# COLLISION-FREE MEDIUM ACCESS CONTROL SCHEME FOR AD-HOC NETWORKS

Zygmunt J. Haas and Jing Deng
School of Electrical Engineering
Cornell University
Ithaca, NY 14853
haas@ee.cornell.edu

Siamak Tabrizi
US Air Force
Air Force Research Laboratory
Rome, NY 13441

Abstract – The Dual Busy Tone Multiple Access (DBTMA) scheme was designed for decentralized multi-hop networks. In DBTMA, the RTS/CTS dialogue is used to reserve the channel. In addition, two busy tones are employed to eliminate collisions between the RTS/CTS control packets and DATA packet transmissions. DATA packet collisions can seriously decrease channel utilization for a MAC protocol, as DATA packets are much longer than control packets. Since DBTMA completely avoids DATA packet collisions, its performances are superior to other proposed protocols. In this paper we analyze the possible cases of DATA packet collision and prove that it is collision-free.

## I. INTRODUCTION

In an ad hoc network, a single channel is shared by a number of nodes. Packet collisions are unavoidable due to the fact that traffic arrivals are random and there is a non-zero propagation time between any pair of transmitter and receiver. Medium Access Control (MAC) schemes are used to coordinate access to the single channel in the network. Due to the non-transitivity property of the radio communication, the well-known *hidden terminal problem* [1] [2] and the *exposed terminal problem* [2] may occur. These problems severely affect the channel utilization of MAC protocols.

In a communication network with randomly accessing nodes, DATA packet collisions are the main obstacle to obtain high network utilization. It's the design objective of MAC protocols to decrease the probability of DATA packet collisions. Thus a MAC protocol with DATA packet collision-free property is attractive, especially for wireless networks such as ad hoc networks. Based on our previous work on the DBTMA protocol [3], we claim and prove that DBTMA is a DATA packet collision-free protocol. This helps us in understand the reasons for the superior network utilization of DBTMA, as compared with other proposed protocols for ad hoc networks.

We discuss related works in next section. An example and detailed description of our DBTMA protocol are included in the third section. In the fourth section, we give our proof of its DATA packet collision-free characteristic. The fifth section concludes our work.

## II. RELATED WORKS

In order to use the single shared channel efficiently, the well-known hidden terminal problem needs to be solved by MAC protocols in multi-hop networks. The situation, when one node is in the range of the receiver but not the transmitter, can lead to increased number of collisions; such a node is referred to as a *hidden terminal*. Since it is out of the range of the transmitter, the node is not able to sense the on-going transmission, and it needs to be notified explicitly. An exposed terminal is a node which is in the range of the transmitter but not the receiver. The exposed terminal may transmit at the same time as the transmitter. However, using the traditional Carrier Sensing Multiple Access (CSMA) scheme, the exposed terminal will defer from accessing the channel, lowering the network capacity.

Tobagi [1] introduced a protocol that uses a busy tone to solve 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. While receiving, the central base station sends out a busy tone signal to all nodes and keeps them (except the current transmitter) from accessing the channel. Hidden terminals also sense the busy tone and back-off. However, the original BTMA protocol cannot be used in ad hoc network because of lack of central stations in such networks.

In Multiple Access Collision Avoidance (MACA) [2], Karn originally proposed the use of short control packets, the Request-To-Send (RTS) and 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 to the "transmitter" with a CTS packet. All other nodes that hear the RTS packet back-off for time long enough for the "transmitter" to receive and to respond to the CTS packet. All nodes that hear the CTS packet back-off for time long enough for a data packet reception. MACA reduces the probability of data packet collisions by introducing extra message exchange. It also solves the hidden terminal problem with the RTS/CTS dialogue.

Bharghavan [4] 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 that it is using the channel. The ACK packet allows conveyance of fast acknowledgments back to the transmitter. A new back-off algorithm, the MILD algorithm, was also introduced to solve some of the unfairness problems of MACA. Additional features of the MILD algorithm, such as back-off value copying and multiple back-off values for different destinations, further improved MACAW performance. However, because of the use of the DS and the ACK message, exposed terminal problem was reintroduced.

