A New MAC Protocol for Wireless Packet Networks

Jun Peng * Liang Cheng * Biplab Sikdar †

Dept. of Computer Science and Engr. *
Lehigh University
19 Memorial Drive West, Bethlehem, PA 18015
{jpeng,cheng}@cse.lehigh.edu

Dept. of Electrical, Computer and Systems Engr. †
Rensselaer Polytechnic Institute
110 8th Street, Troy, NY 12180
bsikdar@ecse.rpi.edu

***Abstract*— Medium access control (MAC) protocols for wireless packet networks usually need to be distributed for flexibility and robustness. In a distributed way, however, collision detection becomes difficult and collided packets are usually still fully transmitted with existing wireless MAC protocols, which wastes the precious medium resource of the network. This paper proposes a new MAC protocol that realizes effective and efficient collision detection in wireless packet networks by the use of pulses of random-length pauses. Our comprehensive simulation results show the capability of the new protocol for significantly improving the throughput of future wireless packet networks.**

I. INTRODUCTION

With their increasing popularity, wireless LANs sharing the same segment of radio spectrum may easily have overlapping areas. Severe medium contentions are therefore becoming common phenomena in wireless LANs. In addition, wireless ad hoc networks and mesh networks have received attentions in recent years due to their easy deployment in infrastructure-less environments. In such wireless networks, medium contentions may be even severer due to multi-hop traffic and larger numbers of nodes. Dealing with collisions have therefore become a critical issue for wireless networks.

In wireless packet networks, contention-based medium access control has the advantages of flexibility and robustness over schedule-based control and thus has become a popular strategy in such environments. The most widely used mechanism to avoid collisions in contention-based medium access control is probably "carrier sense" [1]. With carrier sense, nodes listen before they transmit. Only if the medium is sensed idle, nodes may transmit after proper backoffs. This "physical" carrier sense technique, however, can not always avoid collisions because of hidden terminals [2] and the propagation delays of signals [3].

Another technique called "virtual" carrier sense is used by some wireless MAC protocols such as [4], [5], [6], [7] to deal with hidden terminals, which are basically wireless nodes that can not sense a sender but may cause a collision at the receiver. With virtual carrier sense, a sender that acquires the medium after conducting physical carrier sense exchanges short control frames (i.e., RTS and CTS frames) with its receiver for two purposes. One is handshaking, while the other is to notify their neighbors of their transmission schedule.

Virtual carrier sense also has limited effectiveness in avoiding collisions. The exchanged control frames of two nodes may be lost to their neighbors due to the same problem of hidden terminals. In addition, node movement and the discrepancy between the transmission range and interference range of a node may affect the effectiveness of virtual carrier sense [8].

Due to the limited effectiveness of existing techniques for collision avoidance, collisions are not rare phenomena in wireless packet networks. When collisions happen, existing wireless MAC protocols usually can not promptly detect them and thus collided packets are still fully transmitted, which wastes the scarce medium resource. To address this problem, the new MAC protocol proposed in this paper uses a pulse-based approach to realize fully-distributed collision detection in wireless packet networks. By using "pulses" of random-length pauses, the proposed MAC protocol enables two or more nodes to detect each other when they transmit at the same time. Our extensive simulations have shown the effectiveness of the proposed protocol in detecting collisions and improving the throughput of wireless networks.

An out-of-band control channel was originally proposed for dealing with hidden terminals in wireless networks [2]. The BTMA [2] and RI-BTMA [9] protocols use a single control channel to address the hidden terminal problem. The DBTMA protocol [10] uses two control channels to address the hidden terminal problem and improve the spatial reuse of radio spectrum. An out-of-band control channel has also been used for other purposes in other wireless MAC protocols, such as priority scheduling [11], energy saving [12], and power control [13]. The proposed MAC protocol in this paper, however, uses an out-of-band control channel to address a different problem, which is collision detection in wireless networks.

