1

Dual Busy Tone Multiple Access (DBTMA) – A Multiple Access Control Scheme for Ad Hoc Networks

Zygmunt J. Haas and Jing Deng
School of Electrical and Computer Engineering
Cornell University
Ithaca, NY 14853
e-mail: {*haas, jing*}*@ece.cornell.edu*
URL: *http://www.ece.cornell.edu/~haas/wnl.html*

Abstract—

In Ad Hoc Networks, the hidden and the exposed terminal problems can severely reduce the network capacity on the Multiple Access Control (MAC) layer. To address these problems, the RTS/CTS (Ready-To-Send and Clear-To-Send) dialogue has been proposed. However, MAC schemes using RTS/CTS dialogues on single channel could not completely solve the hidden and the exposed terminal problems, as pure "packet sensing" MAC schemes are not safe even in fully connected networks. We propose a new MAC protocol, termed the *Dual Busy Tone Multiple Access (DBTMA)* scheme. The operation of the DBTMA protocol is based on the RTS packet and two narrow-bandwidth out-of-band busy tones. With the use of the RTS packet and the receive busy tone set up by the receiver, we completely solve the hidden and the exposed terminal problems. The busy tone set up by the transmitter provides protection for the RTS packets, increasing the probability of successful RTS reception and, consequently, increasing the throughput. This paper outlines the operation rules of DBTMA and analyzes the performance of the scheme. Simulation results are provided to support the analytical results. It is concluded that the DBTMA protocol is superior to other schemes that rely on RTS/CTS dialogues on a single channel or those that rely on a single busy tone. As a point of reference, the DBTMA scheme out-performs FAMA-NCS by 20-40% in our simulations using the network topologies described in the FAMA-NCS paper. In an Ad Hoc network with a large coverage area, DBTMA achieves a 140% performance gain over FAMA-NCS and a 20% performance gain over RI-BTMA.

I. Introduction

Ad Hoc Network is a collection of wireless hosts forming a temporary network without the help of any established infrastructure or central control. Network operations, such as routing, are performed in a cooperative manner. The applications of Ad Hoc networks are in situations in which the network needs to be deployed rapidly, such as communications in emergency situations. Recently, the Bluetooth technology, as discussed by Haartsen in [1] and presented in [2], was proposed with which devices can form Ad Hoc networks to communicate with each other. Due to the large span of Ad Hoc networks and limited radio transmission range, multi-hop routing is usually used, in which the communication between any two nodes is performed by forwarding the data packet from one node to another until the packet reaches the destination.

In a single-channel Ad Hoc network, one channel is shared by a number of communicating nodes located in close proximity. The throughput of such a network depends largely upon the performance of the Multiple Access Control (MAC) protocol in use, which controls and coordinates the access of the nodes to the shared channel. In order to increase the throughput, many MAC schemes, such as Carrier Sensing Multiple Access (CSMA) by Kleinrock and Tobagi in [3] and CSMA with Collision Avoidance (CSMA/CA) by Colvin in [4], require nodes to sense the common channel before packet transmission. However, collisions, which arise when more than one packet arrives at a node at the same time, are still possible. Two phenomena have major impacts on the capacity of Ad Hoc networks: the hidden and the exposed terminal problems.

Hidden terminals (e.g., node H in Figure 1) are the nodes in the range of the receiver (node B) but out of the range of the transmitter (node A). Since collisions occur at the receiver, sensing the common channel before an attempt to access the channel will not, in general, eliminate access collisions, which reduce the network capacity for transmission of useful data. This is the so-called hidden terminal problem. While the transmissions of the hidden terminals may destroy a data packet at the receiver, the hidden terminals should be allowed to receive data packets. Of course, proper design is required to allow hidden terminals to announce that they are free to receive.

On the other hand, the *exposed terminal* problem, as discussed by Karn in [5] comes about when nodes are in the range of the transmitter but not the receiver, such as node E in Figure 1. If the regular carrier sensing mechanism is used, the exposed terminals will defer from accessing the shared channel, although in such cases parallel communication can take place to increase the network utilization. The culprit is, again, the fact that the collisions occur at the receiver, while channel sensing schemes test the channel condition at the transmitter.

This work has been sponsored in part by the ONR contract number N00014-00-1-0564 and the AFRL contract number F360602-97-C-0133.

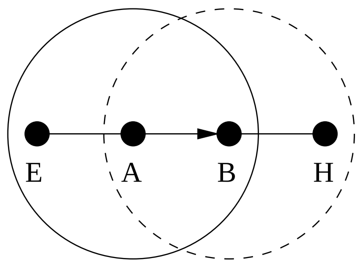

Fig. 1. Hidden/Exposed Terminals

From the above discussion, it is clear that pure carrier sensing mechanism, e.g., the CSMA scheme in the work by Tobagi and Kleinrock [6], does not suffice to achieve high network utilization in Ad Hoc networks. Many other MAC protocols have been proposed, attempting to address the hidden and the exposed terminal problems (e.g., Karn [5], Tobagi and Kleinrock [6], and Bharghavan et al. [7]). Specifically, in works by Karn [5] and Bharghavan et al. [7], the RTS/CTS (Ready-To-Send and Clear-To-Send) dialogue is used. However, these RTS/CTS type MAC protocols do not solve either the hidden or exposed terminal problems. The reason is that, although exposed terminals are permitted to send their RTS packets to request the channel, they will not receive any CTS replies while another node is sending data on the single channel. Also, the hidden terminals still cannot receive, as they are forbidden to transmit any packet (including CTS replies) on the channel. With these packet sensing protocols, all packets are at risk in multi-hop networks (They are not even safe in a fully connected topology).

In this paper, we propose the Dual Busy Tone Multiple Access (DBTMA) protocol. In DBTMA, we use the RTS packets to initiate channel request. Two out-of-band busy tones are used to protect the RTS packets and the data packets, respectively. One of the busy tones, the transmit busy tone, BT_t, set up by the RTS sender, is used to protect the RTS packets. Another busy tone, the receiver busy tone, BT_r, set up by the receiver, acknowledges the RTS packet and provides continuous protection to the incoming data packets. Nodes sensing any busy tone defer from sending their RTS packets on the channel. With the use of the RTS packet and the BT_r signal, the exposed terminals are able to initiate data packet transmissions and the hidden terminals can reply to RTS requests and initiate data packet reception, while data packet transmission is taking place between the transmitter and the receiver.

In this paper, we present the operational rules of the DBTMA protocol in section III and we analyze the performance of DBTMA in section IV. We provide simulation results in section V, illustrating the performance of DBTMA, supporting the analytical results, and comparing it with related schemes. The conclusion from our study, which we present in the last section, is that the DBTMA protocol is superior to other schemes that rely on RTS/CTS dialogues on a single channel or those that rely on a single busy tone. But, first, we discuss related works in the next section.

II. Related Works

In [6], Tobagi and Kleinrock introduced a scheme that uses a busy tone to address the hidden terminal problem. The protocol, named Busy Tone Multiple Access (BTMA), relies on a centralized network operation, i.e., a network with base stations. When a base station senses the transmission of a terminal, it broadcasts a busy tone signal to all terminals, keeping them (except the current transmitter) from accessing the channel. The original BTMA was proposed to be used in a network with a base station and that uses the busy tone in a centralized manner. Although the protocol can be used in Ad Hoc networks with distributed control, its performance is unclear to us.

