A Wireless MAC Protocol With Collision Detection

Jun Peng, *IEEE Member*, Liang Cheng, *IEEE Member*, Biplab Sikdar, *IEEE Member*

Abstract— **The most popular strategies for dealing with packet collisions at the Medium Access Control (MAC) layer in distributed wireless networks use a combination of carrier sensing and collision avoidance. When the collision avoidance strategy fails such schemes cannot detect collisions, and corrupted data frames are still transmitted in their entirety, thereby wasting the channel bandwidth and significantly reducing the network throughput. To address this problem, this paper proposes a new wireless MAC protocol capable of collision detection. The basic idea of the proposed protocol is the use of pulses in an out-of-band control channel for exploring channel condition and medium reservation and achieving *both* collision avoidance and collision detection. The performance of the proposed MAC protocol has been investigated using extensive analysis and simulations. Our results show that as compared with existing MAC protocols, the proposed protocol has significant performance gains in terms of node throughput. Additionally, the proposed protocol is fully distributed and requires no time synchronization among nodes.**

Keywords: MAC, wireless, collision detection, collision avoidance, CSMA, CSMA/CA

I. Introduction

Due to their ease of deployment and simplicity, distributed MAC protocols such as the IEEE 802.11 Distributed Coordination Function (DCF) are widely used in computer networks to allow users to statistically share a common channel for their data transmissions. In wireless networks, a critical drawback of distributed MAC protocols is the inability of nodes to detect collisions while they are transmitting. As a result, bandwidth is wasted in transmitting corrupted packets and the achieved throughput degrades. This situation is exacerbated as the number of nodes in the network increases since now the rate of collisions increases. To address this issue, this paper proposes a distributed MAC protocol capable of detecting collisions in wireless networks which outperforms existing MAC protocols.

The Aloha protocol [1] was the first MAC protocol proposed for packet radio networks. With pure Aloha, a node sends out a packet immediately upon its arrival at the MAC sub-layer and a collided packet is retransmitted with a probability p immediately or after each packet transmission time. CSMA/CD (carrier sense multiple access with collision detection) [2] employs two mechanisms to enhance the medium utilization in *wired* local area networks (LANs): "carrier sense" and "collision detection". Carrier sense requires a node to listen before transmitting, and collision detection requires a node to transmit and listen at the same time for terminating a possible collision. Although CSMA/CD has been proved to be very successful in wired LANs, it cannot be directly employed in wireless networks because of two problems. The first is the hidden terminal problem [3]. Two mutually hidden terminals are two nodes that cannot sense each other (due to the distance or obstacles between them) but can still interfere with each other at a receiver. With hidden terminals, carrier sense alone cannot effectively avoid collisions. The other problem for CSMA/CD in wireless networks is that in the same wireless channel, the outgoing signal can easily overwhelm the incoming signal due to high signal attenuation in wireless channels. This problem makes it difficult for a sender to directly detect collisions in a wireless channel.

Some existing MAC protocols [4], [5], [6], [7], [8] depend on in-band control frames for exploring the possible future channel condition for a data frame and also for reserving the medium for the data frame. However, when the collision avoidance strategy fails, a corrupted data frame is still fully transmitted. Another category of protocols [3], [9], [10] uses one or more out-of-band control channels to avoid collisions. These protocols are more effective in dealing with hidden terminals and thus reduce the probability of collisions in a network. However, they are incapable of detecting collisions either and if the collision prevention strategies of these protocols fail, the collided data frames are still transmitted in their entirety.

To address the collision detection problem in distributed wireless networks, this paper proposes a new MAC protocol using pulses in a narrow-band control channel. The control channel reserves the medium around the transmitting nodes while data is sent in a separate channel. To avoid any confusion, we note that the control channel pulses in the proposed protocol are quite different from those used in the physical layer of Ultra-Wide-Band (UWB) wireless networks. Compared with the pulses in the data channels of UWB networks, the pulses in the control channel of the proposed protocol have different sizes, structures, and purposes. The pulses used by the proposed protocol are for controlling the medium access to a *single* data channel, but not for high-rate transmissions or channel division multiple access. These pulses are significantly larger than those in UWB networks and also have a different structure with random-length silent phases. The proposed protocol uses the pulses to accomplish two objectives simultaneously. One is collision avoidance, which is basically channel condition exploration and medium reservation as done by traditional wireless MAC protocols such as the IEEE 802.11 DCF. The other objective accomplished by the pulses is "live" collision detection. "Live" detection means that when a collision happens, it is detected almost immediately instead of being detected after the end of the transmissions.

The performance of the proposed pulse-based MAC protocol is investigated with extensive analysis and simulations. Our results show that the proposed protocol has significant performance gains over existing wireless MAC protocols in terms of node throughput in a distributed wireless network. In particular, the gains can reach more than fifty percent when the network load is high and hidden terminals exist.

The rest of the paper is organized as follows. Section II introduces the background and related work. Section III presents the details of the proposed MAC protocol. Section IV introduces an analytic model to evaluate the saturation throughput of the proposed protocol in one-hop networks. Section V analytically compares the proposed protocol with some existing MAC protocols in general multihop networks. Section VI investigates the bandwidth required for the control channel. Section VII evaluates the proposed MAC protocol with extensive simulations and compares it with existing protocols. Finally, Section VIII concludes the paper.

II. Background And Related Work

A. Carrier Sensing and Collision Avoidance

The most widely used mechanism to avoid collisions in contention-based medium access control is probably "carrier sensing" [11], which is used in both wired and wireless networks. We now describe the drawbacks associated with this mechanism that motivate the development of a scheme with collision detection. With carrier sensing, a node listens before it transmits. If the medium is busy, the node defers its transmission. After the medium has been sensed idle for a specified amount of time, the node usually takes a random backoff before transmitting its frame. The random backoff is for avoiding collisions with other nodes that are also contending for the medium.

Besides the "physical" carrier sensing technique introduced above, the IEEE 802.11 DCF also employs a technique called "virtual" carrier sensing. The virtual carrier sense technique relies on in-band control frames to deal with hidden terminals. Before sending a data frame into the idle medium after proper deferrals and backoffs, a source sends out a Request To Send (RTS) frame to contact the receiver and reserve the medium around the source. If the receiver receives the RTS frame and its channel is determined to be clear, the receiver sends out a Clear To Send (CTS) frame to respond to the sender and reserve the medium too. The data transmission then begins if the handshake and medium reservation process succeeds.

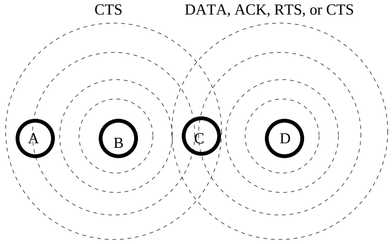

Fig. 1. The Chained Hidden Terminal Phenomenon

Several situations may cause difficulties to the virtual carrier sensing technique. One of them is the "chained" hidden terminal phenomenon. Basically, in a data transaction in the MAC layer, the CTS frame sent by a receiver to suppress the hidden terminals of the initiating sender may be lost at the receiver's neighbors due to the receiver's own hidden terminals. In such a case, some hidden terminals of the initiating sender may not be suppressed. An example is shown in Fig. 1 where node A is the initiating sender and node B is the receiver. The CTS frame generated by node B is corrupted at node C (a hidden terminal of node A) by the signals of node D, which is a hidden terminal of node B.

Node mobility may also limit the effectiveness of the virtual carrier sensing technique with a small probability. With virtual carrier sensing, only nodes which have received the medium reservation message know when to defer. Therefore, when a node newly moves into a neighborhood and misses the preceding reservation information, it becomes an un-suppressed hidden terminal to an on-going data transaction.

Another phenomenon that may impact virtual carrier sensing is that the interference range of a node can be larger than its data transmission range [12]. Therefore, even if a node is out of the range of another node for successfully receiving its CTS frame, the node may still interfere with the other node's data reception.

A more effective way to suppress hidden terminals is to use an out-of-band control channel [3], [9]. With a single data channel, control information cannot be delivered when the data frame is in transmission. With an additional control channel, however, control signals can always be present whenever necessary, which improves the ability of hidden-terminal suppression.