The rest of the paper is organized as follows. Section II presents the proposed MAC protocol in detail. Section III evaluates it with extensive ns-2 [14] simulations. Finally, Section IV summarizes the paper.

II. THE PROPOSED MAC PROTOCOL

A. Protocol Basics

With the proposed MAC protocol, the control channel only carries "pulses" and "pulses" only appear in the control channel. "Pulses" are basically single-tone waves with pauses of random lengths, as shown in Fig. 1. A node transmits pulses in the control channel when it is transmitting a packet in the data channel. In addition, the proposed MAC protocol does not use RTS or CTS control frames in the data channel.

The proposed MAC protocol operates in the following basic way. An initiating sender having a packet to transmit first

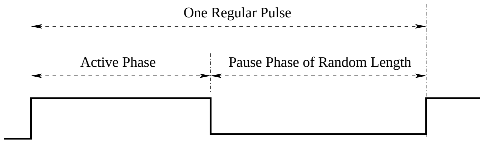

Fig. 1. A regular pulse consists of an active phase of a fixed length and a pause phase of a random length.

performs physical carrier sense in the control channel. After the channel has been sensed idle for a period of time longer than the maximum pause duration of a pulse, the node takes a random backoff. If the node does not hear other nodes during its backoff, it starts to generate pulses in the control channel and transmit the packet in the data channel upon the expiration of its backoff timer. Otherwise, the node keeps monitoring the control channel.

The initiating sender expects a CTS pulse after it finishes transmitting the headers of the frame containing the packet. If the node obtains the CTS pulse, it continues to transmit the packet. Otherwise, the sender aborts its transmission. On the other hand, after determining that the packet is intended for it, the intended receiver sends back a CTS pulse and starts to "relay" the pulses in the control channel.

The sender expects an acknowledgment from the receiver after finishing transmitting the packet. If the sender does not obtain the acknowledgment, it will retransmit the packet. This whole process repeats until an acknowledgment is obtained for the packet or the retransmission limit is reached.

B. The Pulses

As shown in Fig. 1, a pulse consists of an active phase of a fixed length and a pause phase of a random length. Single-tone waves are transmitted in the control channel in the active phase only. The active phase of a pulse signals a busy data channel, while the pause phase is mainly for collision detection. When a node is generating pulses, it still monitors the control channel in its pulse pauses.

A CTS pulse does not have a pause phase and its length is determined by the integer in a field of the MAC header of the received data frame, which is randomly drawn by the initiating sender. After the intended receiver determines that the data frame is intended for it, it sends back a CTS pulse in the following first pause detected in the control channel. A sender waiting for a CTS pulse divides its pause into two parts. One is the CTS window of a fixed length, while the other is the residual random pause following the CTS window. An initiating sender regards a CTS pulse legitimate if the pulse is of the expected length and received in the CTS window.

If a node is receiving a data frame *intended* for it (determined by reading the headers of the incoming frame), the node relays each pulse received in the control channel. Basically, upon detecting the emergence of a pulse (instead of after receiving a whole pulse), the relaying node starts its relayed pulse. A relayed pulse, however, has a shorter active phase than the original one, which is to prevent the sender of the original pulse from hearing the relayed pulse.

Fig. 2 demonstrates a transaction in the MAC sub-layer with the proposed MAC protocol. Node A is the sender, node B is the receiver, and node C is a hidden terminal. The figure shows the signals in the two channels of the three nodes. Because node C is a hidden terminal, it can only receive signals transmitted by node B. The longer packet is the data packet, while the shorter one is the acknowledgment packet. The pulse denoted by dashed lines is the CTS pulse sent by node B.