Floor Acquisition Multiple Access (FAMA) [5] is a refinement of MACAW protocol. A non-persistent CSMA scheme is used at the beginning of each free slot. Each ready node has to compete for the channel (the floor) before they can use the channel to transmit data. FAMA ensures that no data packet collides with any other packet, given that there is no hidden terminals. It also claims that FAMA performs as well as MACA under the hidden terminal situation and as well as CSMA otherwise.

The RTS/CTS dialogue is supposed to prevent all other nodes in the receiver's range from transmitting. However, CTS packets can still be destroyed by collisions. Our analysis and simulation show that the probability of CTS packet collision in a multihop network that uses the basic RTS/CTS dialogue rules can be as high as 60%, when the network operates at high traffic load. In the DBTMA protocol, we proposed to use two out-of-band tones, $BT_r$ (the receive busy tone) and $BT_t$ (the transmit busy tone) to decouple the communications in the two directions. In particular, the $BT_r$ signal addresses the hidden terminal and the exposed terminal problems. We claim that DBTMA is a DATA packet collision-free protocol, because of the combined use of RTS/CTS dialogue and the two busy tones.

## III. DBTMA

In the DBTMA scheme, the single shared channel is split into two sub-channels: the *data channel* and the *control channel*. Data packets are transmitted on the data channel. Control packets (e.g., RTS and CTS) are transmitted on the control channel. Two narrow-band tones (the busy tones, $BT_r$ and $BT_t$) are added to the control channel with enough spectral separation. $BT_t$ (the transmit busy tone) and $BT_r$ (the receive busy tone) indicate that the node is transmitting and receiving on the data channel, respectively.

DBTMA operates as follows: A ready node (the "transmitter") sends out an RTS packet to request the channel. When the intended destination (the "receiver") receives the RTS packet and decides that it is able to receive the data packet, it sets up its $BT_r$ signal and replies with a CTS packet. Upon receiving the CTS packet, the source sets up its $BT_t$ signal and starts the data transmission. All other nodes sensing the $BT_r$ signal defer from transmitting. All other nodes sensing the $BT_t$ signal determine that they cannot receive. Through this mechanism, hidden terminals back-off and exposed terminals will be allowed to use the channel, since hidden terminals can sense the $BT_r$ signal and exposed terminals can sense the lack of the $BT_r$ signal.

A ready node has to sense the $BT_r$ signal first. It can only send out its RTS packet when it doesn't sense any $BT_r$ signal. It also keeps on sensing the $BT_r$ signal until the end of its RTS packet transmission, because other nodes might set up the $BT_r$ during this time. If it senses a $BT_r$ signal in this period, it defers from transmitting even it receives the CTS packet from its intended receiver. On the receiver side, when it receives an RTS packet destined to itself, it senses the $BT_t$ signal to see if there is any node in range transmitting on the data channel. If not, it decides that it can use it to receive and replies with a CTS packet. It keeps silent otherwise. More detailed description of our protocol in the Appendix A.

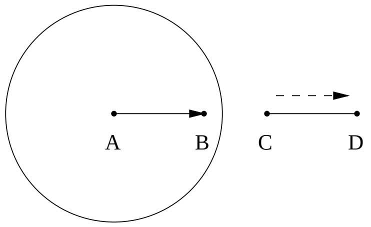


Figure 1: Possible DATA Packet Collisions

## IV. COLLISION-FREE PROPERTY

To prove that in DBTMA there are no DATA packet collisions, we use the following denotations:

- $\tau$: the maximum propagation time between two neighbor nodes. We also denote $\tau_{ab}$ the propagation time between node A and node B;
- $\gamma$: the transmission time of a control packet (RTS or CTS);
- $\delta$: the transmission time of a DATA packet. $\delta_h$ is denoted as the transmission time of the DATA header.

