sets the WAIT_FOR_DATA timer. If the source node receives CTS before the WAIT_FOR_CTS timer expires, the node transmits DATA. Otherwise, the source node backs off and retransmits at a later time. Any neighbor of the source node that did not receive DATA within the expected WAIT_FOR_DATA time period transmits NAK to the source. If the source node does not receive NAK after the WAIT_FOR_NAK timer expires, the source node assumes that all the neighbors have successfully received DATA. Otherwise, the source node will back off and retransmit DATA at a later time. Any node hearing the CTS that is not a neighbor of the source node sets its state to YIELD and yields long enough for the source node to transmit DATA and receive the possible NAK.

To illustrate the protocol, we assume from Fig. 2 that node C is the source node.

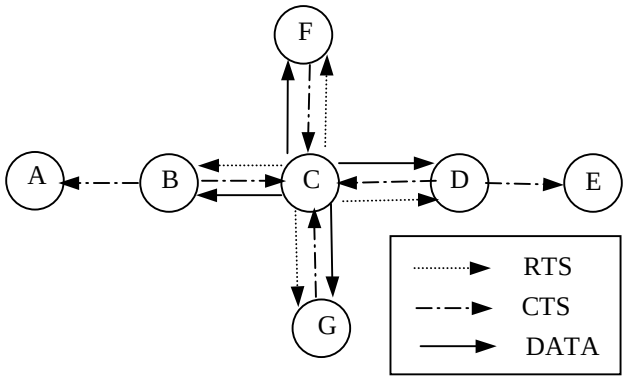

Fig. 2. Illustration of BSMA.

Node C first goes through the collision avoidance phase and then sends RTS to all of its neighbors. Nodes B, D, F and G receive the RTS and respond with CTS. Nodes A and E receive the CTS, set their states to YIELD and yield long enough for node C to transmit DATA and receive possible NAK. Node C, upon receiving the CTS from the strongest transmitter (capture is assumed), transmits the data frame and waits for NAK. Nodes B, D, F and G correctly receive the data frame and remain silent. If any of them did not receive DATA within the expected time period, it will send NAK to node C. Again, node C will receive the strongest NAK because of capture. Node C would then have to retransmit DATA at a later time upon receiving the NAK. The attentive reader will notice that in spite of the capture assumption, the above procedure is not completely fail safe. First, we must require that, when node, say X, starts the broadcast procedure, no other node, say Y, within a three hop radius will also start broadcasting, during the RTS/CTS vulnerable interval. If this happens, it is possible that part of the nodes which are neighbors of both X and Y will get only one or the other broadcast, depending on the distance and signal strength. Other nodes may miss the broadcast because they are in the YIELD state. Another important assumption is uniform transmission across all nodes. This prevents nodes outside of the YIELD shield to "penetrate" such shield during the data transmission phase.

III. SIMULATION CONFIGURATIONS

We have simulated BSMA using GloMoSim [13]. 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) [8][10] 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 broadcast support feature of BSMA. 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, and therefore perform better with BSMA. In summary, strict reliable broadcast is not required for ODMRP. But, efficient broadcast is likely to improve the performance significantly.

The On-Demand Multicast Routing Protocol (ODMRP) creates and maintains a mesh of nodes known as the forwarding group. The forwarding group is responsible for forwarding the multicast packets to the multicast members in the mesh via flooding. ODMRP uses a soft state approach to maintain multicast groups. Thus, members are refreshed periodically instead of sending explicit leave messages to leave the group. Since ODMRP is an on-demand protocol, group membership and multicast routes are established and updated by the source on a need-only basis (on-demand). The protocol consists of two phases: request and reply. During the request phase, each source floods the network with a JOIN DATA packet when the source has data to send. When a node receives the JOIN DATA, it records the upstream node ID and rebroadcasts the packet to its neighbors unless JOIN DATA packet circulation is prevented by source ID check and duplicate detection/discard. Once the JOIN DATA packet reaches a multicast group receiver, the receiver creates a JOIN TABLE packet and broadcasts the packet to its neighbors. This is the reply phase. Upon receiving the JOIN TABLE, a node determines whether or not it is the next node of one of the entries in the JOIN TABLE. If so, the node knows it is on the path to the source and thus assigns itself as a member of the forwarding group. It then builds its own JOIN TABLE and broadcasts to its neighbors. The JOIN TABLE is propagated until it reaches the source. The process of sending the JOIN DATA packet to multicast members and the JOIN TABLE packet to the source forms a mesh of nodes called the forwarding group. The sender refreshes the membership by sending the JOIN DATA packet periodically.

We consider several topologies for our simulation experiments, varying from a 3 x 3 grid to 20 and 25 nodes randomly placed in a 1000m by 1000m area. 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. Radios with 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 1460B.

We use the packet delivery ratio of ODMRP to determine the effectiveness of BSMA. 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 is a common measure of the effectiveness of a multicast protocol [9].