B. Spectrum Reuse and The Capture Phenomenon

The radio spectrum needs to be spatially reused in a multihop wireless network for improving network throughput. Better spectrum reuse allows more transmissions to go on simultaneously in the network without collisions. A phenomenon closely related to spectrum reuse is "capture" which implies that when two frames collide at a receiver in a wireless network, one of the frames may still be correctly decoded if the received power of the frame is higher than that of the other

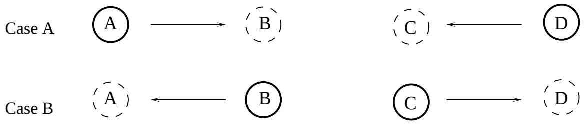

Fig. 2. Collisions Involving Capture

by a threshold. However, as we now show, the capture effect is not sufficient to eliminate collisions and collision detection is required to prevent bandwidth wastage on corrupted frames.

To illustrate the possibility of collisions in the presence of capture effect, two scenarios are shown in Fig. 2 (nodes are in a line for easy demonstration). In the first case, nodes A and D are the initiating senders, while nodes B and C are their receivers, respectively. In the second case, nodes B and C are the senders and nodes A and D are their receivers, respectively. In these two cases, assuming same transmission power levels and ambient noise, captures for the data frames may easily happen at the receivers because the senders are much closer to their receivers than the interference sources.

However, for combating high link error rates, acknowledgments for data frames are widely used in the MAC sub-layer of wireless networks. Therefore, interference may come not only from the initiating senders but also from their receivers. In both cases shown in Fig. 2, the two senders have to finish their transmissions almost at the same time for all the data and acknowledgment frames to be received without collisions. For example, in case A shown in Fig. 2, if node A finishes its data transmission earlier than node D, then node B will send its acknowledgment frame to node A while node C is still receiving the data frame from node D. A collision may therefore easily occur at node C. Similarly, if node D finishes its transmission earlier, node B may easily have a collision. The same thing is true for case B. The corrupted frame, however, will be an acknowledgment instead of a data frame.

In reality, two nodes may not finish their transmissions at the same time, since their frames may have different sizes and their transmissions may begin at different times. Thus collision detection is important in these cases to terminate the colliding transmissions.

C. Related Work

The hidden terminal problem was probably the earliest problem addressed by an out-of-band channel in medium access control for wireless packet networks. The Busy Tone Multiple Access (BTMA) protocol [3] and the Receiver-Initiated BTMA (RI-BTMA) protocol [9] use a single control channel to suppress hidden terminals. The DBTMA (Double Busy Tone Multiple Access) protocol [10], however, uses two control channels to address the hidden terminal problem and improve the spatial reuse of radio spectrum.

Priority scheduling is another topic in medium access control that may borrow assistance from an out-of-band control channel. Some protocols such as [13] and [14] rely on in-band control frames for priority scheduling at the MAC sub-layer. The protocol in [15] relies on the duration of a "black burst" to deliver the priority information for a real-time packet. The Busy Tone Priority Scheduling (BTPS) protocol [16] uses double busy tones to ensure medium access privileges for high-priority packets.

The Power Aware Multi-Access with Signaling (PAMAS) protocol [17] uses a separate signaling channel to power off nodes that are not actively transmitting or receiving packets for the purpose of saving battery energy. The Power Controlled Multiple Access (PCMA) protocol [18] employs control signals with interruptions, which are also called pulses, for improving spatial reuse of radio spectrum. In PCMA, an active receiver broadcasts its noise tolerance information from time to time in an out-of-band control channel. Each broadcast is a short segment of single-tone signal with noise tolerance information encoded in its power level. The protocol proposed in this paper also broadcasts periodical pulses in its control channel. However, unlike the pulses in PCMA, the pulses in the proposed protocol have random-length pauses designed to address a different problem, which is collision detection in wireless networks.

Finally, although the schemes in [19] and [20] (HiperLAN) also aim at addressing the collision detection problem in wireless networks, they were designed for wireless LANs. These schemes share one basic idea with CSMA/CA, which is transmitting a short control frame to check for collisions before a data packet is transmitted. However, if a collision occurs on a *data* frame, the collision will not be detected. The scheme proposed in this paper, however, is designed for general wireless packet networks and it is capable of live collision detection.

III. The Proposed MAC Protocol

A. Protocol Basics

The MAC protocol proposed in this paper assumes that each node has the ability to simultaneously transmit on two channels, the control and data channels, with two antennas and their associated communication circuitry. The control channel has a much smaller bandwidth as compared to the data channel and is used for transmitting medium reservation related signals while the data channel is for transmitting the data and acknowledgments. Instead of relying on bit-based frames, the control channel employs pulses to deliver control information. The pulses in the control channel are single-frequency waves with random-length pauses (more details of the pulses are given in Section III.B). In the proposed protocol, pulses only appear in the control channel and the control channel only carries pulses. When a node is an active sender or receiver in the data channel, it monitors the control channel all times except when it itself is transmitting in the control channel. If a node is transmitting in the data channel but detects a pulse in the control channel, it aborts its transmissions.

To describe the operation of the protocol we consider what happens when the MAC sub-layer at a node, say node A, receives a packet to transmit to node B. Before node A can transmit, it first listens to the control channel to make sure that it is idle. If the control channel is found idle for a period of time longer than the maximum pause duration of a pulse, node A starts a random backoff timer whose value is drawn from the node's contention window. If the node detects no pulse before its backoff timer expires, it proceeds to transmit the packet upon the expiration of its backoff timer. Otherwise, the node cancels its backoff timer and keeps monitoring the control channel.

As soon as the backoff timer of node A expires, it starts to transmit pulses in the control channel along with the packet in the data channel. Once the node has finished transmitting the frame header in the data channel, it expects the intended receiver, node B, to have received the information and reply with a CTS pulse in the control channel. The CTS pulse is transmitted by node B during a pause in the pulses being sent by node A in the control channel. If node A does not obtain the expected CTS pulse in the following pause period after the frame header is transmitted, node A aborts its transmissions in both channels. If node A obtains the expected CTS pulse, it keeps transmitting. Node A, however, may still abort its transmissions after obtaining the expected CTS pulse if it detects a pulse of another node in one of its pulse pauses later, which indicates a colliding situation. If the node aborts its transmissions due to the lack of the expected CTS pulse or the detection of a pulse of another node, it doubles its contention window and then returns to monitor the control channel.

After node A fully transmits the packet, it expects an acknowledgment from the receiver. If the node does not obtain the expected acknowledgment, it doubles its contention window and starts to monitor the control channel again to look for a retransmission opportunity. The whole process repeats until either node A obtains an acknowledgment for the packet or the retry limit is reached. The node discards the packet in the latter case and resets its contention window to the minimum size in both cases.

The above description is for the case of a unicast packet. In the case of a broadcast packet, the proposed protocol uses the basic CSMA protocol as in the IEEE 802.11 DCF. The rest of the paper focuses on the transmission of unicast packets.

B. The Contention and CTS Pulses

As shown in Fig. 3, a contention pulse in the proposed protocol consists of two phases, an active phase of a fixed length and a pause phase of a random length. Busy-tone waves are only transmitted in the active phase of a pulse. The active phase of a contention pulse signals a busy data channel, while the pause phase is for collision detection.

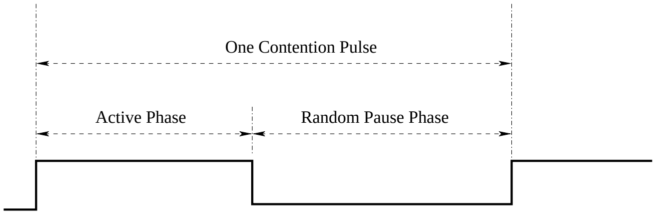

Fig. 3. A contention pulse consists of two phases, an active phase of a fixed length and a pause phase of a random length. Busy-tone waves are transmitted in the control channel in the active phase only.

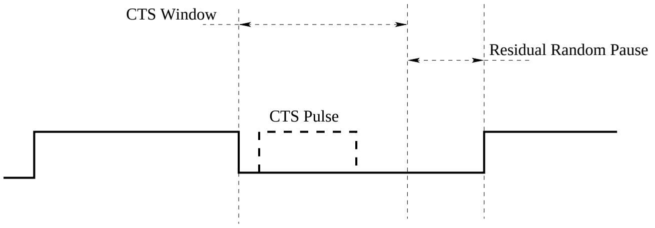

Fig. 4. A CTS pulse is delivered in a pause phase of a contention pulse.