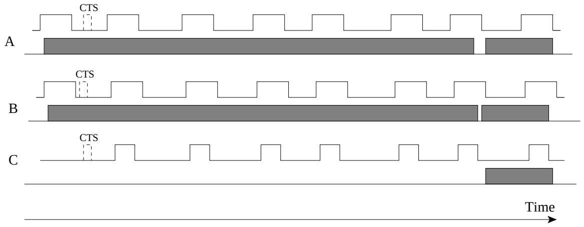

Fig. 2. Signals in the control and data channels of three nodes. Node A is the sender, node B is the receiver, and node C is a hidden terminal.

C. State Transition Diagrams

This subsection describes the proposed protocol with two state transition diagrams. The state transition diagram for a sender with the proposed protocol is shown in Fig. 3. As shown in the diagram, there are five possible states for a sender, which are Monitoring, Contending, Handshaking, Transmitting, and Waiting-for-acknowledgment.

In the Monitoring state, a node monitors the control channel to obtain channel states. When the MAC sub-layer of a node receives a packet from the upper layer, the node becomes active and starts to monitor the control channel. After the control channel has been idle for a specified duration (i.e., when a contention point for medium service arrives), the active node enters the Contending state.

In the Contending state, the node starts a backoff timer for a random backoff. If the node detects a pulse in the Contending state before its backoff timer expires, the node cancels its backoff timer and returns to the Monitoring state. If the backoff timer expires successfully, the node enters the Handshaking state, in which the node generates pulses in the control channel and transmits its data frame in the data channel.

After the node enters the Handshaking state, it expects a CTS pulse in a pulse pause. If the node detects a CTS pulse, it continues to transmit its data frame and transits to the Transmitting state. Otherwise, the node returns to the Monitoring state.

In the Transmitting state, the node transits to the Waiting-for-acknowledgment state after the frame is fully transmitted. However, if the node in the Transmitting state detects a pulse in the control channel, it aborts its transmission and returns to the Monitoring state.

In the Waiting-for-acknowledgment state, the node expects an acknowledgment from the receiver. After the reception of an acknowledgment or the expiration of a timer in the Waiting-for-acknowledgment state, the node goes back to the Monitoring state. If the node receives no acknowledgment for its packet, it will retransmit the packet.

The state transition diagram for a receiver is shown in Fig. 4. As shown in the figure, a receiver may be in one of its five states, which are Monitoring, Determining, Handshaking, Receiving, and Acknowledging. After a node detects an incoming frame, it goes to the Determining state.

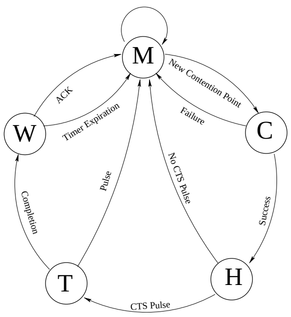

Fig. 3. Sender State Transition Diagram

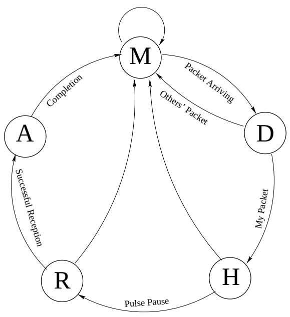

Fig. 4. Receiver State Transition Diagram

In the Determining state, the node checks the MAC header of the incoming frame to determine if the frame is addressed to it. If the frame is addressed to it, the node enters the Handshaking state, in which the node continues to receive the frame and starts to relay the pulses in the control channel. However, if the frame is not addressed to the node, the node goes back to the Monitoring state.

In Handshaking, the node starts to transmit a CTS pulse as soon as the pulse in the control channel enters its pause phase. After finishing sending the CTS pulse, the node transits to the Receiving state. However, if no pulse pause is detected, the node returns to the Monitoring state after the sender aborts its transmission due to the absence of a CTS pulse.

In the Receiving state, if the node receives the frame without errors, it enters the Acknowledging state, in which the node sends back an acknowledgment for the received packet. Otherwise, the receiver transits to the Monitoring state.

After finishing transmitting the acknowledgment frame in the Acknowledging state, the node stops relaying pulses in the control channel and returns to the Monitoring state.