To gauge the effectiveness of BSMA, we compare the performance of ODMRP when ODMRP is running on top of BSMA with that of the typical CSMA broadcast approach.

IV. SIMULATION RESULTS

In this section, we compare BSMA against CSMA and show that BSMA can significantly improve upon the CSMA approach of broadcasting packets.

Our first set of experiments focuses on a grid topology as shown in Fig. 3. In the grid experiment, nine stationary nodes are placed in a grid topology as shown in Fig. 3. Nodes 1, 3, 5 and 7 are transmitting data to node 4 at the same time. Each node can only reach its immediate neighbors. This is a rather unusual multicast pattern, in that the multicast group consists of four senders and just one receiver. However, the grid experiment was orchestrated to evaluate the performance of the two MAC protocols in situations where hidden terminals exist. The simulation results of the experiment are given in Fig. 4.

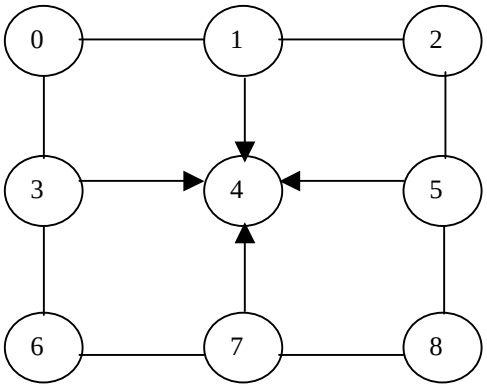

Fig. 3. Grid topology with nine nodes, with nodes 1, 3, 5 and 7 transmitting to node 4.

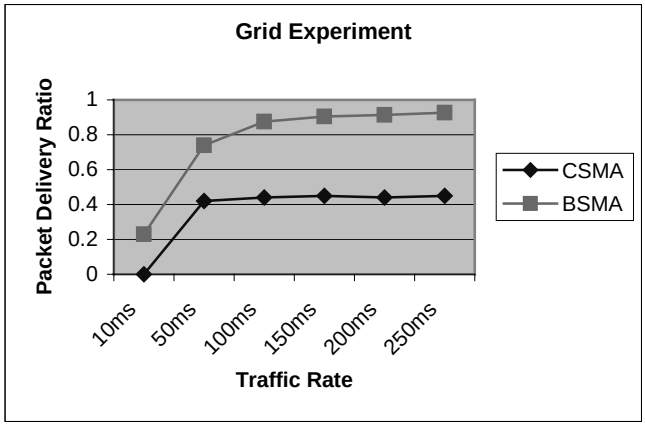

Fig. 4. Grid experiment.

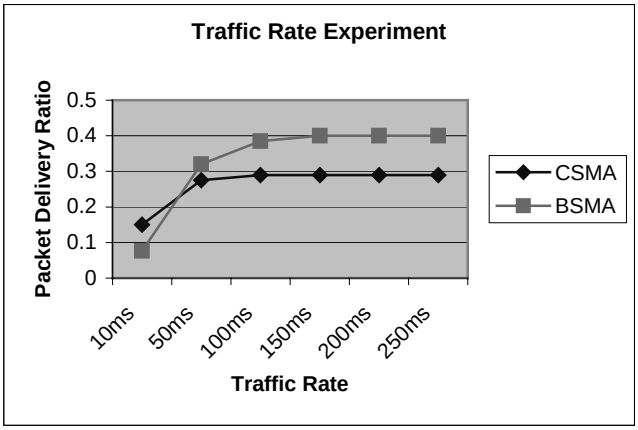

Fig. 5. Traffic rate experiment.

Here, we observe that at high rates, CSMA collapses. With a packet size of 1460B and channel bandwidth of 2Mbps, the transmission delay of each data packet is 5.84ms. At a packet interdeparture rate of 10ms and with four nodes transmitting to node 4 all at once, the channel contention becomes so high that hardly any packets get through to node 4. Keep in mind that ODMRP requires a request and a reply procedure to be completed before the forwarding group is set up. Due to the extremely high contention, the success rate of forming the mesh is low and as a result the packet delivery ratio suffers significantly. However, BSMA, using RTS/CTS/NAK, is able to combat the packet loss due to collisions and obtain a respectable packet delivery ratio of 23 percent. As the packet interdeparture time increases, BSMA continues to outperform CSMA. In fact, the packet delivery ratio of BSMA asymptotically converges to 92 percent while CSMA tops at 45 percent. The improvement of BSMA can be attributed to the fact that the RTS/CTS/NAK control frames are given time to combat packet loss as the traffic rate decreases. Nodes that transmit RTS control frames but did not receive a CTS in return or that encountered a NAK are given time to back off and retransmit at a later time where contention is less and before the next scheduled data packet is to be transmitted. On the other hand, packet loss recovery is not possible with CSMA. Therefore, the performance under CSMA degrades.