While a node is transmitting data in the data channel, it monitors the control channel in the pause phases of its pulses. There is usually a transition delay of a couple of microseconds for an antenna to switch its state. This transition delay is however small as compared with the duration of a pulse, which is usually several tens of microseconds. Similarly, the detection time of a pulse is also trivial as compared with the duration of a pulse. If a node detects a pulse during one of its pulse pauses, the node stops transmitting in both channels.

A CTS pulse, which delivers the clear channel signal, is slightly different from a contention pulse. Recall that a node sends a CTS pulse in response to a data frame that it receives. A CTS pulse does not have a pause phase and the length of its active phase is specified by a field in the received MAC header of the data frame, which contains an integer randomly selected by the initiating sender. A CTS pulse is sent back to the initiating sender during the pause phase of one of the pulses of the initiating sender. In the rest of the paper, pulses, unless specified otherwise, denote contention pulses.

Fig. 4 demonstrates how a CTS pulse is delivered in a pulse pause. A sender waiting for a CTS pulse segments its pulse pause into two parts. One is the CTS window, while the other is the residual pause of a random length. The sender regards a CTS pulse legitimate only if the CTS pulse is of the expected length and received in the CTS window (note that the size of the CTS window is fixed and a CTS pulse is designed to fit in this window).

For dealing with hidden terminals, contention pulses are also "relayed" by a data-frame receiver after the receiver checks the received data frame header and determines that the frame is intended for it. This ensures that the nodes in the vicinity of the receiver are also aware of the ongoing transmission. A receiver starts its relayed pulse upon the detection of the arrival of a new pulse. Since the length of the active phase is

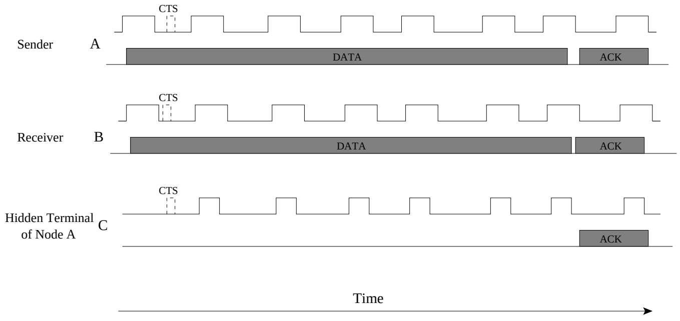

Fig. 5. Signals in The Control and Data Channels of Three Nodes

fixed and the same for all nodes, the receiver is already aware of the length of the pulse to be relayed. The active phase of a relayed pulse is, however, shorter than that of the original pulse by a couple of microseconds. When the relayed pulse is being transmitted, the source of the original pulse is still transmitting its own pulse. Therefore, the source of the original pulse will not detect the relayed pulse. A pulse sender in the rest of the paper denotes a node that is either generating original pulses or relaying pulses in the control channel.

With the loose synchronization mechanism introduced above (i.e., the simultaneous relay of a contention pulse by the receiver though for a few microseconds lesser), a sender and its receiver do not need additional strict synchronization for pulse relaying, which is a great advantage in distributed wireless networks. The lack of strict synchronization between a sender and its receiver has, however, one consequence, which is that the first contention pulse of the sender is not relayed by the receiver. This is because a receiver relays pulses only after it receives and checks the data frame header and ensures that it is the intended receiver.

If a hidden terminal starts to transmit before the receiver starts to relay pulses, the loose synchronization between the sender and the receiver will be disrupted by the pulses of the hidden terminal. In addition, the loose synchronization may also be lost due to reasons such as signal fading in the control channel. If the loose synchronization is lost, the sender will receive pulses in its pulse pauses and thus will abort. In such a case, the sender will initiate new transmissions later.

Fig. 5 demonstrates a transaction in the MAC sublayer with the proposed protocol. Node A is the sender, node B is the receiver, and node C is a hidden terminal of node A. The figure shows the signals in the two channels of the three nodes. Because node C is a hidden terminal of node A, it can only receive signals transmitted (or relayed) by node B. Note that if a node receives pulses from both node A and node B, it still recognizes the pulses from node A, since node B's pulses are in the "shadow" of node A's pulses in the time domain. Additionally, the pulses carry no bits so that no traditional "collision" happens here.

Finally, the delay before a MAC header is received by the intended receiver determines the minimum time that elapses before a collision may be detected, as inferred from the absence of a CTS pulse. This delay is mainly determined by the frame transmission rate and we now characterize this delay. According to the IEEE 802.11 specifications [8], a general MAC header for a data frame is 30-byte long. The proposed protocol adds another field of 1 byte to deliver the expected length of a CTS pulse. The total MAC header has therefore 248 bits with the proposed protocol. As specified in IEEE 802.11, for a Direct Sequence Spread Spectrum (DSSS) physical layer, the physical layer convergence procedure (PLCP) preamble has 144 bits, while the PLCP header has 48 bits. The total physical layer header is therefore 192-bit long. In such a case, a MAC header can be completely received after 440 bits. If the data frame is transmitted at 1 Mbps, then 440 bits can be transmitted in $440\mu s$. The physical layer may also do "whitening" on the payload, which can generate a delay of up to "8 octets", as indicated in the IEEE 802.11 specifications. In such a case, the total delay before a MAC header is received by an intended receiver is therefore about $504\mu s$.

C. Collision Avoidance and Detection

This subsection further explains how the proposed MAC protocol achieves collision avoidance and collision detection. As in the CSMA case, the proposed protocol considers it a potential colliding situation when a transmitting node detects another transmitting node. For collision avoidance, the proposed protocol uses handshake and medium reservation procedures like those used by traditional wireless MAC protocols. The difference is that in the proposed protocol these procedures are moved to the control channel where CTS pulses are used for handshaking and the pulse relay is used for medium reservation. When the collision avoidance fails, the collision detection mechanism comes into play and this is the essential difference between the proposed protocol and other wireless MAC protocols.

To understand how the proposed protocol resolves collisions, we consider the case where two neighboring nodes cause collisions. If two neighboring nodes draw the same backoff delays at a contention point for medium access, they start to transmit signals in the data and control channels almost at the same time. If both receivers of the two senders cannot correctly read the frame headers due to the resulting collision (i.e., the address or another field in the header does not have a legitimate value), neither of them will send back a CTS pulse. Both senders will therefore terminate their transmissions and the collision is resolved automatically. If only one of the two receivers can correctly read the frame header, the sender of the other receiver will, in general, abort its transmissions due to the lack of a legitimate CTS pulse. The collision is therefore also resolved in such a case.

If both receivers can correctly read the frame headers, each of them will send back a CTS pulse with the length specified in the MAC headers of their respectively received data frames. If the two initiating senders do not draw the same CTS length, then the sender that draws a shorter one may not receive a legitimate CTS pulse and thus abort its transmissions.

If both senders receive legitimate CTS pulses, one sender will usually still need to abort its transmissions (since their ACK frames may be interfered or cause interference, as explained in Section II.B). The collision detection mechanism starts to work in such a case. With pauses of random lengths, the pulses of the two senders will desynchronize each other over time. After the desynchronization, the sender with a longer pause will detect the pulse of the other sender and then release both channels. A collision is therefore resolved.

The above description of collision detection is not restricted to two transmitting nodes who are neighbors. As introduced earlier, pulses are relayed by nodes that are receiving data frames intended for them. Therefore, two nodes that are hidden terminals to each other still detect each other if they transmit at the same time.

D. Clarifications

One requirement not explicitly stated in the above descriptions of the proposed protocol is that a pulse should have a length that is much smaller than the length of a data frame. Since pulses are designed for collision detection, pulses should be repeated at a frequency that makes it feasible for a collision to be detected before the colliding transmissions finish by themselves. A small number of, such as five to ten, pulses during each transmission of a data frame is adequate for effective collision detection, as shown in Section IV.

One phenomenon that needs to be mentioned is multipath fading, which occurs when a signal reaches a receiver through multiple paths. Multipath is a common phenomenon in urban areas due to obstacles and reflectors. Multipath may cause fluctuating amplitudes and phases in signals, which are harmful for signal decoding. Pulses, however, are not as sensitive to multipath fading as bit-based frames. First, a pulse has a much longer duration than a bit in a frame. For example, if a data frame has 512 bytes of payload and there are 5 pulses in its transmission duration, then each pulse has a length of at least 819 bits. Second, only the amplitude fluctuation has a significant impact on the pulse detection.

