DUAL BUSY TONE MULTIPLE ACCESS (DBTMA) - PERFORMANCE EVALUATION

Zygmunt J. Haas, Jing Deng

School of Electrical Engineering
323 Frank Rhodes Hall
Cornell University
Ithaca, NY 14853
e-mail: haas@ee.cornell.edu

Abstract – The Dual Busy Tone Multiple Access (DBTMA) scheme was designed for decentralized networks such as Packet Radio Networks (PRN's). We use the RTS/CTS dialogue to reserve the channel in DBTMA. In addition, we use two busy tones to notify neighbor nodes of the on-going transmission and reception after a successful channel reservation. Our analysis and simulations show that DBTMA provides higher network utilization, which is 100% more than the basic RTS/CTS schemes. We present the details of our analysis in this paper. In order to evaluate our protocol in a more general network, our analysis and simulations are based on a fully-meshed decentralized multi-hop network topology.

I. INTRODUCTION

When the network span is much larger than nodal transmission range, nodes in a peer-peer network may need to use multi-hop communications. An example of multi-hop network is a Packet Radio Network (PRN). In a PRN, a single channel is shared by a number of nodes. Packet collisions are unavoidable due to the fact that transmissions are random and that there is a none-zero propagation time between any two nodes. Medium Access Control (MAC) schemes are used to coordinate access to the single channel. 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 this paper, we review different MAC protocols used in PRN's. To improve the network utilization and delay performance in PRN, we have introduced a new protocol, named Dual Busy Tone Multiple Access (DBTMA) [3]. In DBTMA, we use two out-of-band busy tones [1] to notify neighbor nodes of the on-going transmission. Nodes sensing the busy tones back-off, avoiding packet collision. We evaluate, by simulation and through theoretical analysis, the throughput of our protocol.

The next section of this paper discusses related works, followed by a section describing the details of DBTMA. We provide theoretical analysis results in the fourth section. The fifth section concludes the work.

II. RELATED WORKS

In order to obtain higher network utilization, MAC protocols for PRN's need to solve the hidden terminal problem and the exposed terminal problem. The situation when one node is in the range of the receiver, but not the transmitter, can lead to increased number of collisions; and the 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. Similarly, an exposed terminal is a node which is in the range of a transmitter but not a receiver. The exposed terminal may transmit at the same time as the other transmitter. However, using 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

[1] Busy tones are implemented as transmission of unmodulated carrier waves. Thus, the required bandwidth is very small.

base station sends out a busy tone signal to all nodes and keeps them (except the transmitter) from using the channel. In a communication network with base stations, hidden terminals sense the busy tone and back-off. Since there are no base stations in the PRN network model, the original BTMA protocol cannot be used in our work.

In 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 caused by hidden terminals through the use of 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 was implemented for acknowledgment. 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.

MACA-BI was proposed by Talucci [5]. In a network with periodic data traffic, the receiver can predict its future reception times. It initiates the reception by sending out an RTR (Ready-To-Receive) message to request the data. By suppressing the RTS packet, MACA-BI improved its performance. However, since most transmission instances cannot be predicted due to the burstiness of data traffic, the scheme may not be practical. In fact, in the case of periodic data traffic, one can use more efficient protocols, e.g., reservation type protocol.

The RTS/CTS dialogue is supposed to prevent all other nodes in the receiver's range from transmitting. However, the probability of CTS packet collision remains. In the case of high traffic load, nodes may try to access the channel frequently, destroying the CTS packet reception at the receiver. Our analysis and simulation showed that the probability of CTS packet collision in a network that uses the basic RTS/CTS dialogue rules is very high (and can be as high as 60%), when the network operates at high traffic load. In our protocol, we use two out-of-band tones, BT_r (the receive busy tone) and BT_t (the transmit busy tone) to decouple communications in the two directions. In particular, the BT_r signal addresses the hidden terminal and the exposed terminal problems.