D. Collision Avoidance and Detection

A fundamental task of a MAC protocol is to avoid collisions. Major sources of collisions in wireless networks are hidden terminals. With virtual carrier sense, a receiver uses a CTS *frame* to reserve the medium (i.e., to deal with the hidden terminals) and meanwhile notify the sender of a clear channel. “Busy tone” is another technique used by some existing protocols [2], [9] to address the hidden terminal problem.

Although busy tone is usually more effective in dealing with hidden terminals, it does not carry any address information. Therefore, if two senders try to initiate transmissions at the same time, the sender with an unready receiver may still hear the busy tone sent by the ready receiver of the other sender. In such a case, one of the senders will cause a collision at its intended receiver.

The proposed protocol employs CTS pulses of random lengths to address the false clear-channel notification problem of the busy tone technique. When a sender sends out a data packet, it includes a random integer number in the frame header. After the receiver receives the header of the frame, it sends back a CTS pulse of a length determined by the integer in the header. Only if a sender detects a CTS pulse of the expected length, does the sender continue to transmit.

However, no contention-based MAC protocol can completely avoid collisions. With physical carrier sense, two nodes in random backoff may draw similar delays. With virtual carrier sense, control frames may be lost to neighbors. In addition, busy tone may give false clear-channel notification to a sender. All these are sources of packet collisions.

The proposed protocol employs pulses of random-length pauses to detect current or potential packet collisions in wireless packet networks. When two nodes draw similar backoff delays, they may be unaware of each other and simultaneously start transmitting signals. If neither receiver of the two senders can correctly read their packets due to collisions, neither of them will send back a CTS pulse. In such a case, both senders will abort their transmissions and the collision is resolved. If only one of the two receivers can correctly read its packet, the sender of the other receiver will, in general, abort its transmission due to the absence of a legitimate CTS pulse. The collision is therefore also resolved.

Another case is that both senders may receive a legitimate CTS pulse if both of their receivers can correctly read their frames. However, one of the senders still needs to withdraw in such a case. An example is shown in Fig. 5. In this example, both senders B and C may receive a legitimate CTS pulse from their receivers A and D, respectively, even if they start transmitting their packets at the same time. However, if node B has a shorter packet, then node A will start transmitting its acknowledgment when node C is still transmitting its data packet to node D. A collision may therefore occur at node B. Similarly, if node C has a shorter packet, then a collision may occur at node C. Therefore, one of them needs to withdraw if nodes B and C start to transmit at the same time.

With the proposed MAC protocol, the random-length pauses in the pulses of node B and node C are able to resolve the potential collision. With pauses of random lengths, the pulses of nodes B and C will desynchronize with each other in their active phases as they pass on. After the desynchronization, one sender, such as B, will detect the other and then abort its transmission.

III. PROTOCOL EVALUATION

The proposed MAC protocol is named “PulseAcc” due to the essential roles that pulses play in the protocol. This section presents the evaluation results for PulseAcc. The evaluation has been conducted with extensive simulations using ns-2 [14].

A. Simulation Configuration Details

The PulseAcc protocol implemented in our simulations takes the following parameters for its pulses. The active phase of a pulse has a length of $50\mu s$, while the size of the CTS

Fig. 5. Senders B and C may transmit at the same time without causing collisions. However, one of them still needs to withdraw to avoid a potential future collision that involves their receivers' acknowledgment packets.

window is 150μs. Additionally, the residual pause of a pulse is randomly drawn from a window of 50μs. The length of a CTS pulse in the implemented PulseAcc protocol is randomly drawn in the set of 20, 40, 60, 80, and 100μs.

We have also compared the proposed protocol with two other existing protocols in our evaluations. One is the IEEE 802.11 DCF, which only uses in-band control frames. The other is the RI-BTMA protocol [9], which, like PulseAcc, uses a single control channel. RI-BTMA employs a single busy tone for a receiver to deliver clear-channel information and reserve medium.