In the proposed MAC protocol, a receiver does not immediately declare the end of the active phase of a pulse when the power in the control channel falls below a threshold; the receiver only does so after the power stays below the threshold for a specified amount of time. With this design, short fadings do not affect the pulse detection. If there are long fadings in the control channel, the data channel might also experience fadings since in real life scenarios the two channels are expected to be in the same allocated band. In such a case, the data frame may not be correctly decodable anyway even with channel coding.

IV. Saturation Throughput

To evaluate the performance of our proposed protocol we now develop an analytic model to evaluate its saturation throughput in one-hop networks. The model uses a mean-value analysis similar to that developed in [21] and [22] for the IEEE 802.11 MAC protocol. We assume that a network with N nodes uses our proposed protocol in the MAC layer to schedule their transmissions. Since we are interested in the saturation throughput, we assume that all nodes always have packets to send. The channel transmission rate is denoted by C bits/sec and the length of each packet is assumed to be L bits.

In order to evaluate the saturation throughput, we first analyze the exponential back-off mechanism associated with the proposed protocol and its associated collision rates. As per the details of the protocol described in Section III, each station begins its backoff process once the channel is sensed idle for a specified period of time, which we denote by T_{idle}. The first attempt at transmitting a given packet is performed using contention window or CW value equal to CW_{min}. For each unsuccessful transmission attempt, CW is doubled until it reaches the upper limit of CW_{max} specified by the protocol or the maximum retransmission limit M is reached. We also use the notation $m = \log_2(CW_{max}/CW_{min})$. We denote the probability that an arbitrary packet transmission results in a collision by p. Then, in the absence of retransmission limits, the probability that $CW = W$ is given by

$$\Pr\{CW = W\} = \begin{cases} p^{k-1}(1-p) & \text{for } W = 2^{k-1}CW_{min} \\ p^m & \text{for } W = CW_{max} \end{cases} \tag{1}$$

where $k \leq m$. Note that when the retransmission limit $M < m$, the contention window does not grow to CW_{max}. Now with probability $1-p$ the first transmission is successful and the average backoff window of such a packet is $CW_{min}/2$. With probability $p(1-p)$ the first transmission fails and the packet is successfully transmitted in the second attempt (using a backoff window of $2CW_{min}$) which adds CW_{min} to the average backoff window seen by the packet. Continuing along these lines for cases with larger numbers of losses, the average backoff window in the saturated case is given by

$$\overline{W} = \begin{cases} \frac{1-p-p(2p)^m}{1-2p}\frac{CW_{min}}{2} & M \geq m \\ \frac{1-p-p(2p)^{M-1}}{1-2p}\frac{CW_{min}}{2} & M < m \end{cases} \tag{2}$$

Assuming that each node has a constant probability of transmission in each idle slot, the probability of a node's transmission in a slot is given by $\tau = 1/\overline{W}$. Then the probability that a node's transmission is successful is the probability that none of the other $N-1$ nodes transmit in the slot, i.e., $1-p = (1-\tau)^{N-1}$. Thus

$$p = 1 - \left(1 - \frac{1-2p}{1-p-p(2p)^{M-1}}\frac{2}{CW_{min}}\right)^{N-1} \tag{3}$$

where we have considered the case $M < m$. To derive the saturation throughput, we observe the system in a unit of time. We denote the rate of transmission attempts by the nodes in a unit time by r_x, the rate of *successful* transmissions by r_s and the rate of collisions by r_c. Now, since each transmission is successful with probability $1-p$, the average number of transmissions per packet is $1/(1-p)$. The average number of transmissions per successful transmission is also given by r_x/r_s. Thus

$$\frac{1}{1-p} = \frac{r_x}{r_s} \tag{4}$$

While each collision may involve a number of stations, to a first degree approximation we assume that each collision involves only two stations. Thus

$$r_c = \frac{r_x - r_s}{2} \tag{5}$$

We denote by T the average cycle time, or the renewal period between two successive transmissions. The cycle time in the case of a successful transmission, T_s is given by

$$T_s = T_{idle} + \overline{W}\delta + T_{data} + T_{ack} \tag{6}$$

where δ is the duration of a backoff timeslot and T_{data} and T_{ack} represent the times required to transmit the data and ACK frames, respectively. The cycle time in the case of a collision T_c is given by

$$T_c = T_{idle} + \overline{W}\delta + T_{cd} \tag{7}$$

where T_{cd} is the collision detection time with the proposed protocol. Note that in the expressions above, we have used $\overline{W}$ instead of the expectation of the minimum of N backoff periods as an approximation. The average cycle time is thus

$$T = (1-p)T_s + pT_c \tag{8}$$

We also have

$$\frac{1}{T} = r_s + r_c \tag{9}$$

Combining Eqns. (4), (5) and (9) we get

$$r_s = \frac{2(1-p)}{2-p}\frac{1}{T} \tag{10}$$

and the saturation throughput η is thus

$$\eta = \frac{2(1-p)}{2-p}\frac{T_{payload}}{T} \tag{11}$$

where $T_{payload}$ is the time required to transmit just the data payload and given by L/C. The per node throughput is η/N and p is obtained by solving Eqn 3.

In order to validate the model above, we now compare its results with simulation results using the ns-2 simulator. For these results, the parameters chosen were $C = 1$ Mbps, $L = 512$ bytes, $CW_{min} = 32$, $CW_{max} = 1024$, $M = 4$, $\delta = 20\mu$s, $T_{idle} = 250\mu$s, ACK frame length of 14 bytes, MAC and PLCP headers of 31 and 6 bytes respectively and physical layer preamble of 144 bits (i.e., the DSSS physical layer parameters specified in IEEE 802.11). In addition, a pulse has an active phase of 50μs, a CTS window of 150μs, and a residual random pause uniformly distributed from 0 to 50μs.

The collision detection time T_{cd} of the proposed protocol depends on the transmission rate of the data frame. An initiating sender expects a CTS pulse after transmitting the frame header. If the sender does not obtain the expected CTS pulse, it stops transmitting. A higher transmission rate therefore makes contention collisions detected faster. In the 1 Mbps case, the MAC header needs about $504\mu s$ to be received (see Section III.B). This is for the case where an initiating sender detects a collision by the lack of a CTS pulse, which is also the dominant mechanism of collision detection for our saturation throughput analysis.

The other case is that an initiating sender may detect a collision by detecting a pulse in one of its pulse pauses after receiving a CTS pulse, which may occur frequently in a multihop network (an example is shown in Case B of Fig. 2). Note that the residual random pause of a pulse is drawn uniformly from 0 to $50\mu s$, so the probability that two nodes draw residual pauses of a difference within $5\mu s$ (i.e., the propagation delay) can be calculated as follows. Let x and y be two independent random variables with a uniform distribution between 0 and 50.

$$p(|x-y| < 5) = 2p(0 < x - y < 5) = 0.19 \tag{12}$$

Therefore, each passing of a pulse pause gives a probability of 0.81 for two transmitting nodes to detect each other. In such a case, a collision on average is detected after the passing of 1.23 pulses. Since the average pulse length (active phase plus pause phase) is $225\mu s$, the time used to detect a collision by pulses is about $780\mu s$ (including the CTS detection time).

Under the assumptions of our analysis all nodes are in the transmission range of each other. Thus two simultaneous transmissions will usually result in the corruption of both the frame headers and thus no CTS pulses will be transmitted. All collisions will then be detected by the absence of CTS pulses and thus $T_{cd} = 504\mu$s. In Figure 6 we plot the saturation throughput per node as a function of the number of nodes in the network. The simulation results shown in the figure were obtained through ns-2 simulations of the same scenario and using the same parameter settings as the analysis. We note the close match between the analytic and simulation results, which validates the model.

V. Performance in More General Cases

Besides the saturation throughput in one-hop networks, it is also of interest to understand how the proposed protocol improves the throughput of a multihop network over protocols like the IEEE 802.11 DCF. We use a "macro" model to achieve this goal.

Instead of focusing on how the medium state changes over time, we focus on the average medium time, $\overline{T}$, spent for successfully delivering a data packet in the network and receiving its acknowledgment frame. When the medium contention is successful for a frame (i.e. the associated backoff timer expires and the channel is idle), the frame is transmitted. From an individual node's perspective, only when the node succeeds in medium contention and starts transmitting does it consume medium time. There is a probability, p_c, that the transmission will experience a collision from direct medium contention, since another node may have an expired backoff timer too. Even if the frame experiences no collision from direct medium contention, it may still experience collision caused by hidden terminals. We denote the "natural" probability of the existence of harmful hidden terminals by p_h, which is the probability that a frame is corrupted by hidden terminals when there is no mechanism such as virtual carrier sensing to suppress hidden terminals. If there is a hidden terminal suppression mechanism, the probability of harmful hidden terminals to a frame is usually reduced. We denote the factor of such a reduction by f $(f < 1)$ (i.e., the harmful-hidden-terminal probability changes from p_h to fp_h).