Tobagi and Kleinrock proposed and studied the Split-channel Reservation Multiple Access (SRMA) scheme for a network with a number of terminals and one central station in [8]. The whole channel is split into two sub-channels for message transmission and control packet transmission (RAM mode), or three sub-channels for message transmission, request transmission, and answer-to-request transmission (RA mode). A ready node sends its request to the central station on the request channel in an ALOHA or CSMA manner. Successful requests will be acknowledged by the central station before the data packet is transmitted.

In the Receiver-Initiated Busy-Tone Multiple Access scheme (RI-BTMA) proposed by Wu and Li [9], a packet preamble is sent to the intended receiver by the transmitter. Once the preamble is received correctly, the receiver sets up an out-of-band busy tone and waits for the data packet. The transmitter, on sensing the busy tone, sends the data packet to the destination. The busy tone serves two functions: to acknowledge the channel access request and to prevent transmissions from other nodes. RI-BTMA was proposed to be used in slotted manner. The correct operation of RI-BTMA depends largely on the synchronization of slots, which is usually difficult to achieve globally in Ad Hoc networks, especially of the mobile type.

In Multiple Access Collision Avoidance (MACA) [5], Karn originally proposed the use of short control packets, the Request-To-Send (RTS) and the Clear-To-Send (CTS) packets, for collision avoidance on the shared channel. A ready node transmits an RTS packet to request the channel. The receiver replies with a CTS packet. The reception of the CTS packet acknowledges that the RTS/CTS dialogue has been successful and starts the transmission of the actual data packet. All other nodes that hear the RTS packet back off for a time long enough for the transmitter to receive the CTS packet. All other nodes that hear the CTS packet back off for a time long enough for the receiver to receive the data packet. However, MACA becomes ALOHA with the existence of hidden terminals. MACA was proposed to solve the hidden/exposed terminal problems, but it does not solve either the hidden or exposed terminal problems.

Bharghavan [7] suggested the use of the RTS-CTS-DS-

DATA-ACK message exchange for a data packet transmission in the MACAW protocol. The DS (Data Sending) packet was added to notify all nodes in the transmitter's range of its following use of the shared channel. The ACK packet was implemented for immediate acknowledgment and the possibility of fast retransmission of collided data packets. A new back-off algorithm, the Multiple Increase and Linear Decrease (MILD) algorithm, was also proposed in the paper to address some of the unfairness problems in accessing the shared channel. Additional features of the MILD algorithm, such as back-off interval copying and multiple back-off intervals for different destinations, further improve the performance of MACAW. However, similar to MACA, MACAW does not solve either the hidden or exposed terminal problems.

In [10], Fullmer and Garcia-Luna-Aceves proposed the Floor Acquisition Multiple Access (FAMA) scheme. In FAMA, each ready node has to acquire the channel (the "floor") before it can use the channel to transmit its data packets. FAMA uses both carrier sensing and RTS/CTS dialogue to ensure the acquisition of the "floor" and the successful transmission of the data packets. In [11], FAMA-NPS (FAMA Non-persistent Packet Sensing) was studied and it was shown that "packet sensing" schemes, such as FAMA-NPS, MACA, and MACAW, could not solve the hidden/exposed terminal problems.

FAMA was further extended to FAMA-NCS (FAMA Non-persistent Carrier Sensing). FAMA-NCS, with the use of the carrier sensing scheme and longer CTS packets, provides a "CTS dominance" mechanism to ensure correct floor acquisition and collision-free data packet reception. Once a node has begun the transmission of a CTS packet, any other node within its range that transmits an RTS packet simultaneously will hear at least a portion of the dominating CTS packet after returning from transmit mode and backoff. In FAMA-NCS, no CTS packet will ever collide with a data packet. However, the "CTS dominance" mechanism may have adverse effect when RTS packet collisions take place. When nodes sense the carrier of collided packets, they mistakenly treat these collided RTS packets to be "CTS dominance", which inhibits them from sending any packet for a time long enough to receive a data packet. The channel capacity is wasted. This false "CTS dominance" effect is more severe when FAMA-NCS operates in Ad Hoc networks with hidden terminals, where RTS collisions happen more frequently under heavy traffic even with the use of carrier sensing. Finally, FAMA-NCS does not solve the exposed terminal problems, although it addresses the hidden terminal problems successfully.

In the IEEE 802.11 MAC layer protocol [12], an access method called Distributed Coordination Function (DCF), which implements the CSMA/CA protocol proposed in the work by Colvin [4], is used. It is an extension to the basic RTS/CTS dialogue. After sensing the channel free, an RTS packet will be sent and the CTS packet indicating the readiness to receive the data at the receiver will be transmitted back to the source. This scheme is similar to the MACA protocol, with the addition of the CSMA mechanism. While the CSMA scheme lowers the probability of RTS packet collisions, IEEE 802.11 MAC layer protocol does not solve either the hidden or exposed terminal problems.

In [13], Gummalla and Limb proposed a Wireless Collision Detection (WCD) scheme based on their transceiver architecture design, in which a feedback channel is implanted in the main data channel. The WCD scheme was proposed to be used in high speed distributed wireless LAN, in which the turn-around time of the half-duplex radio becomes significant compared with packet transmissions time. Every neighbor node sensing the start of the data packet transmission sets up the feedback signal before the end of the Receiver Detection Interval (RDI). The feedback signal inhibits any transmission from all neighbors during RDI (This effectively inhibits all 2-hop neighbors of the transmitter to transmit during this period of time). After RDI, the intended destination decodes the header of the data packet, matching the destination ID on the header and local ID, and leaves the feedback signal on, while all other neighbors set off the feedback signal. The feedback signal after RDI works as the confirmation of the transmitted data packet and notification to neighbor nodes. By sensing the feedback signal after RDI, the transmitter keeps transmitting the packet. If no feedback signal is sensed after RDI, the transmitter stops the transmission.

Except for the use of the feedback signal generated from the neighbors of the RTS sender in RDI, the WCD scheme is very similar to the RI-BTMA scheme in operational rules. Operating in slotted manner, the WCD scheme requires network-wide time synchronization, which could be harder to achieve in ad hoc networks compared with wireless LAN.

Protection of the data packets at the receiver has to be guaranteed to achieve good performance for a MAC protocol in Ad Hoc networks. The RTS/CTS dialogue was introduced to prevent all other nodes in the receiver's range from using the channel. However, the use of this dialogue on single channel can not solve all the the hidden and the exposed terminal problems, although FAMA-NCS does solve the hidden terminal problem with the help of carrier sensing mechanism. The use of in-band CTS packet effectively inhibits the data transmission of the exposed terminals and the data reception of the hidden terminals. Furthermore, as the possibility of CTS packet collisions at neighbor nodes remains, collisions of data packets are inevitable, unless an additional mechanism is provided to protect data packets. In particular, since a CTS packet may not be received correctly at some neighbors, these nodes might send their RTS requests on the channel during the time the data packet is being received, leading to the destruction of the data packet and reduction of the network utilization. To address these problems, we've introduced here the Dual Busy Tone Multiple Access (DBTMA) scheme, whose operation rules are given in the following section.