The following are the assumptions used in our discussion:

- We assume that there is no interference between the control channel and the DATA channel;
- We neglect the packet processing time, the transmit-to-receive turn-around time, the busy tone set up/off time and the busy tone sensing time;
- $\gamma > 2\tau$ (In fact, $\gamma$ should be greater than $2\tau$ plus all the necessary processing time that we have assumed negligible.);
- A "transmitter" keeps on sensing the $BT_r$ signal until the end of its RTS packet transmission. It defers from accessing the channel if it senses a $BT_r$ signal during this period[1].
- If a receiver doesn't receive the header of a DATA packet after $2\tau + \delta_h$ seconds, it sets off its $BT_r$ signal and goes into the IDLE state.

[1] It will not defer from transmission because of the $BT_r$ signal from its intended receiver, since the receiver will not set up the $BT_r$ signal until it receives the whole RTS packet.

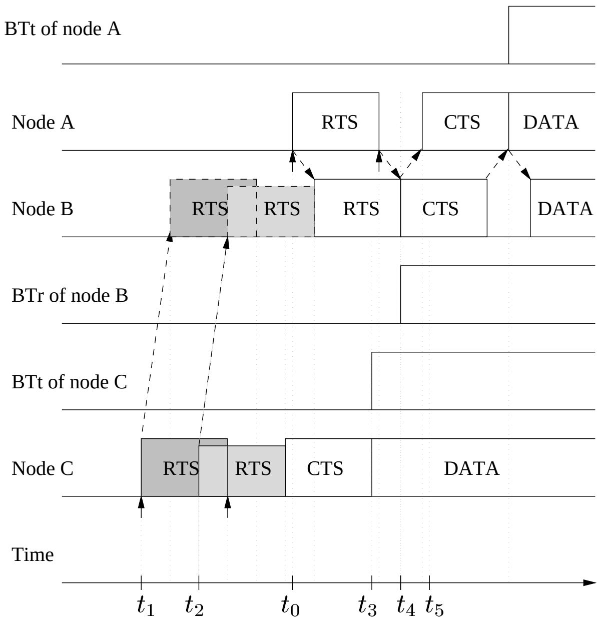


Figure 2: Time Diagram of DBTMA

- We assume that a node that hears a collided control packet on the control channel backs-off for a time duration of $\lambda + 2\tau$ seconds.

We will discuss a general case, shown in Fig. 1, where node A sends an RTS packet to node B at time $t_0$. The packet is received by node B successfully. Note that if this RTS packet is not received correctly, no DATA transmission will be taken place. Node B will receive the RTS packet at time $t_4 = t_0 + \tau_{ab} + \gamma$.

DATA packet collisions can happen only when node C (or any other hidden terminal) transmits its DATA packet during the time node B is receiving the DATA packet from node A. There are five cases of possible DATA packet collisions:

[i] If node C sends out its RTS packet to request channel before $t_1 = t_4 - \tau_{bc} - 2\gamma - 2\tau$ (Fig. 2), it will receive a CTS packet, if there is any, from its intended destination (e.g. node D) before $t_3 = t_4 - \tau_{bc}$ (note that $2\gamma + 2\tau$ is the maximum time for an RTS/CTS dialogue). Node C will set up its $BT_t$ signal before time $t_3$. The $BT_t$ signal will reach node B $\tau_{bc}$ seconds later, before time $t_4$. When node B senses the busy tone at $t_4$, it will defer from replying with a CTS packet and the DATA packet from node A will not be sent;