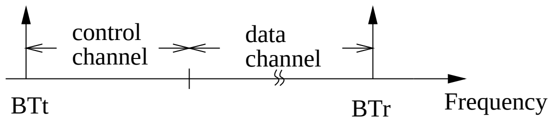

Figure 1: Frequency Chart

III. DBTMA

In our DBTMA scheme, we split the single shared channel 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), shown in Fig. 1, indicate that the node is transmitting and receiving on the data channel, respectively.

Our protocol can minimize the probability of data packet collisions with the use of the busy tones. The operation of the scheme is as following: A ready node (the source) sends out an RTS packet to request the channel. When the destination node (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 the source receiving the CTS packet, it 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.

Back-off algorithm is another issue addressed in our work. Bharghavan [4] pointed out the problem with using Binary Exponential Back-off (BEB) algorithm as a back-off procedure. In BEB, a node doubles its back-off value after each collision or time-out. Upon successful transmission, a node resets its back-off value to the minimum. The BEB scheme can result in significant unfairness when two nodes with high traffic load compete on the channel. One node dominates in its use of the channel, while the other one backs-off with maximum back-off value. To solve this problem, upon success, the MILD algorithm [4] increases back-off values by the factor of 1.5 and decreases back-off values by one step, which is the sum of round trip propagation time of the network. MILD also implements back-off value copying and different back-off values for different destinations. A sender includes its back-off value in a special field of the data packet it sends out. Back-off value copying is a scheme in which every node that hears a successful data packet transmission copies the back-off value in the packet. MILD also keeps separate back-off values for different destinations in order to distinguish the traffic load among different destination nodes.

In our protocol, we adopted the MILD algorithm from MACAW, but not the back-off copying or the multiple back-off values, because we found that copying the back-off value would unnecessarily fluctuate, while different back-off values migrate among areas with different traffic loads.

We present the details of our protocol in the Appendix B.

IV. PERFORMANCE EVALUATION

We have simulated the performance of DBTMA, using the following assumptions:

- Each node is a Poisson source with a packet arrival rate of λ. The packet length is constant at 1024 [bits].
- Nodes are distributed throughout the PRN area randomly, with density $\rho\,[nodes/km^2]$.
- We neglect the control packet processing time. We also neglect nodal mobility and the time to process acknowledgments. In particular, we have shown in [3] that nodal mobility is not a major source of packet collisions.
- We use single channel for data packets and control packets in our performance evaluation.

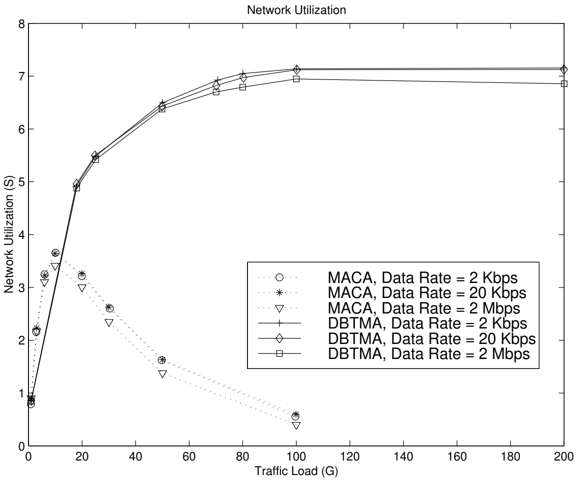

Figure 2: Simulation Results (R = 1 km)

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 and we varied the transmission range (R=0.5 km, 1 km) and the link data rate (2.048, 20.48 and 2048 Kbps). For comparison purposes, we also run the simulation for the RTS/CTS-based scheme (similar to the MACAW scheme [4]).