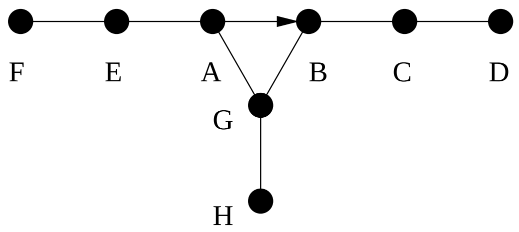

Fig. 2. An Example Network

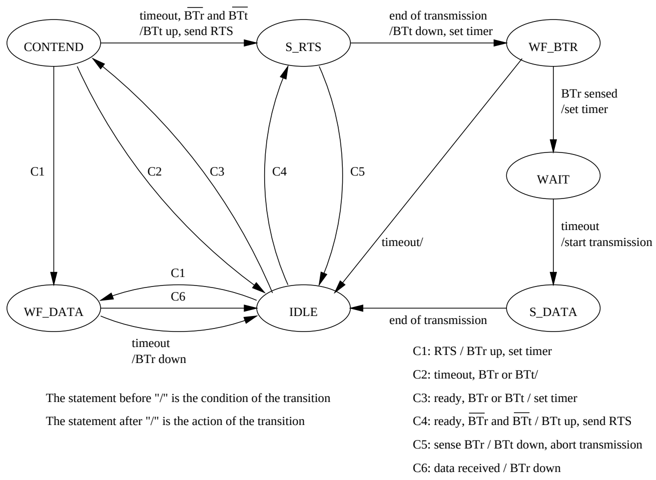

Fig. 3. The Finite State Machine of DBTMA

III. The DBTMA Protocol

In the DBTMA protocol, two narrow-bandwidth tones are implemented with enough spectral separation on the single shared channel. BT_t (the transmit busy tone) and BT_r (the receive busy tone), indicate whether the node is transmitting RTS packets or receiving data packets, respectively. The transmit busy tone (BT_t) provides protection for the RTS packets to increase the probability of successful RTS reception at the intended receiver. We use the receive busy tone (BT_r) to acknowledge the RTS packet and provide continuous protection to the transmitted data packets. All nodes sensing any busy tone are not allowed to send RTS requests. When the start of the BT_r signal is sensed, a node sending RTS packet is required to abort such transmission immediately. Indeed, the RTS packets and the receive busy tone solve the hidden and the exposed terminal problems.

The operation of the DBTMA protocol will be explained by the way of an Ad Hoc network example, shown in Figure 2. In this figure, a solid line between any two nodes indicates that they can hear each other. Hence, node C is a hidden terminal to the transmission from node A to node B, and node E is an exposed terminal if it wants to communicate with node F, for example (but not with node A).

A node implementing the DBTMA protocol can be in one of the following seven states: IDLE, CONTEND, S_RTS, S_DATA, WF_BTR, WF_DATA, and WAIT. Figure 3 depicts the Finite State Machine (FSM) of the DBTMA scheme. A node with no packets to send stays in the IDLE state. When a node has a packet to send but it is not allowed to send the RTS packet, it stays in the CONTEND state. Nodes sending RTS or DATA packets are in the S_RTS or S_DATA states, respectively. The RTS packet sender waits for the acknowledgment from its intended receiver in the WF_BTR state. The receiver waits for the data packet in the WF_DATA state.

When node A has a data packet to send while it is in the IDLE state, it tries to sense the BT_r and the BT_t busy tone signals. If none of the busy signals is present (which means that no one in node A's transmission area is receiving data packet or sending RTS packets), it turns on its BT_t signal, sends an RTS packet to node B, and goes into the S_RTS state. Otherwise, it sets a random timer and goes into the CONTEND state. By the end of the RTS transmission, node A turns off its BT_t signal, sets a timer, and goes into the WF_BTR state. When node B receives the RTS packet, it turns on its BT_r signal, replying to node A and announcing that it is waiting for incoming data packet. Then it sets up a timer and goes into the WF_DATA state.

Node A continuously monitors the BT_r signal when it is in the WF_BTR state. When a BT_r signal is sensed, it knows that its channel request has been successful. Before node A sends the data packet, it waits a mandatory waiting time ($t_{mw} = 2\tau$)[1] in the WAIT state. This mandatory waiting time is meant to allow all possible RTS transmissions in the range of the receiver to be aborted. Upon timeout in the WAIT state, node A goes into the S_DATA state and sends the data packet. By the end of its transmission, node A goes into the IDLE state. Upon successful reception of the data packet, node B turns off the BT_r signal and goes into the IDLE state, ending the communication. If, by any reason, node B does not receive the data packet before the timer expires, it turns off the BT_r signal and goes into the IDLE state.

Upon timeout in the CONTEND state, node A turns on its BT_t signal and sends its RTS packet if no busy tone signal is sensed. Otherwise, it goes back into the IDLE state. From the perspective of the other nodes in the neighborhood, their operations can be described as following: When the BT_t and/or the BT_r signal is sensed, a node (e.g., node E, G, or C) is not allowed to send any RTS request. When the start of a BT_r signal is sensed while a node (e.g., node G or C) is in the S_RTS state, it aborts its RTS transmission, turns off its BT_t signal, and goes back to the IDLE state.

We show the time diagram with operations of node A and node B in Figure 4. Additional details of the DBTMA operation rules are presented in the Appendix I.

IV. Performance Analysis

In order to study the performance of the DBTMA protocol, we adopt the method developed by Tobagi and Klein-

[1] τ is the maximum propagation delay between the transmitter and the receiver.

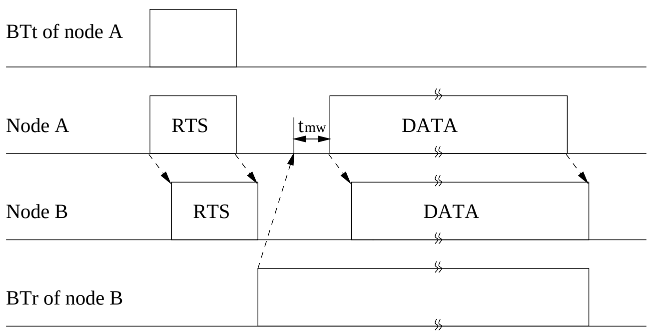

Fig. 4. Time Diagram of DBTMA

rock in their study of CSMA and BTMA [6] and further used by Fullmer and Garcia-Luna-Aceves in FAMA [11]. The network model consists of a large number of terminals communicating with each other over a single channel. All nodes are within the range of each other. We have the following assumptions for the DBTMA protocol and the analysis:

- The radio transmission range of the Ad Hoc networks in which the DBTMA scheme operates is about tens or hundred of meters. There is no capture effect or fading on the channel.
- Any overlap by transmissions at a receiver causes the receiver to not understand either packet. Packet collisions are the only source of packet error.
- The data processing time and the transmit/receive turnaround time at each node are negligible.
- We assume that the busy tone signal and the data signal have the same transmission range.
- The inter-channel interference between busy tone signals and data signal is negligible.
- The bandwidth consumption of the busy tones is negligible compared to the bandwidth of the data channel. [2]
- The busy tone detection delay is t_d, which depends on the communication hardware and might not, in general, be negligible.
- The mandatory waiting time is set to $t_{wm} = 2\tau$.
- The transmission time of the RTS packet (γ) is larger than $t_d + 4\tau$.
- The network has a large number of nodes, which collectively generate a Poisson traffic with mean aggregate rate of λ channel requests per second.
- The data packet transmission time, the RTS packet transmission time, and the maximum one way propagation delay are δ, γ, and τ, respectively.