In the RI-BTMA protocol implemented in our simulations, data packets are acknowledged and retransmitted when lost, as in the other two protocols. In addition, an initiating sender in RI-BTMA also generates single tone signals when receiving the acknowledgment packet, which is to suppress the hidden terminals of its receiver.

Another important detail of our simulations is that we have used "blank" broadcast packets of small intervals to simulate pulses and tones in the control channel. "Blank" means that these packets do not carry address or other information. When a node receives a blank packet at the right power level (i.e., above the carrier sense threshold) in the control channel, it detects a pulse or tone signal, while the length of a pulse signal is "measured" as the time duration in which blank packets continuously flow in.

In our simulations, the ad hoc network has 50 nodes in an area of 500 by 500 square meters. The link rate is 2 Mb/s. For RI-BTMA and PulseAcc, the control channel takes the same power level as that of the data channel, which is 0.025 watts. This power level gives each node a carrier-sense range of about 300 meters with the default power threshold settings of ns-2. There are a maximum number of 25 randomly-initialized CBR background flows in the ad hoc network. In addition, the routing protocol in our simulations is the Dynamic Source Routing protocol (DSR).

B. Simulation Results

We first examined how these MAC protocols performed as the traffic load varied in the network and nodes were stationary. In a series of simulations, the packet intervals of background traffic varied from 1.0 to 0.0625s with a decrease factor of 0.5 and the packet size was 512 bytes. A test flow, however, kept its packet interval constant at 0.25s to monitor the actual throughput that it could obtain in each case of network load. Fig. 6 shows the percentage of the packets of the test flow that are successfully received by the flow receiver as network load varies (for easy reading, we convert packet intervals to flow rates, which determine the network load).

The three protocols have similar performance when the network load is light, as shown in Fig. 6. In particular, when the packet interval of the background traffic is 1.0 or 0.5s, the flow throughput is almost one hundred percent with all the three protocols. However, when the packet interval decreases to 0.25s, IEEE 802.11 DCF shows a deep throughput decrease to about 20$\%$. In the same case, the throughput of RI-BTMA decreases to about 75$\%$, while PulseAcc keeps its throughput almost at 100$\%$. Similar one-step deep decrease happens to RI-BTMA and PulseAcc when the packet interval further goes down to 0.125s, as shown in Fig. 6. After the deep decrease, each protocol shows relatively flat changes.

The simulation results are shown in Fig. 7 for the case in which nodes take random waypoint movement and have a minimum and a maximum speed of 1 and 10m/s, respectively (the average pause time is 0.5s). Node movement, in general, may cause difficulties to both medium access control and routing in a network and thus decreases network throughput. However, node movement may also increases the throughput of a network. For example, node movement may create new paths or connects broken paths in a network. In addition, node movement may alleviate the medium contention at a hot spot in a network. The finally demonstrated impact of node movement on a network is determined by the interactions of all these factors in the whole network. As shown in Fig. 7, after nodes become mobile, all the three protocols show enhanced throughput at the higher end of network load but lowered throughput at the lower end. The gaps between PulseAcc and RI-BTMA, however, increase in almost all the cases, as shown by Fig. 6 and Fig. 7.

The flow throughput has relatively flat changes after the network load reaches a specific level, as shown in Figs. 6 and 7. These flat changes appear when the network has already had saturation traffic, which is indicated by the low throughput shown in the figures. In a network already saturated with traffic, the number of packets served by the medium in a time unit does not change much as the traffic load further increases; the excessive packets are mostly dropped by the link queues of the flow initiators. In such a case, the actual bandwidth that a flow obtains in the network does not change dramatically.