The results of our simulations are given in Fig. 2, Fig. 3, and Fig. 4. Fig. 2 and Fig. 3 show results under the assumption that every unsuccessful packet is discarded at the transmitter. In Fig. 2, we show the results with the transmission range of 1 km. We find 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 packet collisions. We draw a similar conclusion from Fig. 3 with transmission range of 0.5 km. We also find that the network utilization in the case of $R = 0.5\ km$ is about 4 times the utilization with $R = 1\ km$. The explanation for this behavior is that the number of possible concurrent transmissions is proportional to $1/R^2$. In Fig. 4, we show our simulation results based on the assumption that an unsuccessful packet backs-off, rather than being discarded. In these results, we find more stable network utilization in high traffic load. This is because of our back-off algorithm used in these simulations.

We have also analyzed DBTMA theoretically. Using the results we have obtained in Appendix A, we

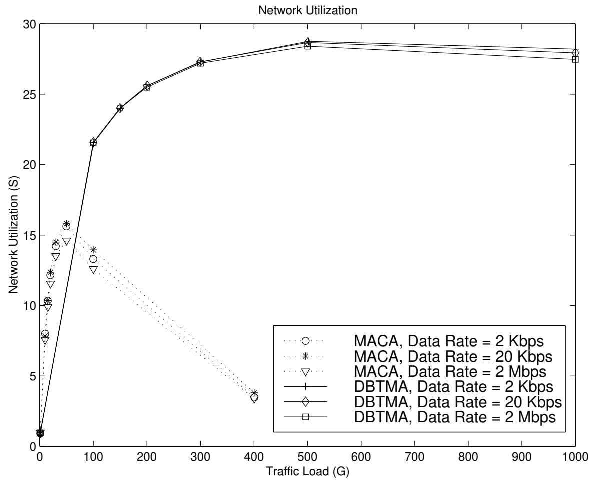

Figure 3: Simulation Results (R = 0.5 km)

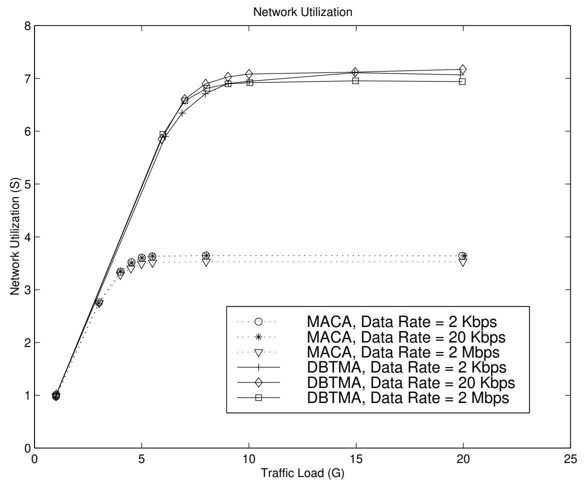

Figure 4: Simulation Results (R = 1 km) with Back-off Function

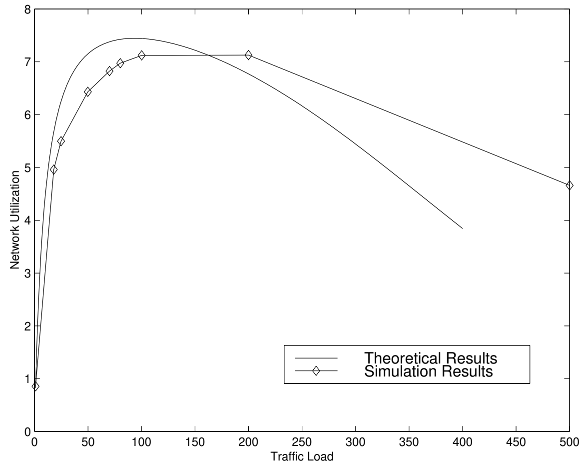

Figure 5: Theoretical Results vs. Simulation Results

compare our results in Fig. 5. Although the simulation results mostly confirm our analytical results, there are some discrepancies between the two. After achieving network capacity, the theoretical results tend to drop faster than the simulation results. Our explanation is that in our analysis, we pick one transmission area and calculate its throughput, assuming that nodes outside of this area will not affect the transmission inside. However, this is not true in practical networks. In high traffic loads, nodes outside the area in question might drive some hidden terminals back-off, which increases the probability of control packets success and the channel throughput.