We further assume that the radio signal propagation delay between any two nodes is τ, hence the channel capacity we obtain is a lower bound.

[2]As discussed by Tobagi and Kleinrock in [6], the bandwidth consumption of a busy tone signal could be in the range of [0.1, 10] Khz with the main data channel as 100 Khz. Although we can't find any data sheet on busy tone hardware implementations, we expect that each of the busy tones can be implemented with the bandwidth of 10 Khz.

We treat the transmission cycle on the channel as a renewal process. We define a busy period, in which there is a transmission on the shared channel, as the time between two consecutive idle periods. A busy period might be a period with successful data transmission, or a period with packet collisions. The channel throughput can be expressed as, as discussed by Kleinrock and Tobagi in [3]:

$$S = \frac{\overline{U}}{\overline{B} + \overline{I}} \tag{1}$$

where $\overline{U}$, $\overline{B}$, and $\overline{I}$ are the average utilization time for data packet transmission, the average busy time, and the average idle time of the channel, respectively, in each cycle.

An RTS packet originated from any node (e.g., node A) is successful if no other RTS packets are sent in the first $t_d + \tau$ seconds. Because this is the sum of the busy tone detection delay and the maximum propagation delay, the BT_t signal set up by node A will be sensed by all nodes after $t_d + \tau$ seconds. So the probability of success of the RTS packet from node A is the probability that there is no arrival during this period of time:

$$P_s = e^{-\lambda(t_d+\tau)}. \tag{2}$$

When the RTS packet is successfully received at the intended receiver (e.g., node B), it will set up its BT_r signal and wait for the data packet. We argue that when the RTS packet is successfully received and the BT_r signal is set up, data packet reception will be guaranteed. An intuitive explanation is the following: All nodes sensing the BT_r signal will abort their RTS transmissions and keep silent. There must not be any other node sending data packets in the range of node B. Otherwise, node B would not have received the RTS packet successfully. Appendix II presents our theorem and proof on this issue.

A successful transmission period (T_s) consists of the transmission time of an RTS packet plus the propagation delay, the busy tone detection delay plus the propagation delay, the mandatory waiting time ($t_{mw} = 2\tau$), the transmission time of the data packet plus the propagation delay, and the period of time for the BT_r signal to be cleared on the channel (τ). So T_s is:

$$T_s = \gamma + \tau + t_d + \tau + t_{mw} + \delta + \tau + \tau = \delta + \gamma + t_d + 6\tau.$$

A failed busy period (T_f) consists of more than one RTS packet. Since no new RTS packets will be sent $t_d + \tau$ seconds after the start of node A's RTS packet, the longest failed busy period is $\gamma + t_d + \tau$. The shortest failed busy period is the situation when more than one RTS packets are sent at approximately the same time, with the failed busy period as $\gamma + \tau$. We assume that the colliding RTS packet arrives uniformly in the duration of $[0, t_d + \tau]$, so the average failed busy period is the average of the longest and the shortest value:

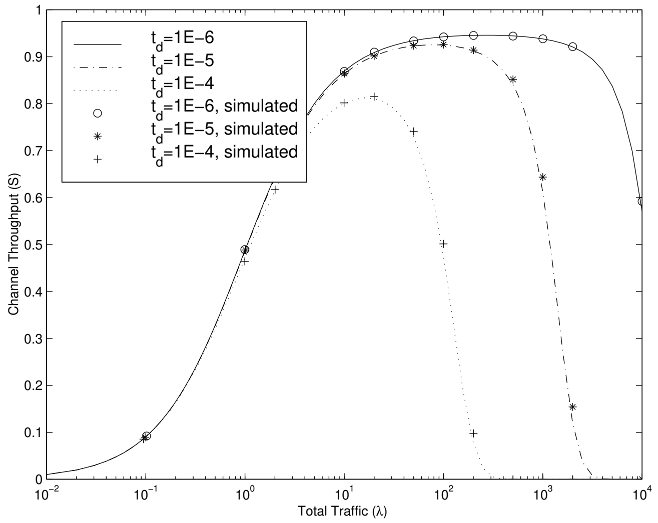

Fig. 5. Channel Throughput of DBTMA with Different t_d

$$T_f = \frac{\gamma + t_d + \tau + \gamma + \tau}{2} = \gamma + \tau + \frac{t_d}{2}. \quad (3)$$

The average busy period is, therefore,

$$\overline{B} = P_s T_s + (1 - P_s) T_f. \quad (4)$$

The average utilization time is the product of the successful probability of busy periods and the data packet transmission time:

$$\overline{U} = P_s \delta. \quad (5)$$

The average idle period is the average inter-arrival time of RTS's from all nodes. Since the RTS packets arrive according to the Poisson distribution, we have:

$$\overline{I} = \frac{1}{\lambda}. \quad (6)$$

Substituting Eq. (4), (5), and (6) into Eq. (1), we obtain the channel throughput of the DBTMA protocol in the discussed network model:

$$S = \frac{P_s \delta}{P_s(\delta + \gamma + t_d + 6\tau) + (1 - P_s) T_f + 1/\lambda}, \quad (7)$$

where P_s and T_f are given by Eq. (2) and (3), respectively.

In Figure 5, we draw the channel throughput of DBTMA with different busy tone detection delay (t_d). In the figure, we considered a wireless network with channel data rate of 1 Mbps. The data packet length is 4096 bits and the RTS packet length is 200 bits. There are 20 nodes in the 50×50 m^2 network. The radio transmission range is 35 meters,

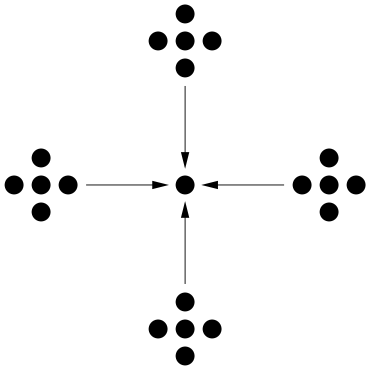

Fig. 6. Exampled Network of N Independent Groups (N=4)

which is the maximal distance between any two nodes [3], with a maximum one-way propagation delay of 0.12 μsec. The considered busy tone detection delays (t_d) are 10^{-6}, 10^{-5}, and 10^{-4} second. Each simulation represents 100 seconds "real time". The lines show our analytical results and the symbols represent the simulation results. Good match between analytical results and simulation results is achieved. The small discrepancy can be explained by the finite number of nodes in the simulated network and the infinite number of nodes in the analytical model.