We first consider the case of CSMA, which does not use RTS/CTS. If we denote the transmission times for a data frame and its acknowledgment frame by T_{data} and T_{ack}, respectively, and the average backoff time in a contention as $\overline{T_{bkoff}}$, then the medium time consumed for a data frame including all its retransmissions is as follows (note that each failed transmission consumes the contention backoff time plus frame transmission time)

$$\overline{T_{CSMA}} = \frac{1}{(1-p_c)(1-p_h)}(\overline{T_{bkoff}} + T_{data}) + T_{ack}, \quad (13)$$

where we assume the ACK frame does not experience collision and the interframe space times are negligible.

We then consider the case of CSMA/CA (i.e., IEEE 802.11 DCF in this paper). The exchange of the RTS/CTS frames with CSMA/CA may have two folds of benefits. One is that when there is a collision caused by medium contention, the collision cost is low because an RTS frame is short. The other benefit is that the RTS/CTS frame exchange reduces the probability of harmful hidden terminals. However, the RTS/CTS exchange generates control overhead too.

With the probability of a contention collision denoted by p_c and the natural probability of harmful hidden terminals denoted by p_h, an RTS frame with a transmission time of T_{rts} consumes a medium time of

$$\overline{T_{rts}} = \frac{1}{(1-p_c)(1-p_h)}(\overline{T_{bkoff}} + T_{rts}) \quad (14)$$

before it is successfully received by the intended receiver. If the RTS is successfully received, we may assume that the CTS will not have a collision at the initiating sender, considering that a CTS frame is short and the RTS has already reserved the medium around the initiating sender. However, there is still a probability, fp_h, that harmful hidden terminals still exist for the data frame (see Section II.A). When there are harmful hidden terminals for the data frame, the RTS/CTS/Data process needs to be repeated. If we denote the transmission time of a CTS by T_{cts}, then the medium time consumed for a data frame including all the retransmissions is

$$\overline{T_{CSMA/CA}} = \frac{1}{1-fp_h}(\overline{T_{rts}} + T_{cts} + T_{data}) + T_{ack} \quad (15)$$

in the CSMA/CA case.

Lastly, we consider the case of the proposed protocol (we name it *PulseAcc* due to the essential roles of pulses in the protocol). There is no RTS/CTS exchange in the data channel with the proposed protocol. Also, if there is a collision, it is terminated. If we denote the average time for detecting a collision by $\overline{T_{cd}}$, then the medium time consumed for a data frame including all its retransmissions is shown as follows

$$\overline{T_{PulseAcc}} = \frac{1}{(1-f'p_h)(1-p_c)}(\overline{T_{bkoff}} + \overline{T_{cd}}) + (hT_{data} - \overline{T_{cd}}) + hT_{ack}, \quad (16)$$

where $f^{'}$ denotes the hidden terminal reduction factor for PulseAcc and h denotes the factor by which the frame transmission times increase with PulseAcc due to the reduced data bandwidth (some bandwidth is used by the control channel in PulseAcc). As analyzed in Section VI, the control channel including the guardband uses at most 2% of the allocated band of an IEEE 802.11 system, resulting in a h of 1/0.98=1.02.

According to the IEEE 802.11 specifications, an RTS frame has a size of 20 bytes, a CTS frame has a size of 14 bytes, and an ACK frame also has a size of 14 bytes. If we consider a DSSS physical layer, the total physical layer header is 192 bits. It is recommended that control frames are transmitted at the basic link rate of 1 Mbps in an IEEE 802.11b system. Therefore, in this case, an RTS frame has a transmission time of 352μs (i.e., $T_{rts} = 352\mu s$), a CTS frame has a transmission time of 304μs (i.e., $T_{cts} = 304\mu s$), and an ACK frame also has a transmission time of 304μs (i.e., $T_{cts} = 304\mu s$). Although the average backoff time $\overline{T_{bkoff}}$ for a contention changes with the load in the network, it impacts the performance of all protocols equally. Using experiments on IEEE 802.11 DCF with the ns-2 simulator [23] and its default configurations, we observed that on an average, a node experienced 100 backoff slots in a saturated situation of more than 50 nodes, resulting in an average backoff delay of 2000μs (a timeslot of 20μs).

The transmission time T_{data} for a data frame varies with the size of the packet and the transmission rate. For illustration we may assume that the packet has a size of 512 bytes and the transmission rate is 1 Mbps (note that RTS/CTS control frames are transmitted at the basic link rate so that a higher transmission rate for the data frame means higher control overhead for the CSMA/CA protocol). With all headers considered, the transmission time for a data frame in this case is about 4800μs.

The transmission rate may also affect the pulse parameters and thus the delay before a collision is detected. If a pulse has an active phase of 50μs, a CTS window of

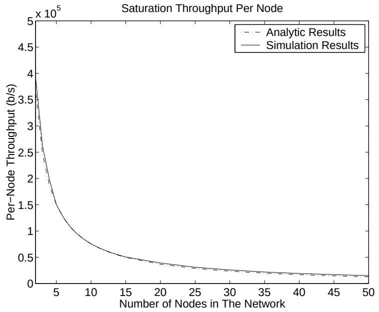

Fig. 6. The Average Throughput of a Node in a Wireless LAN with Saturation Traffic (Analytic and ns-2 Simulation Results)

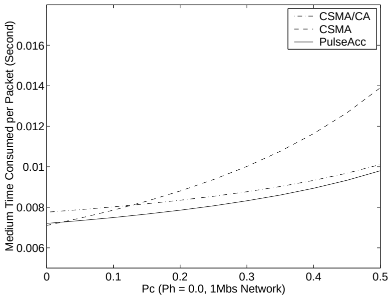

Fig. 7. Medium Time Used for Successfully Delivering a Data Packet ($p_h = 0.0, 1Mbps$)

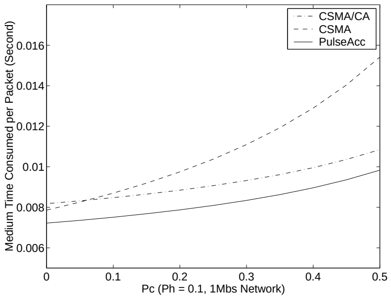

Fig. 8. Medium Time Used for Successfully Delivering a Data Packet ($p_h = 0.1, 1Mbps$)

150μs, and a residual random pause uniformly drawn from 0 to 50μs, then a collision can be detected after 504μs and 780μs in the case of the absence of a CTS pulse and in the case of pulse contentions, respectively, as calculated in Section IV. The collision detection time T_{cd} should therefore have an average value below 642μs in our case of network and pulse parameters.

Fig. 7 shows the average medium time consumed for successfully delivering a data packet in the case of no hidden terminals ($p_h = 0.0$). As shown in the figure, the proposed PulseAcc protocol uses the least amount of medium time among the three protocols for successfully delivering a data packet. Interestingly, due to its control frame overhead (i.e., RTS/CTS), CSMA/CA does not work better than CSMA when the collision probability is low.

Fig. 8 shows the protocol performances for the case in which there are some hidden terminal problems ($p_h = 0.1$). Heuristically, we set the hidden-terminal reduction factor to 0.2 and 0.05 for CSMA/CA and PulseAcc, respectively, due to the limitations of in-band control frames and the power of an out-of-band control channel. These heuristic values do not necessarily reflect reality but are used for illustration. As shown in Fig. 8, the performance gains of PulseAcc over other protocols increase in this case with hidden terminals. In addition, CSMA/CA shows clearer gains over CSMA in this case.

VI. Control Channel Bandwidth

The bandwidth required for the control channel of the proposed protocol is determined by the parameters of the pulses transmitted in the channel. A pulse in the proposed protocol has a fixed-length active phase with single-tone signals and a random-length pause phase without signals. The bandwidth of a pulse is therefore determined by the length of its active phase. According to Fourier theory, the bandwidth of a pulse of an active length T is about $2/T$ if the pulse is properly shaped [24].