The simulation results are shown in Fig. 8 for the case in which the maximum node speed is increased to 20m/s in our network. In this case, the flow throughput decreases whatever the network load is. As shown in Fig. 8, the throughput decreases significantly as compared to that in the preceding case of a maximum node speed of 10m/s. This decrease is particularly true for IEEE 802.11 DCF and RI-BTMA, whose throughput, on average, drops almost by half. PulseAcc, however, shows slighter drops on average.

One type of control overhead for the IEEE 802.11 DCF is the exchange of control frames such as RTS and CTS frames. A bigger packet size has the possibility of reducing such overhead for the IEEE 802.11 DCF due to the reduced number of packets for delivering the same amount of data. However, bigger packets cause heavier medium waste when collisions happen. The simulation results are shown in Fig. 9 for the case of a maximum node speed of 10 m/s but the packet size is increased to 1024 bytes (the amount of data delivered in the network is kept the same). As shown in the figure, the performance gains of the proposed protocol do not change significantly in this case of bigger packets.

The above results are conformed to by the number of collisions detected by PulseAcc in the whole network. As

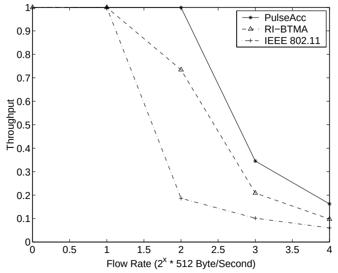

Fig. 6. Throughput versus Network Load (Stationary Nodes)

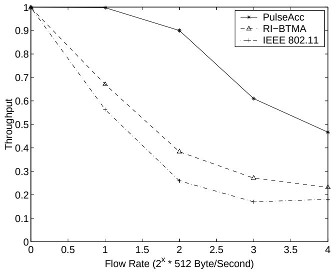

Fig. 7. Throughput versus Network Load (Max. Node Speed: $10m/s$)

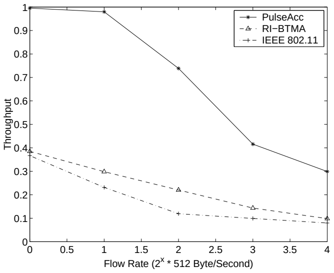

Fig. 8. Throughput versus Network Load (Max. Node Speed: $20m/s$)

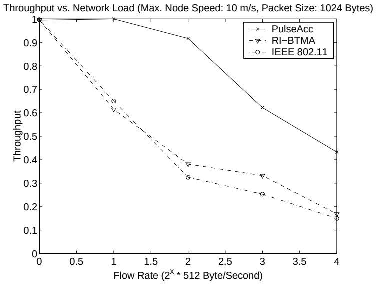

Fig. 9. Throughput versus Network Load (Max. Node Speed: $10m/s$, Increased Packet Size)

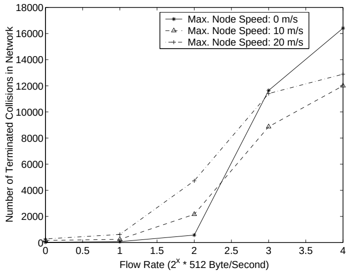

Fig. 10. The Number of Collisions Detected in the Network By PulseAcc (The Impact of Node Movement)

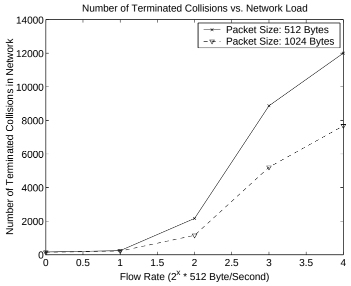

Fig. 11. The Number of Collisions Detected in the Network By PulseAcc (The Impact of Packet Size)

shown in Fig. 10, the number of detected collisions in the case of a maximum node speed of $10m/s$ is consistently lower than that in the case of a maximum node speed of $20m/s$. As shown in Fig. 7 and Fig. 8, the flow throughput in the $10m/s$ case is consistently higher than that in the $20m/s$ case. A higher number of collisions in a network indicates severer medium contention in the network, which decreases the medium utilization and hence network throughput.