It can be observed that the channel throughput for small t_d is always above 0.9. When t_d is 10^{-6} second, the channel throughput of DBTMA is 0.94. It decreases to 0.92 when t_d is changed to 10^{-5} second, because the longer vulnerable period of each RTS packet leads to lower probability of successful RTS requests and larger overhead. When t_d is 10^{-4} second, which is half of the RTS packet transmission time (γ), the performance of DBTMA degrades to 0.82. We can also notice the earlier decrease of channel throughput as a function of the traffic load for larger t_d, because of the longer vulnerable period of the RTS packets.

The analytical results for a non-fully connected network are more difficult to obtain. Hence we resort to simulations.

V. Performance Evaluation

In order to evaluate the performance of DBTMA, we have simulated Ad Hoc networks implementing the DBTMA protocol and other related protocols in the OPNET simulator and our own C/C++ simulator. Each of our simulation results represents the average of 10 random runs. When the channel data rate is 1 Mbps, each simulation represents a "real time" of 100 seconds. The "real time" is 400 seconds when the channel data rate is 256 Kbps.

Firstly, we studied the performance of the DBTMA scheme under different hidden terminal situations. We simulated the DBTMA protocol in an Ad Hoc network with N ($N = 1, 2, \cdots 6$) independent groups and one common receiver. Each groups contains 5 nodes, which are in the

[3]We assumed the simulated network to be a closed coverage area, which effectively creates a torus. So the four corners are treated as one point in distance calculation.

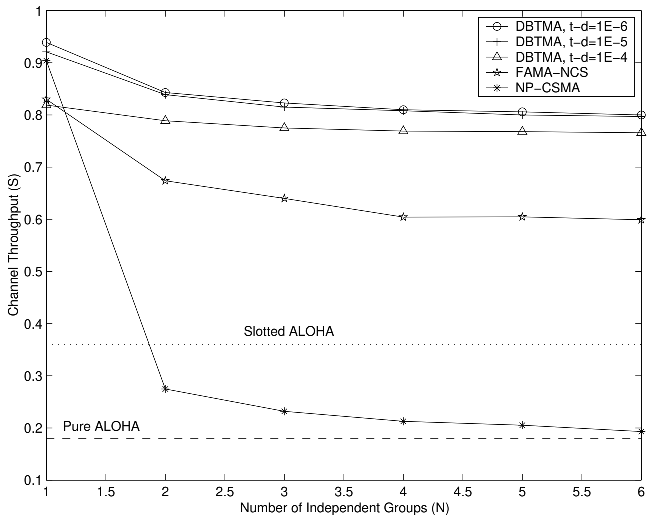

Fig. 7. Channel Throughput of DBTMA with Different N

range of each other. All these nodes in the N groups generate data traffic to send to the common receiver (central station), which sits at the center of the network. Figure 6 shows an example of such network with $N = 4$. We borrowed this example of network from the work by Fullmer and Garcia-Luna-Aceves [11]. The length of the RTS packet is 200 bits, the length of the data packet is 4096 bits, and the channel data rate is 1 Mbps. The radio transmission range is about 2 km, with propagation delay as 6.7 μsec. [4] Figure 7 compares the DBTMA protocol with the FAMA-NCS protocol [5] and other related MAC protocols in the same environment.

From Figure 7, we find that the DBTMA scheme has higher channel throughput than any other MAC schemes we show on the graph. When $N = 1$, DBTMA achieves network utilization of 0.94 for small t_d and 0.82 when t_d is 10^{-4} second. The Non-Persistent CSMA (NP-CSMA) scheme has a throughput of 0.90, while the FAMA-NCS scheme has 0.83. However, the performance of FAMA-NCS scheme degrades to 0.6 when N increases to 6, because of higher probability of RTS packet collisions and the unnecessary idle time of the channel after RTS collisions (false "CTS dominance"). When N is 6, the DBTMA protocol has a throughput of 0.8 or 0.77, depending on the value of t_d. The NP-CSMA scheme degrades quickly as N increases. Eventually it performs the same as the pure ALOHA result, which was reported by Kleinrock and Tobagi [6]. For comparison purpose, we also draw the performance of pure ALOHA and slotted ALOHA. Hence, DBTMA increases over FAMA and CSMA with diminishing returns. With hidden terminal, as N increases, CSMA asymptotically approaches ALOHA (18%), and FAMA-NCS approaches MACA (60% – for fully connected network).

[4]We expect the DBTMA scheme to operate in most Ad Hoc networks with radio transmission range smaller than 1 km. So, these results are meant for comparisons only.

[5]In our FAMA-NCS implementation, $\gamma = 200$ μsec and $\epsilon = 0$.

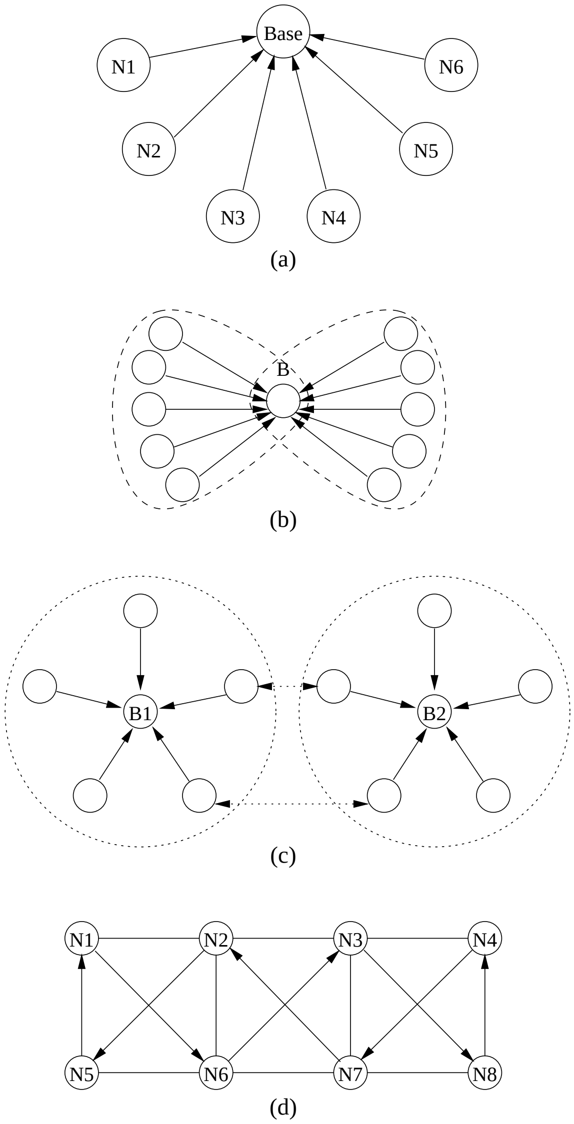

Fig. 8. Simulated Topologies

Hence, for practical values of t_d, the DBTMA scheme out-performs the FAMA-NCS scheme and the NP-CSMA scheme in these network topologies. As the portion of hidden terminals increases (N increases), the performance gain of the DBTMA scheme over the other two increases as well. We assume $t_d = 10^{-6}$ second for the rest of the discussions in this section.