If we consider a case where the basic link rate and the data rate are 1Mbps and 10Mbps, respectively, and a data packet has 512 bytes, then the transmission time for a data frame including headers is about $1ms$ (note that it is recommended in the IEEE 802.11 standard that the physical layer preamble and header are transmitted at the basic link rate). A pulse in such a case has a period of about 100μs if there are 10 pulses during each frame transmission. If the pulse is active in one fourth of its duration, then its active length will be about 25μs, which indicates a bandwidth of about 80kHz. A CTS pulse may be shorter than a regular pulse. In our implementation, the shortest CTS pulse is about one half of the active length of a regular pulse, which means a doubled bandwidth of 160KHz.

In addition, a guardband is needed between the control channel and the data channel in the allocated band. To obtain the size of the guardband, we may use the GSM practice as an example. In the GSM-900 system, both forward and reverse links use a band of 25MHz. For each of the 25MHz bands, there are guardbands of 100KHz at both ends [25]. An IEEE 802.11 system also uses a band over 20MHz and therefore it is reasonable for the guardband between the control channel and the data channel in our protocol to be about 200KHz. The total bandwidth required for the control channel, i.e., the pulse bandwidth plus the guardband, is therefore 360KHz in our example. As compared with the 22MHz band specified in the IEEE 802.11b standard, the bandwidth of the control channel is less than 2% of the allocated band (wider bands are considered for other IEEE 802.11 systems).

The required control channel bandwidth will increase if the link speed increases. For example, we may consider an IEEE 802.11g system. In such a system, the maximum link rate is 54Mbps, while the recommend basic link rate is 2Mbps in the case of coexistence with IEEE 802.11b systems. A frame carrying a packet of 512 bytes will be transmitted in about 620μs in such a case. We may use a pulse structure with an active phase of 10μs, a CTS window of 40μs, and a residual pause window of 10μs. In addition, a CTS pulse may be drawn from a pool of 10, 15, 20, 25, and 30μs. In such a case, the control channel will use a bandwidth of about 400KHz (i.e., the pulse bandwidth plus the guardband), which is still less than 2% of the allocated band for an IEEE 802.11 system.

VII. Simulation Results

A. Evaluation Model and Configuration Details

The proposed MAC protocol has been evaluated in both wireless LANs and ad hoc networks using the Network Simulator 2 (ns-2) [23]. If it is not specified otherwise, each node in our simulations always has packets to send and the destination of each packet is randomly drawn among the neighbors of the node. With this saturation traffic model, all traffic starts and stops at the MAC layer and no routing or other upper-layer factors are involved in the simulations. For comparison, the results for three other existing wireless MAC protocols are also shown. One is the CSMA/CA protocol specified in the IEEE 802.11 standard, and another is the CSMA protocol. These two protocols do not use an out-of-band control channel. The third protocol is RI-BTMA [9], which like PulseAcc, uses a single control channel.

In addition, the bandwidth overhead of the control channel is considered in the simulations for PulseAcc and RI-BTMA. As analyzed earlier, PulseAcc uses at most two percent of the allocated band for the control channel including the guardband. For RI-BTMA, we assume that one percent of the assigned band is used for the control channel because RI-BTMA does not use pulses. In the case of 1Mbps link rate, PulseAcc and RI-BTMA have data rates of 0.98Mbps and 0.99Mbps, respectively. In the case of 54Mbps link rate, their data rates are (0.98×54) Mbps and (0.99×54) Mbps, respectively, in the simulations.

In the RI-BTMA protocol implemented in our simulations, data frames are acknowledged and retransmitted when lost, as in the other three protocols. The retry limit is 4 for all protocols. In addition, an initiating sender in RI-BTMA also generates single tone signals when receiving the acknowledgment frame for suppressing the hidden terminals of its receiver. The contention window of a node is adjusted in all protocols using the binary exponential backoff mechanism of IEEE 802.11.

Our implemented PulseAcc protocol uses the following parameters for its pulses in the 1Mbps link rate case. The active phase of a pulse has a length of 50μs, and the size of the CTS window is 150μs. Additionally, the residual random pause of a pulse is drawn from a window of 50μs. The length of a CTS pulse is randomly drawn from the set of 20, 40, 60, 80, and 100μs. In the 54Mbps link rate case, these parameters become smaller accordingly. In particular, the length of the active phase of a pulse is 10μs, the CTS window is 40μs, the residual pause window is 10μs, and a CTS pulse may have a length of 10, 15, 20, 25, or 30μs.

In the ad hoc network in our simulations the nodes are distributed in an area of 500 by 500 square meters. For RI-BTMA and PulseAcc, the control channel uses the same power level as that of the data channel, which is 0.025 watts. With the default configurations of the power thresholds in ns-2, this power level gives each node a data-transmission and carrier-sense (CS) range of about 150 and 300 meters, respectively. In addition, the link rate of the ad hoc network can be either 1Mbps or 54Mbps. In the former case, the basic link rate is 1Mbps too. In the latter case, the basic link rate is 2Mbps, as recommended in the IEEE 802.11g standard in the case of coexistence with IEEE 802.11b systems.

B. Wireless LAN Case

This section presents the simulation results for the wireless LAN case. Fig. 9 shows the LAN throughput versus the number of nodes in the LAN. As shown in the figure, PulseAcc can have more than 5% performance gains over RI-BTMA in the wireless LAN case. The gains of PulseAcc may reach more than 20% over IEEE 802.11 or CSMA. There are several other things that can be observed in Fig. 9. When there are less than five nodes in the LAN, CSMA may have a higher performance than other protocols. This happens because all the other three protocols have their control overhead. RI-BTMA and PulseAcc have reduced data rates due to the control channel overhead, while the IEEE 802.11 (i.e., CSMA/CA in this paper) has the RTS/CTS overhead. However, as the number of nodes in the LAN increases, there are increased collisions in the LAN. In such cases the extra control procedures of the other three protocols start to pay off. The gains of CSMA are therefore either reduced in the IEEE 802.11 case or reversed in the RI-BTMA and PulseAcc cases. Due to its collision detection capability, PulseAcc has a higher performance than RI-BTMA, even though it has a lower data rate than RI-BTMA.

Fig. 10 shows the medium access delay versus the number of nodes in the LAN. The medium access delay for a packet is defined here as the time from the arrival of the packet at the MAC layer to either the successful transmission of the packet or the drop of the packet due to excessive retransmissions. The delay results shown in Fig. 10 conform to the throughput results shown earlier. In general, a protocol introduces less medium access delays when it generates more throughput in the LAN.

Fig. 11 shows the collision rate versus the number of nodes in the LAN. The collision rate is the number of collisions detected by PulseAcc at a node over the number of transmissions at the node in the LAN. As shown in Fig. 11, the collision probability becomes higher when there are more nodes in the LAN, which is expected.

C. Few Hidden Terminals Case

This section shows the simulation results for static ad hoc networks with few hidden terminals. With the default carrier sense (CS) power threshold in ns-2, the CS range of a node is almost double the transmission range. In such a case, there are few hidden terminals in the ad hoc network in the simulations (no obstacles are simulated in ns-2). If nodes are evenly distributed in the ad hoc network, fifteen nodes are the minimum for the network to be connected. Therefore, all the simulation results for ad hoc networks start from fifteen nodes.

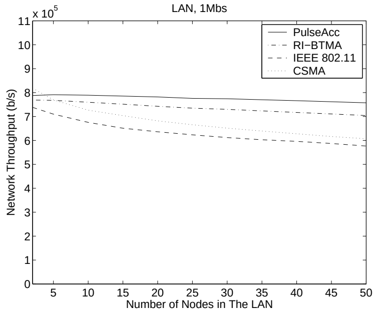

Fig. 9. Network Throughput vs. Number of Nodes

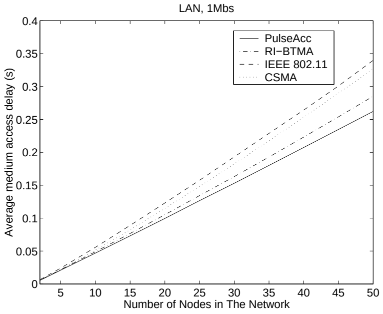

Fig. 10. Medium Access Delay vs. Number of Nodes

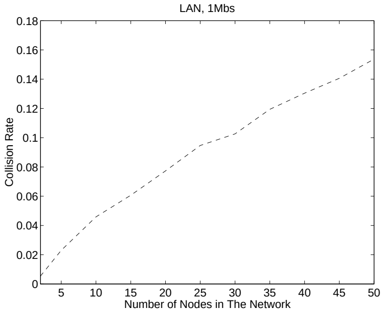

Fig. 11. Collision Rate vs. Number of Nodes

Fig. 12 shows the node throughput versus the number of nodes in the network for the case in which the link rate is 1Mbps. As shown in the figure, IEEE 802.11 still has the lowest performance in this case because its RTS/CTS control overhead still does not pay off due to the low probability of hidden terminals in the network. CSMA, however, has the highest performance in this few-hidden-terminal case. In addition, PulseAcc and RI-BTMA have similar performance in this few-hidden-terminal case. This happens because PulseAcc has a lower data rate than RI-BTMA and its collision detection capability is not fully demonstrated in this case.