Since traffic rate has a major impact in the grid experiment where hidden terminals exist, we further investigate the effect of traffic rate under a more random topology and a more realistic multicast scenario. To this end, we generate 20 nodes that are randomly placed in a 1000m x 1000m area, each with a radio power range of 250m. There are five senders and five receivers in the multicast group. Again, the packet transmission interval varies from 10ms to 250ms. Fig. 5 depicts the result.

In Fig. 5, BSMA reaches a packet delivery ratio of 40 percent while CSMA levels out to at about 30 percent. In this scenario, flow and congestion control is critical. UDP does not carry out flow control. BSMA is able to improve the packet delivery ratio of CSMA by 10% in part because the RTS/CTS/NAK mechanism acts as a rudimentary flow control scheme; if no CTS is received after transmitting RTS or if a NAK is encountered, the sender backs offs and retransmits at a later time. We note that at high traffic rate (10ms), BSMA under performs CSMA, which appears to contradict the results of the grid experiment. However, in this scenario, there are more nodes, senders and receivers involved. Furthermore, nodes in the traffic rate experiment are placed randomly in a 1000m x 1000m area. As a result, the senders and receivers of the multicast group are more likely to be two to three hops apart from each other as opposed to one hop in the grid experiment. All of the above changes drastically increase the overall network load compared to that of the grid experiment. Thus, with a transmission delay of 5.84ms for each packet transmission, the control frames of BSMA are not given enough time to combat the loss and deliver the new incoming packets at the same time. As a result, the queue size grows, resulting in large queuing delays and packet drops due to queue overflow.

Next, we investigate the effect of the number of multicast senders on performance. In this experiment, 25 nodes are randomly placed in a 1000m by 1000m area, each with a radio power range of 250m. There are five multicast receivers and the number of multicast senders ranges from 1 to 20. The interdeparture time of the packets is 200ms (5 packets per second). All nodes are stationary throughout the experiment. Fig 6 shows the result as we vary the number of multicast senders in ODMRP.

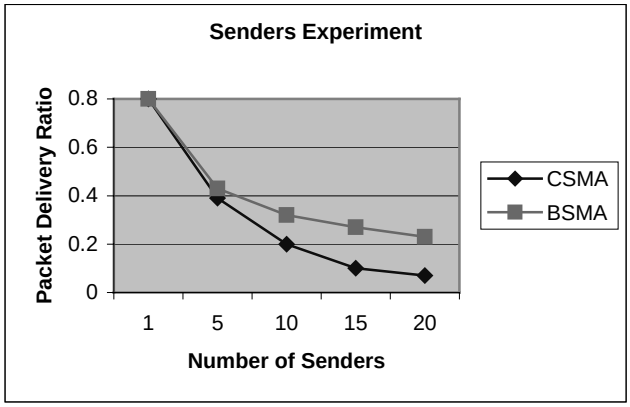

Fig 6. Senders experiment.

We note from Fig 6 that with a single sender, the packet delivery ratio is high (80 percent) for both protocols. With a single sender, channel contention is rare and therefore we would expect a high packet delivery ratio no matter what MAC protocol we use to broadcast packets. As the number of senders increases, we observe dramatic performance degradation as expected. Still, the BSMA improves upon the CSMA broadcast. Our approach has an asymptotic packet delivery ratio of 20 percent as the number of sender increases compared to that of the typical CSMA broadcasting practice, which is only 5 percent.

V. CONCLUSION

In this paper, we presented a novel MAC protocol, BSMA, which enhances packet broadcasting. BSMA is superior to the traditional CSMA method as the number of multicast senders increase. Also, BSMA functions best when the network load is low to medium. When the network load is high, the RTS/CTS/NAK control frames merely contribute to network congestion. It is important to note that BSMA does not guarantee the delivery of broadcast packets, but rather improves upon the delivery. More research is required in the direction of strictly reliable multicast, i.e., the assurance that all multicast neighbors have received the packet. The use of NAKs only determines whether nodes that respond to the RTS are correctly receiving the data. Nodes that are neighbors of the sender but did not respond to the RTS (possibly due to channel noise or packet collisions) are neglected. Thus, BSMA does not assure that all neighbors have received the required data. In a unicast environment, reliably delivery of data is accomplished through ACKs sent back by the destination. However, in multicast, this scheme is not feasible since it will lead to ACK implosion. Moreover, in wired networks (e.g. the Internet) the reliable multicast problem is typically solved at the transport and/or application level. That approach is justified by the fact that link loss is negligible in wired networks, and the major cause of loss is congestion (which is prevented with separate mechanisms such as TCP windowing, for example). In the wireless case, link loss due to interference is the major problem. Thus, it is necessary to act at the MAC and link layer first in order to achieve measurable efficiency. The transport and application layer recovery mechanisms will also be invoked. But, they will be much more effective if the MAC layer has been made reasonably reliable.

REFERENCE

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

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

[3] 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.

[4] 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.

[5] 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.

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

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

[8] 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.

[9] 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.

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

[11] 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.

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

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