Secondly, we compared the DBTMA protocol and similar protocols for some specific network topologies. To allow meaningful comparison with the FAMA-NCS protocol, we've evaluated the performance of DBTMA in the same network configurations as used in the work by Fullmer and Garcia-Luna-Aceves [11] and depicted in Figure 8. Figure 9 shows the simulation results of DBTMA, FAMA-NCS, and MACAW [6]. The channel data rate is 256 $Kbps$ and nodes are 6 km from each other, with maximum propagation delay of 20 μsec.

[6]We didn't implement MACAW in our simulator, but borrowed the results from the work by Fullmer and Garcia-Luna-Aceves in [11].

Configuration	DBTMA	FAMA-NCS	MACAW
(a)	.94	.78	.63
(b)	.84	.59	.49
(c) B1	.94	.75	.45
(c) B2	.94	.75	.39
(d) average	.69	.49	.06
(d) N1, 4, 5, 8	.90	.57	.07
(d) N2, 3, 6, 7	.48	.42	.05

Fig. 9. Channel Throughput Comparisons

In Figure 8, a solid line with an arrow represents the direction of the data traffic generated by the source node. A solid line without arrow represents that the two nodes are in the range of each other. Dotted lines with arrows show that the two nodes can overhear each other even though they are not in the same communication group.

In configuration (a) of Figure 8 all nodes can hear each other and all traffic is directed to the base node. Configuration (b) has two independent groups which share the same receiver. Configuration (c) has two relatively independent communication groups, with two pairs of nodes being able to overhear each other. In configuration (d), eight nodes form a simple multi-hop network.

As reported in Figure 9, the DBTMA scheme outperforms the FAMA-NCS and the MACAW scheme in these networks. The DBTMA scheme achieves a channel throughput of 0.94 in configuration (a), which is 20% higher than that of the FAMA-NCS scheme. In configuration (b), the throughput of the DBTMA scheme is 0.84, which is 40% higher than that of FAMA-NCS. The DBTMA scheme has approximately 20% performance gain over the FAMA-NCS scheme in configuration (c). In configuration (d), the DBTMA scheme achieves higher average channel throughput than the FAMA-NCS scheme does, with a 40% increase.

The explanation is, despite the fact that both schemes provide correct protection to data packet receptions, the DBTMA scheme completely solves the hidden and the exposed terminal problems, while the FAMA-NCS scheme does not address the exposed terminal problems. For example, concurrent transmissions such as $N1$ to $N6$ and $N4$ to $N7$ (or, $N2$ to $N5$ and $N3$ to $N8$) in configuration (d) are possible in the DBTMA scheme, but they are not allowed in the FAMA-NCS scheme. The FAMA-NCS scheme also mistakenly treats collided RTS packets as "CTS dominance" and the channel is wasted while being idle. With the presence of hidden terminals in configuration (b) and (d), the probability of RTS packet collisions is higher, leading to more severe false "CTS dominance" problem.

We have also simulated and studied the DBTMA protocol in other network operational conditions. In Figure 10, we show the effect of the ratio of the RTS packet length and the data packet length (L_r/L_d) in a fully connected network, in which every node chooses its destination randomly for each generated data packet. The length of the data packet is 4096 bits and the channel data rate is 1 Mbps. There are 20 nodes randomly distributed in a 50×50 m^2 area. The radio transmission range is 35 meters, with a maximum propagation delay of 0.12 μsec.

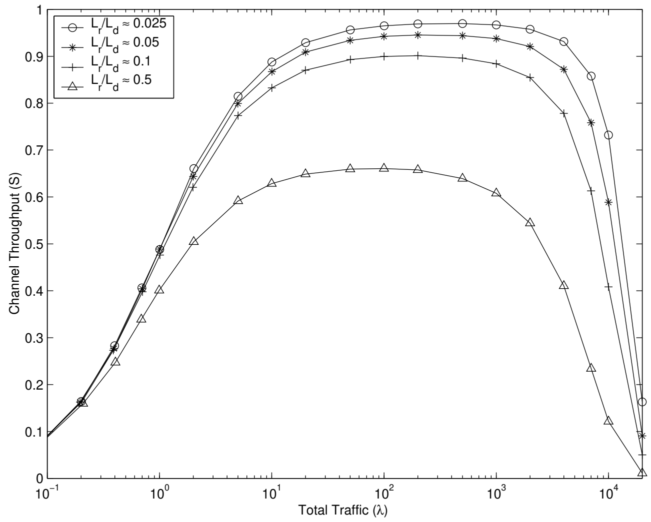

Fig. 10. Performance of Different Length of Control Packet

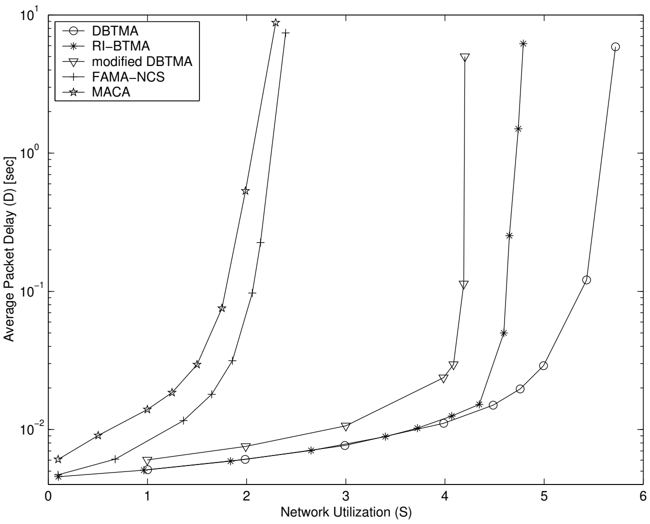

Fig. 11. Network Utilization of DBTMA in Multi-hop Networks

As expected, channel throughput decreases with the increase of L_r/L_d. The channel throughput of the DBTMA scheme is 0.96 when L_r/L_d is 0.025 ($L_r = 100$ bits). This value decreases to 0.94 as L_r/L_d increases to 0.05 ($L_r = 200$ bits). When L_r/L_d is 0.5 ($L_r = 2000$ bits), the throughput is 0.66. The explanation is that the transmission time of the RTS packet contributes to the duration of failed busy periods and overhead of the successful busy periods. When L_r is larger, the overhead will be larger and the throughput will be lower.

Finally, in Figure 11, we compare mean packet delay performance of DBTMA, RI-BTMA, FAMA-NCS, and MACA in an Ad Hoc network with coverage area of 400×400 m^2 and radio transmission range of 100 meters, with maximum propagation delay as 0.33 μsec. 50 nodes are randomly distributed in the network. The RTS packet length is 200 bits, the data packet length is 4096 bits, and the channel data rate is 1 Mbps. In order to compare the packet delay performance of these protocols, we implemented a simple Binary Exponential Back-off (BEB) scheme to allow the blocked and collided data packets to be retransmitted. We also assumed instant acknowledgment of data packet reception for MACA, since other schemes guarantee collision-free data packet receptions. The packet arrival on each node is Poisson distributed and each node randomly selects a neighbor as the destination of each packet. The *modified DBTMA* scheme is the DBTMA scheme without the use of BT_t signal. We defer the discussion of this scheme to the end of the section.