Also shown in Fig. 10 is that as network load goes from low to high, the number of detected collisions in the case of stationary nodes is first lower and then higher than those in the other two mobile-node cases. As shown in Figs. 6, 7, and 8, the flow throughput in the stationary-node case is first higher and then lower than that in the other two cases. The results shown in Fig. 10 therefore also conform to the results shown in Fig. 6.

The number of collisions detected in the network is also shown in Fig. 11 for the case in which the packet size increases to 1024 bytes. As shown in the figure, the number of collisions detected in the network is greatly reduced in the case of bigger packets, which is what we expect because the number of packets used to deliver the same amount of data halves in such a case.

IV. SUMMARY

This paper presents a new MAC protocol that effectively and efficiently detects collisions in a fully distributed way in wireless networks. The basic approach is to use out-of-band pulses of random-length pauses. The active phases of pulses signal a busy data channel, while the random-length pauses enable two or more transmitting nodes to detect each other and thus resolve current or potential collisions. Our comprehensive simulation results have shown that the proposed MAC protocol has the capability to significantly improve the throughput of wireless packet networks.

REFERENCES

[1] L. Kleinrock and F. A. Tobagi, “Packet switching in radio channels: Part i - carrier sense multiple-access modes and their throughput- delay characteristics,” *IEEE Transactions on Communications*, vol. 23, pp. 1400–1416, 1975.

[2] F. A. Tobagi and L. Kleinrock, “Packet switching in radio channels: Part II - the hidden terminal problem in carrier sense multiple access and the busy tone solution,” *IEEE Transactions on Communications*, vol. 23, pp. 1417–1433, 1975.

[3] J. F. Kurose and K. W. Ross, *Computer Networking, A Top-Down Approach Featuring the Internet, 2nd Ed.* New Jersey: Pearson Education, 2002.

[4] A. Colvin, “CSMA with collision avoidance,” *Computer Commun.*, vol. 6, pp. 227–235, 1983.

[5] P. Karn, “MACA - a new channel access method for packet radio,” in *Proc. of the 9th ARRL Computer Networking Conference*, Ontario, Canada, 1990.

[6] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, “MACAW: a medium access protocol for wireless LANs,” in *Proc. of the ACM SIGCOMM*, London, United Kingdom, August 1994.

[7] C. L. Fullmer and J. J. Garcia-Luna-Aceves, “Floor acquisition multiple access (FAMA) for packet-radio networks,” in *Proc. of the ACM SIGCOMM*, September 1995.

[8] K. Xu, M. Gerla, and S. Bae, “How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks?” in *Proc. of the IEEE GLOBECOM*, Taipei, Taiwan, November 2002.

[9] C. Wu and V. O. K. Li, “Receiver-initiated busy-tone multiple access in packet radio networks,” in *Proc. of the ACM SIGCOMM*, Stowe, Vermont, August 1987.

[10] Z. J. Haas and J. Deng, “Dual Busy Tone Multiple Access (DBTMA) - a multiple access control scheme for ad hoc networks,” *IEEE Transactions on Communications*, vol. 50, pp. 975–985, June 2002.

[11] J.L.Sobrinho and A.S.Krishnakumar, “Real-time traffic over the IEEE 802.11 medium access control layer,” *Bell Labs Technical Journal*, pp. 172–187, 1996.

[12] S. Singh and C. S. Raghavendra, “Pamas–power aware multi-access protocol with signalling for ad hoc networks,” *ACM SIGCOMM Computer Communication Review*, pp. 5 – 26, 1998.

[13] J. P. Monks, V. Bharghavan, and W. W. Hwu, “A power controlled multiple access protocol for wireless packet networks,” in *Proc. of the IEEE INFOCOM*, Anchorage, Alaska, April 2001.

[14] The network simulator - ns-2. [Online]. Available: http://www.isi.edu/nsnam/ns/