However, when the link rate increases to 54Mbps, PulseAcc shows clear gains over the other protocols, as shown in Fig. 13. The main source of performance gains for PulseAcc is collision detection. When a collision is detected, some medium time is saved by not transmitting collided packets. Such saved medium time can be used to transmit more bits when the link rate is higher. This explains why PulseAcc shows higher gains in the 54Mbps link rate case. Note that although a single detected collision in the 54Mbps link rate case saves less time than that in the 1Mbps case, there are more collisions to detect in the 54Mbps case because more frames are transmitted in a given period in the 54Mbps case.

Fig. 14 and Fig. 15 show the delay results for the 1Mbps link rate case and the 54Mbps case, respectively. As shown in these two figures, the delay results conform to the throughput results shown earlier. In addition, packets experience less medium access delays in the 54Mbps network, which is expected.

D. More Hidden Terminals Case

Ad hoc networks have hidden terminals in general. In reality, the main sources of hidden terminals are obstacles such as buildings, hills, and trees. To simulate scenarios with more hidden terminals, we increase the CS power threshold of the nodes in the network to shrink the CS range so that more hidden terminals may occur for a transmitting node. With the default settings of ns-2, the CS power threshold of a node is more than twenty times lower than the receive power threshold. The simulation results shown in this section are for the case in which the CS power threshold is increased to half of the receive power threshold.

Fig. 16 and Fig. 17 show the node throughput versus the number of nodes in the network for the 1Mbps and 54Mbps link rate cases, respectively. As shown in these figures, when there are hidden terminals, PulseAcc shows significant gains over the other three protocols in both 1Mbps and 54Mbps link rate cases. This is because hidden terminals cause collisions and PulseAcc obtains performance gains by terminating collisions. The medium access delay results shown in Fig. 18 and Fig. 19 conform to the throughput results, as in earlier cases.

We now show the collision detection results for PulseAcc. Fig. 20 shows the collision rate versus the number of nodes in the network for the 1Mbps link rate case, and Fig. 21 shows the results for the 54Mbps case. As defined earlier, the collision rate is the number of collisions detected by PulseAcc at a node over the number of transmissions at the node. As shown in these two figures, when there are more hidden terminals in the network, PulseAcc detects more collisions, which is expected. In addition, a frame has a lower probability to encounter a collision in the 54Mbps case than in the 1Mbps case, particularly when there are hidden terminals in the network, as shown by the comparison of Fig. 20 with Fig. 21. This is because a frame has a shorter transmission time (i.e., a shorter exposure time) when the link rate is higher so that the frame has a lower probability of being corrupted by hidden terminals.

E. The Effect of Environmental Noise

In this section we present the results of the effect of environmental noise on the protocol performances. Environmental noise may have different effect on different MAC protocols. In all cases, environment noise may corrupt data frames. In the IEEE 802.11 case, the RTS/CTS control frames may also be corrupted by environmental noise. In the RI-BTMA case, a sender may wrongly interpret a noise as the tone of its receiver and thus start to transmit to an non-ready receiver. In PulseAcc, environmental noise in the control channel may interrupt the transmission of a node.

To investigate the effect of environmental noise, we place a noise source at the center of the ad hoc network

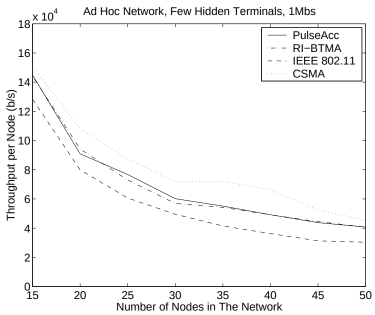

Fig. 12. Node Throughput vs. Number of Nodes

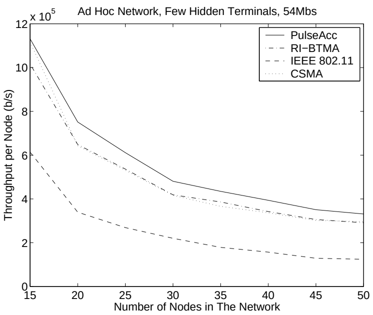

Fig. 13. Node Throughput vs. Number of Nodes

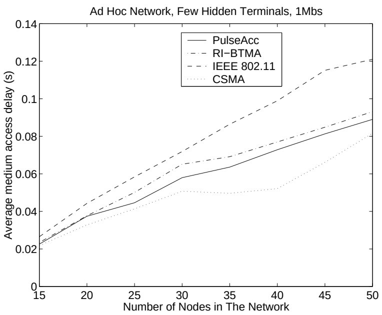

Fig. 14. Medium Access Delay vs. Number of Nodes

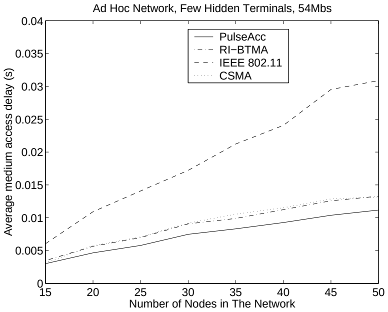

Fig. 15. Medium Access Delay vs. Number of Nodes

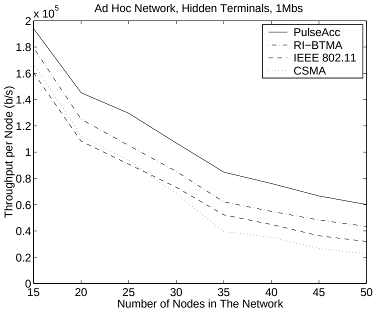

Fig. 16. Node Throughput vs. Number of Nodes

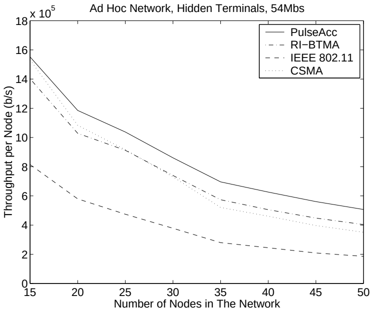

Fig. 17. Node Throughput vs. Number of Nodes

and let the node generate Gaussian noises for both the data and control channels. The average interval of the noise signals is 0.01 second and the mean and standard deviation of the signal lengths are both 0.001 second. With these parameters, there are noise signals in the network for 10% of the time, which reflects a significantly noisy environment.

Fig. 22 shows the node throughput versus the number of nodes in the network for the noise case in which the link rate is 1Mbps and there are hidden terminals. From the comparison of Fig. 22 with Fig. 16, PulseAcc shows higher performance gains over the other protocols when there is noise in the network. One observation explains these results. When a transmitting node senses some noise, the node may need to stop transmitting because the noise may corrupt the frame at the receiver anyway. Therefore, the design of PulseAcc may help in a noisy environment, as shown in Fig. 22. In addition, Fig. 22 shows that RI-BTMA is sensitive to noise, which is expected because the protocol uses tones as the clear-to-send message for a sender and thus noise may trigger a sender to transmit by mistake. For conciseness, delay results are not showed starting from this section.

F. A Comprehensive Scenario

The scenarios used in the preceding sections are for evaluating the proposed protocol with the least interference from other layers, as introduced earlier. This section shows the simulation results for a comprehensive scenario where the ad hoc network has routed traffic and the fifty nodes in the network have random waypoint movement.

In the new scenario, there are up to twenty five constant-bit-rate (CBR) background flows in the ad hoc network, and their rates determine the network load. Meanwhile, a test flow of a constant rate checks the throughput that it can obtain in the network in different cases of network load. The link rate is 1Mbps and there are hidden terminals. In addition, the nodes in the ad hoc network have random waypoint movement. Their minimum speed and maximum speed are 1.0m/s and 5.0m/s, respectively, and their average pause time is 0.5s.