From the graph, it can be observed that the MACA protocol with basic RTS/CTS dialogue and back-off scheme can offer network capacity of 2.2 in the simulated network within reasonable latency. The FAMA-NCS scheme (with back-off) is able to carry maximal throughput of 2.4. The RI-BTMA scheme performs better than both of these schemes, with network capacity of 4.8. The maximal network utilization of the DBTMA scheme is about 5.7, which is 20% higher than that of RI-BTMA and 140% higher than that of FAMA-NCS. Note that these schemes were able to achieve network utilization higher than 1 because of the concurrent transmissions within the network's coverage area.

The explanation of the low performance of MACA is that it does not solve either the hidden terminal problems or the exposed terminal problems. The FAMA-NCS scheme has a similar low performance, because it does not solve the exposed terminal problems. Data packet transmission from the exposed terminals are effectively forbidden on the single channel. The hidden terminals cannot initiate data packet reception, either. FAMA-NCS performs close to MACA in Ad Hoc networks, although it implements carrier sensing and "CTS dominance" mechanisms to support collision-free data packet transmissions. The problem, again, comes from the false "CTS dominance". Note that MACA has almost the same performance as FAMA-NCS does, because we have assumed instant acknowledgment for the MACA scheme in these simulations. We expect the performance of MACA in a real network worse than what is shown, as upper layer retransmissions may take place more frequently.

Both of the DBTMA and the RI-BTMA schemes solve the hidden and the exposed terminal problems. RI-BTMA uses slotted operation, requiring time synchronization. DBTMA provides extra protection for the RTS packets, increasing the probability of successful RTS reception at the intended receiver and throughput. For comparison purpose, we also simulated a modified DBTMA scheme, in which no BT_t signal is used. So the modified DBTMA scheme is an un-slotted version of the RI-BTMA scheme. It has a utilization of 4.2. So RI-BTMA, with the help of the slotted operation, increases the performance by 15% over the modified DBTMA scheme. The DBTMA scheme, with the help of the extra busy tone (BT_t), gains 35% performance over the modified DBTMA scheme, demonstrating the effectiveness of the second busy tone.

VI. Conclusions

In communication networks with a shared channel, MAC protocols synchronize access of multiple nodes to the channel. Due to the random access from nodes, packet collisions are difficult to eliminate totally. Communication networks with hidden terminals pose additional challenges to MAC protocols, because of the lack of the knowledge of on-going communications at these terminals when traditional carrier sensing is used. In order to protect transmission of the data packets, continuous notification of channel state may be used to announce the channel status to all nodes in the range of the node in question.

As the carrier sensing schemes evaluate the state of the channel only at the transmitter, rather than at the receiver, some researchers have proposed to rely on a reservation dialogue (the RTS/CTS dialogue) among the communication nodes. Some of these schemes, e.g., MACA and MACAW, do not solve either the hidden or exposed terminal problems. FAMA-NCS, with the help of the carrier sensing mechanism, addressed the hidden terminal problems successful and left exposed terminal problems unsolved. The use of in-band CTS packet effectively inhibits the data transmission of the exposed terminals and the data reception of the hidden terminals. Furthermore, as the possibility of CTS packet collisions at neighbor nodes remains, collisions of data packets are inevitable unless additional mechanisms are provided to protect them (such as the ones used in FAMA-NCS).

In the proposed-here DBTMA scheme, in addition to the use of the RTS request, two out-of-band busy tones are used. One busy tone, generated at the receiver, serves two functions: (1) notifying the RTS sender that the channel has been successfully acquired, and (2) announcing to its neighbor nodes that it is receiving data packet and that they should not send any packet. The other busy tone, generated at the transmitter while it is sending the RTS packet, gives protection to the RTS packet. With this design, exposed terminals are able to initiate new transmission because they do not need to listen on the shared channel to receive the acknowledgment from their intended receivers. Instead, the acknowledgment of the successful channel request will be sent by means of the receive busy tone. Furthermore, the hidden terminals can reply to RTS requests by simply setting up its receive busy tone. When RTS/CTS dialogues are used on the single channel, such as in the MACA, MACAW, and FAMA-NCS schemes, the hidden terminals cannot send their replies.

Of course, extra hardware is required by the DBTMA scheme. Two busy tone transmitters and sensing circuits need to be incorporated into each communication node. In our study, we did not consider the bandwidth consumption of the busy tones, which might not be negligible. However, we have shown that, with the help of these busy tones, the DBTMA scheme can achieve performance gain as high as 140% over MACA and FAMA-NCS. We believe that this performance gain is high enough to offset the bandwidth consumption of the busy tones. The performance gain of the DBTMA scheme over the RI-BTMA scheme is about 20%, with the help of an extra busy tone and without the requirement of precise global time synchronization.

We believe that this gain is a good incentive to incorporate required hardware at the network nodes. Similar result is also shown in the work by Gummalla and Limb [13] for high speed distributed wireless LAN and we have extended it to ad hoc networks. The novel wireless transceiver architecture proposed and studied in [13] can also be used for the DBTMA scheme to set up the busy tones with small hardware cost. So the advantage of using DBTMA over using other MAC schemes in ad hoc networks is significant.

In our protocol, we have assumed that the inter-channel interference between busy tone signals and data signal is negligible. This might not be the case in the real networks. Careful hardware design may help to minimize the effect of possible interference. Some modifications of the DBTMA protocol might be required.

In this paper, we have presented the DBTMA protocol and we have analyzed its performance under various network conditions. Our analytical and simulation results show that the DBTMA protocol is superior to other schemes that rely on RTS/CTS dialogues on a single channel or those that rely on a single busy tone.

VII. Acknowledgments

We'd like to thank three anonymous reviewers and the editor, Prof. Yuguang Fang, for their valuable comments and suggestions toward the paper.

Appendix I: DBTMA Operation Rules

A. Variable Definitions

- δ: data packet transmission time.
- τ: maximum one way propagation delay.
- t_d: busy tone detection delay.
- t_{wm}: mandatory waiting time ($t_{wm} = 2\tau$).
- BI: backoff interval. [7]