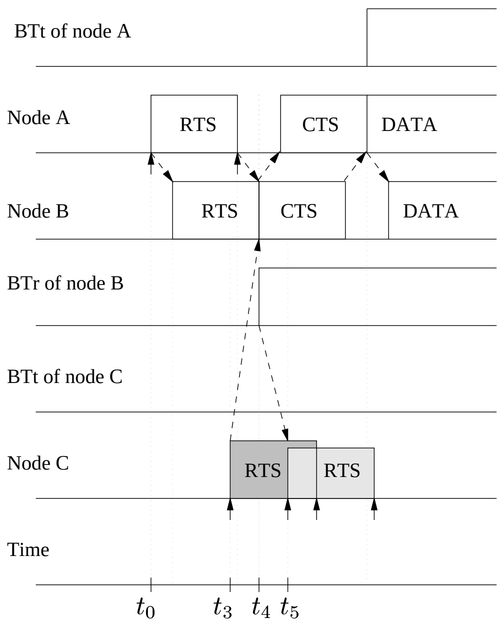


Figure 3: Time Diagram of DBTMA (2)

[ii] If node C sends out its RTS packet at the time between $t_1$ and $t_2 = t_4 - \tau_{bc} - 2\gamma$, node B will overhear the RTS packet (or destroyed RTS, packet if it has collided) and go into the QUIET state at time $t_1 + \tau_{bc} + \gamma + 2\tau + \gamma = t_4$. This makes sure that node B keeps silent when it receives the RTS packet from node A. The DATA packet from node A will not be sent either;

[iii] If node C sends out its RTS packet at the time between $t_2$ and $t_3 = t_4 - \tau_{bc}$, it will destroy the RTS packet from node A at node B. Node B will not receive the RTS packet from node A correctly. The DATA packet from node A will not be sent;

[iv] If node C sends out its RTS packet at the time between $t_3$ and $t_5 = t_4 + \tau_{bc}$ (Fig. 3), node C will sense the $BT_r$ signal from node B at the end of its RTS transmission and defers from sending its DATA packet. Thus, there will be no DATA packet collision in this case;

[v] Finally, it is impossible for node C to send out its RTS packet after $t_5$, since it senses the $BT_r$ signal from node B.

From the above discussions, we find that there will be no DATA packet collisions in all the possible cases. Observe that the it is independent of node C's position. This observation concludes our proof of DATA packet collision-free property for DBTMA.

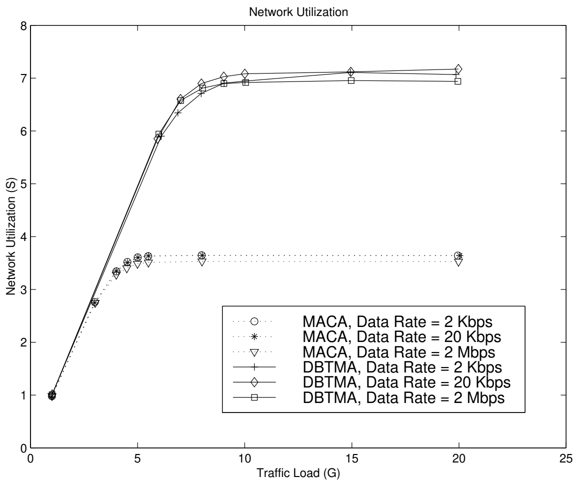


Figure 4: Simulation Results

Simulations were run to find the throughput performance of the DBTMA scheme in ad hoc networks. We also ran simulations for basic RTS/CTS scheme (like MACA) on the same network. Our simulations were implemented in an event driven C program. We used a network with 400 nodes and a $6 \times 6$ $km^2$ coverage area. The transmission range R is 1 km and the link data rate is varied from 2.048 Kpbs to 2.048 Mbps.

The results of our simulations are given in Fig. 4. It shows that DBTMA yields about 100% improvement in the network utilization, as compared with other RTS/CTS-based schemes. We can see a slight difference in the results for different link data rates. The reason for this behavior is that the propagation delay is more significant at higher link data rates, which leads to more control packet collisions.

## V. SUMMARY

The main objective of MAC protocols is to synchronize access of multiple nodes to the shared communication medium, while maintaining high network utilization. The hidden terminal problem and the exposed terminal problem need to be solved in order to improve the performance of a MAC scheme in multi-hop networks.