Fig. 23 shows the percentage of the packets in the test flow that successfully reach the flow receiver versus the flow rate of the background flows (i.e., the network load). As shown in the figure, the proposed protocol has higher gains over the other protocols when the network load is higher. When the network load is high, the relative gains can be more than fifty percent.

VIII. Conclusion

This paper presents a MAC protocol with the capability to detect collisions in distributed wireless networks such as mobile ad hoc networks and mesh networks. The basic idea is to use out-of-band contention pulses that have pauses of random lengths to enable two transmitting nodes to detect each other. Pulses are "relayed" by intended data frame receivers and therefore nodes that are hidden terminals to each other may also detect each other if they transmit at the same time. In addition, CTS pulses are used in the protocol to assist collision detection and reduce control frames in the data channel.

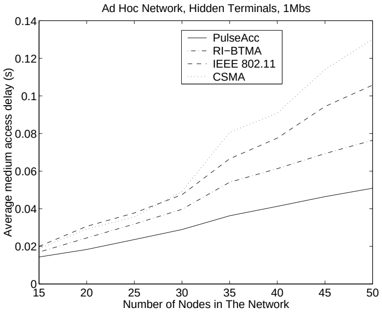

Fig. 18. Medium Access Delay vs. Number of Nodes

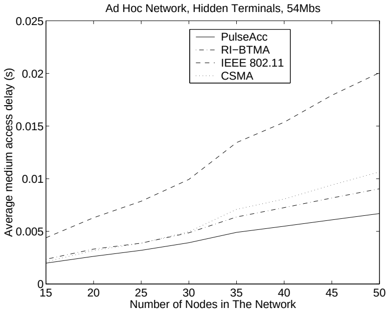

Fig. 19. Medium Access Delay vs. Number of Nodes

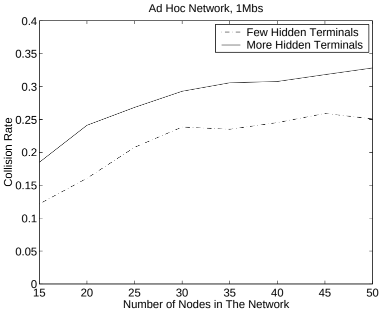

Fig. 20. Collision Rate vs. Number of Nodes

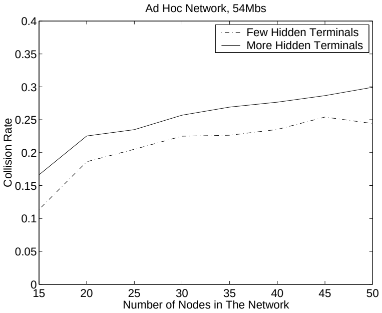

Fig. 21. Collision Rate vs. Number of Nodes

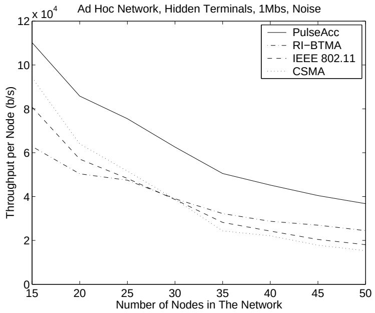

Fig. 22. Node Throughput vs. Number of Nodes

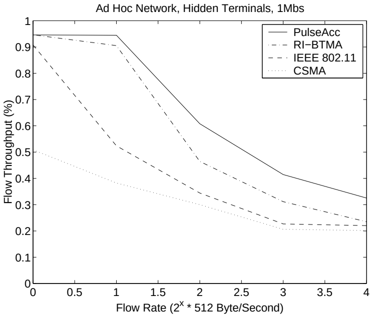

Fig. 23. Flow Throughput vs. Network Load

The comprehensive analysis and simulation results in the paper show that as compared with existing protocols, the proposed MAC protocol achieves outstanding throughput gains in ad hoc networks with hidden terminals.

References

[1] N. Abramson, "The Aloha system - another alternative for computer communications," in *Fall Joint Computer Conference, AFIPS Conference Proceedings*, Montvale, NJ, 1970.

[2] R. M. Metcalf and D. R. Boggs, "Ehthernet: Distributed packet switching for local computer networks," *Communications of the ACM*, vol. 19, pp. 395–404, July 1976.

[3] 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.

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

[5] 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.

[6] 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.

[7] ——, "Solutions to hidden terminal problems in wireless networks," in *Proc. of the ACM SIGCOMM*, French Riviera, France, September 1997.

[8] IEEE 802.11 wireless local area networks. [Online]. Available: http://grouper.ieee.org/groups/802/11/

[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] 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.

[12] 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.

[13] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly, "Distributed multi-hop scheduling and medium access with delay and throughput constraints," in *Proc. of the ACM MOBICOM*, Rome, Italy, July 2001.

[14] M. Barry, A. T. Campbell, and A. Veres, "Distributed control algorithms for service differentiation in wireless packet networks," in *Proc. of the IEEE INFOCOM*, Anchorage, Alaska, April 2001.

[15] 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.

[16] X. Yang and N. H. Vaidya, "Priority scheduling in wireless ad hoc networks," in *Proc. of the ACM MOBIHOC*, EPFL Lausanne, Switzerland, June 2002.

[17] 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.

[18] 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.

[19] R. Rom, "Collision detection in radio channels," *Local Area and Multiple Access Networks*, pp. 235–249, 1986.

[20] P. Jacquet, P. Minet, P. Muhlethaler, and N. Rivierre, "Priority and collision detection with active signaling - the channel access mechanism of HIPERLAN," *Wireless Personal Communications*, vol. 4, pp. 11–25, January 1997.

[21] Y. Tay and K. Chua, "A capacity analysis for the IEEE 802.11 MAC protocol," *Wireless Networks*, vol. 7, no. 2, pp. 159–171, March 2001.

[22] O. Tickoo and B. Sikdar, "On the impact of IEEE 802.11 MAC on traffic characteristics," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 2, pp. 189–203, Februrary 2003.

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

[24] J. Proakis, *Digital Communications, 4th Ed.* New York: McGraw-Hill Science Engineering Math, 2000.

[25] T. Rappaport, *Wireless Communications: Principles and Practice*. New Jersey: Prentice Hall, 1999.

PLACE
PHOTO
HERE

Jun Peng received his Ph.D. degree in electrical engineering from the Electrical, Computer, and Systems Engineering Department of Rensselaer Polytechnic Institute, Troy, NY, USA in 2004. He is currently an assistant professor at the Electrical Engineering Department of University of Texas - Pan American, Edinburg, TX, USA. His general research interests are in computer communication networks and embedded wireless systems. He has published referred papers in conferences and journals in the areas of medium access control, congestion control, error control, and protocol design and analysis. He has recently been working on or exploring topics in medium access control, broadcasting control, routing, and security in the general area of wireless networks. He is a member of IEEE.

PLACE
PHOTO
HERE

Liang Cheng received the BS degree from Huazhong University of Science and Technology, Wuhan, China, and the MS degree from Tsinghua University, Beijing, China. He received the PhD degree in electrical and computer engineering from Rutgers, The State University of New Jersey in May 2002. Since August 2002, he has been with the Department of Computer Science and Engineering at Lehigh University as an assistant professor and directed the Laboratory of Networking Group (LONGLAB). His research interests are wireless networks and mobile computing. He has been the principal investigator (PI) and a co-PI of projects funded by the US National Science Foundation (NSF), the Defense Advanced Research Projects Agency (DARPA), and other sponsors. He has published more than 50 technical papers and holds 1 patent. He is a receipt of Christian R. & Mary F. Lindback Foundation Minority Junior Faculty Award in 2004.

PLACE
PHOTO
HERE

Biplab Sikdar (S'98, M'02) received the B. Tech degree in electronics and communication engineering from North Eastern Hill University, Shillong, India, the M. Tech degree in electrical engineering from Indian Institute of Technology, Kanpur and Ph.D in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, USA in 1996, 1998 and 2001, respectively. He is currently an Associate Professor in the Department of Electrical, Computer and Systems Engineering of Rensselaer Polytechnic Institute, Troy, NY, USA. His research interests include wireless MAC protocols, network routing and multicast protocols, network security and queuing theory.

Dr. Sikdar is a member of IEEE, Eta Kappa Nu and Tau Beta Pi.