B. Communication Rules

- (**Initialization**) Upon powering up, a node goes into the IDLE state. We assume that both the transmitter ($\mathcal{A}$) and the receiver ($\mathcal{B}$) are in the IDLE state before the transmission.
- (**Send RTS**) When $\mathcal{A}$ receives a data packet for transmission to the destination $\mathcal{B}$, it tries to sense the BT_r and the BT_t signals. If no busy tone signal is sensed, it turns on its BT_t signal, sends an RTS packet to $\mathcal{B}$, and goes into the S_RTS state. If $\mathcal{A}$ senses a busy tone signal, it sets a random timer (chosen among $[0, BI]$) and goes into the CONTEND state.
- (**Wait for BT_r**) At the end of the RTS transmission, $\mathcal{A}$ turns off its BT_t signal, sets a timer of $(t_d + 2\tau)$ second, and goes into the WF_BTR state.
- (**Wait for data**) When $\mathcal{B}$ receives the RTS packet from $\mathcal{A}$, it sets up its BT_r signal, sets a timer of $(\delta + t_d + 2\tau)$ second, and goes into the WF_DATA state.
- (**Mandatory wait**) When $\mathcal{A}$ senses a BT_r signal in the WF_BTR state, it sets a timer of ($t_{wm} = 2\tau$) second and goes into the WAIT state.
- (**Send data**) Upon the timeout in the WAIT state, $\mathcal{A}$ transmits the data packet and goes into the S_DATA state.
- (**End of transmission**) At the end of the DATA transmission, $\mathcal{A}$ goes into the IDLE state.
- (**Receive data**) When the data packet arrives or timeout takes place in the WF_DATA state, $\mathcal{B}$ sets off the BT_r signal and goes into the IDLE state.
- (**Contend**) Upon timeout in the CONTEND state, $\mathcal{A}$ tries to sense the BT_r and the BT_t signals again. If no busy tone signal is sensed, it turns on its BT_t signal, sends an RTS packet to $\mathcal{B}$, and goes into the S_RTS state. If $\mathcal{A}$ senses a busy tone signal, it goes back into the IDLE state.
- (**Timeout**) Upon timeout in the WF_BTR state, $\mathcal{A}$ goes into the IDLE state.

C. Defer Rules

- (**Abort RTS**) When a node senses the BT_r signal during the transmission of its RTS packet, it turns off its BT_t signal, aborts the transmission, and goes into the IDLE state.

Appendix II: Proof of Collision-free Data Packet Reception

In Section IV, we claimed that with the help of the BT_r signal and the mandatory waiting time (t_{mw}), when the RTS packet is successfully received and the BT_r signal is set up, data packet reception will be guaranteed. Hence, the DBTMA scheme guarantees collision-free data packet reception. While the claim is made in a fully connected network, we will prove it in regular Ad Hoc networks.

Lemma 1: *No RTS packets would collide with data packet reception at the receiver.*

Proof: Suppose the receiver (node B) receives the RTS packet correctly at time t_0. At time t_0, it sets up its BT_r signal. Since every neighbor of the receiver is at most τ seconds away, the BT_r signal will reach all neighbors at time $t_1 = t_0 + \tau$. They will be able to detect the busy tone at time $t_2 = t_1 + t_d = t_0 + \tau + t_d$. So no RTS packets will be sent after t_2 in the range of the receiver.

Hence the receiver can be sure that all RTS transmissions will be cleared since $t_3 = t_2 + \tau = t_0 + 2\tau + t_d$.

The earliest time the sender (node A) senses the BT_r signal from the receiver is $t_4 = t_0 + t_d$. Because of the mandatory waiting time (t_{mw}), its data packet transmission will not start until $t_5 = t_4 + t_{mw} = t_0 + t_d + t_{mw}$, which is the earliest time the beginning edge of the data packet arriving node B.

[7] The backoff interval should be dynamically controlled by a backoff algorithm, such as BEB and MILD. For simplicity, one may use 10γ as the BI value, as suggested by Fullmer and Garcia-Luna-Aceves in [11].

Thus, we are sure that the data packet is free from RTS collisions if $t_5 \geq t_3$ is satisfied, which is ensured by our assumption $t_{mw} = 2\tau$.

Lemma 2: *No other data packets would collide with data packet reception at the receiver.*

Proof: Suppose there is a node (node C, which in the range of the receiver node B) τ' second away from node B. Since node B gets the RTS packet successfully at time t_0, node C must have not been sending any packet in $[t_0 - \gamma - \tau', t_0 - \tau']$. Otherwise, the transmission from node C would have collided the incoming RTS packet at node B.

Since node C has been silent for a period of time, the only possibility of its data packet transmission would be that it has sent its RTS packet to another node already and is waiting for its busy tone reply in the quiet period. We now prove this is impossible: The latest time for node C to finish its RTS transmission was $t_6 = t_0 - \gamma - \tau'$. Hence the latest time for node C to start sending its data packet would be $t_7 = t_6 + \tau + t_d + \tau + t_{mw}$, where we have assumed node C's intended receiver is τ second away. Remember that $\gamma \geq t_d + 4\tau$:

$$\begin{aligned} t_7 &= t_6 + \tau + t_d + \tau + t_{mw} \\ &= t_0 - \tau' - \gamma + t_d + 4\tau \\ &\leq t_0 - \tau' - (t_d + 4\tau) + t_d + 4\tau \\ &= t_0 - \tau' \end{aligned} \tag{8}$$

This is contradictory to the fact that node C was not sending any packet in $[t_0 - \gamma - \tau', t_0 - \tau']$.

Theorem: *When RTS packet is received correctly and BT_r set up by the receiver, collision-free data packet reception is guaranteed.*

Proof: This can be proved by Lemma 1 and 2. Since neither RTS packet nor data packet would collide with the data packet at the receiver, data packet is free from collisions.

References

[1] J. C. Haartsen, "The Bluetooth Radio System," *IEEE Personal Communications*, vol. 7, no. 1, pp. 28-36, Feb. 2000.

[2] Specification of the Bluetooth System; available online at http://www.bluetooth.com/developer/specification/ specification.asp.

[3] L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels: part I - Carrier Sensing Multiple-Access Modes and Their Throughput-Delay Characteristics," *IEEE Trans. Commun.*, vol. COM-23, no. 12, pp. 1400-1416, Dec. 1975.

[4] A. Colvin, "CSMA with collision avoidance," *Computer Communications*, vol. 6, no. 5, pp. 227-235, Oct. 1983.

[5] P. Karn, "MACA - A new channel access method for packet radio," in *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pp. 134-140, 1990.

[6] 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 Trans. Commun.*, vol. COM-23, no. 12, pp. 1417-1433, Dec. 1975.

[7] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A media access protocol for wireless LAN's," in *Proc. ACM SIGCOMM '94*, pp. 212-225, 1994.

[8] F. A. Tobagi and L. Kleinrock, "Packet switching in radio channels: part III - Polling and (Dynamic) Split-Channel Reservation Multiple Access," *IEEE Trans. Commun.*, vol. COM-24, no. 8, pp. 832-844, Aug. 1976.

[9] C. Wu and V. O. K. Li, "Receiver-initiated busy-tone multiple access in packet radio networks," in *Proc. ACM SIGCOMM '87*, pp. 336-342, 1987.

[10] C. L. Fullmer, J. J. Garcia-Luna-Aceves, "Floor acquisition multiple access (FAMA) for packet-radio networks," in *Proc. ACM SIGCOMM '95*, pp. 262-273, 1995.

[11] C. L. Fullmer, J. J. Garcia-Luna-Aceves, "Solutions to hidden terminal problems in wireless networks," in *Proc. ACM SIGCOMM '97*, pp. 39-49, 1997.

[12] IEEE 802.11, "Wireless LAN MAC and Physical Layer Specifications," Jun. 1997.

[13] A. C. V. Gummalla and J. O. Limb, "Design of an Access Mechanism for a High Speed Distributed Wireless LAN," *IEEE JSAC*, vol. 18, no. 9, Sep. 2000.

[14] H. Takagi and L. Kleinrock, "Output processes in contention packet broadcasting systems," *IEEE Trans. Commun.*, vol. COM-33, no. 11, pp. 1191-1199, Nov. 1985.