In our DBTMA scheme, each node uses two out-of-band busy tone signals in addition to the use of the RTS/CTS dialogue. These tones serve as notification for all nodes in the transmission range and in the reception range of the node in question. This mechanism assures that hidden terminals back-off and the exposed terminals do not defer. Since the busy tones are narrow band signals, the additional hardware requirements are limited. Sensing of the busy tones is simple and can be implemented through a narrow band filter and comparator. By using busy tones in our MAC protocol, the transmission of various packet length is also possible.

## A. DBTMA OPERATION RULES

### General Rules

- RTS and CTS packets are used to initialize a data packet transmission.
- Before a node starts to transmit a data packet, it sets up the $BT_t$ signal until the transmission is completed.
- Before a node replies to the initiator with a CTS packet, it sets up the $BT_r$ signal until the reception is completed.
- (**Initialization**) After powering up, every node goes into the IDLE state.

### Communication Rules

- Both the transmitter ($\mathcal{A}$) and the receiver ($\mathcal{B}$) are in the IDLE states before the transmission.
- When $\mathcal{A}$ receives a data packet for transmission to the destination $\mathcal{B}$, it goes into the CONTEND state.
- In the CONTEND state, $\mathcal{A}$ tries to sense the $BT_r$ signal. It goes into next step only if it doesn't sense any $BT_r$ signal. Otherwise it backs-off.
- $\mathcal{A}$ transmits an RTS packet, sets up a timer and goes into the WF-CTS state.
- $\mathcal{B}$ receives the RTS packet from $\mathcal{A}$. It tries to sense the $BT_t$ signal. It goes into next step only if it doesn't sense any $BT_t$ signal. Otherwise it stays in the IDLE state.
- $\mathcal{B}$ sets up the $BT_r$ signal and sends out a CTS packet. Then it sets up a timer and goes into the WF-DATA state.
- $\mathcal{A}$ receives the CTS packet from $\mathcal{B}$. It sets up the $BT_t$ signal and sends out the data packet.
- $\mathcal{A}$ transmits the data packet. It sets off the $BT_t$ signal and goes into the IDLE state.
- $\mathcal{B}$ receives the data packet from $\mathcal{A}$. It sets off the $BT_r$ signal and goes into the IDLE state.

### Back-off Rules

- When there is a timeout in the WF-CTS state, a node will increase it's back-off value according to: $F_{inc}(x) = min(1.5 \cdot x, BO_{max})$, where $BO_{max}$ is the maximum back-off value.
- When a node in the WF-CTS state receives a CTS packet destined to itself, it decreases it's back-off value according to: $F_{dec}(x) = max(x - 1, BO_{min})$, where $BO_{min}$ is the minimum back-off value.

## REFERENCES


[1] F. A. Tobagi and L. Kleinrock, "Packet Switching in Radio Channels: Part II - the Hidden Terminal Problem in Carrier Sense Multiple-Access Modes and the Busy-Tone Solution," *IEEE Trans. Commun.,* vol. COM-23, no. 12, pp. 1417-1433, 1975.

[2] P. Karn, "MACA - A New Channel Access Method for Packet Radio," in *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pp. 134-140, ARRL, 1990.

[3] Z. J. Haas and J. Deng, "Dual Busy Tone Multiple Access (DBTMA) - Performance Evaluation," in *IEEE VTC'99*, Houston, TX, May 17-21, 1999.

[4] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LAN's," in *SIGCOMM '94*, pp. 212-225, ACM, 1994.

[5] C. L. Fullmer, J. J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Access (FAMA) for Packet-Radio Networks," in *SIGCOMM'95*, pp. 262-273, ACM, 1995.

[6] L. Kleinrock and F. A. Tobagi, "Packet Switching in Radio Channels: Part I - Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics," *IEEE Trans. Commun.* , vol. COM-23, no. 12, pp. 1417-1433, 1975.