V. SUMMARY

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

Some researchers in this field have proposed to totally abandon the carrier-sensing schemes and to rely on a reservation dialogue (the RTS/CTS dialogue) among the communication nodes. Examples of such schemes are the MACA and the MACAW protocols. These schemes, indeed, provide significant improvement to the Carrier Sensing schemes. However, the RTS/CTS-based protocols can still be quite vulnerable to multiple collisions.

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. This mechanism assures that hidden terminals back-off and the exposed terminals do not defer. Our theoretical analysis and simulations show the performance of our protocol to be superior to that of other basic RTS/CTS schemes.

A. THEORETICAL ANALYSIS

Our theoretical analysis is similar to that of Kleinrock and Tobagi [6]. We simplify our analysis, assuming that every unsuccessful data packet is destroyed by the transmitter. We define a parameter called *Maximum Number of Concurrent Transmissions* (M_c) in the coverage area. Using hexagons with radius R to cover the area, the number of hexagons can be determined as $M_c =$

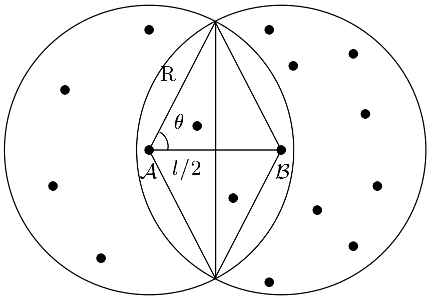

Figure 6: Scenario for Throughput Calculation

$S_{PRN}/(2\sqrt{3}R'^2)$, where S_{PRN} is the size of the coverage area and $R' = (R + (\sqrt{3}/2)R)/2$ (we use this approximation because transmission region is represented by a circle, but hexagons are used to cover the area seamlessly).

The probability of a successful data packet transmission is:

$$\begin{aligned} P_s(l) &= Pr\{RTS\ successful\} \cdot \\ &\quad Pr\{CTS\ successful \mid RTS\ successful\} \cdot \\ &\quad Pr\{data\ successful \mid CTS\ successful\} \end{aligned}$$

The probability of a successful RTS packet (sent by node $\mathcal{A}$ in Fig. 6) is that it is the only transmission in the range of the destination (node $\mathcal{B}$) during its vulnerable time:

$$Pr\{RTS\ successful\} = e^{-(2\gamma+\tau)\lambda(\rho\pi R^2-1)}$$

where γ is the transmission time of a control packet, τ is the propagation delay, λ is the data packet arrival rate at each node, ρ is the nodal density, and R is the transmission radius.

After receiving the RTS packet and deciding that it can receive, node $\mathcal{B}$ sets up its BT_r signal and replies with a CTS packet. Since the BT_r reaches all nodes in node $\mathcal{B}$'s range in τ second, the vulnerable period for the CTS packet is only τ. The CTS packet will be successful if no other nodes start transmission during the vulnerable period. Hidden terminals are the nodes that potentially can start transmitting during this period:

$$\begin{aligned} & Pr\{CTS\ successful \mid RTS\ successful\} \\ = \ & e^{-\tau\lambda N_{ht}} \end{aligned}$$

where N_{ht} is the expected number of nodes as hidden terminals. We calculate it with the use of θ (as defined in Fig. 6):

$$N_{ht} = R(\pi R + l\sin\theta - 2R\theta)$$

We simplify our analysis by approximating $Pr\{data\ successful \mid CTS\ successful\}$ to be 1. Since the BT_t signal and the BT_r signal are set up and the channel is reserved before we transmit a data packet, this approximation is reasonable.

With the above equations, we can express the probability of a successful data packet transmission as a function of l, the distance between the source and the destination nodes:

$$P_s(l) = e^{-\lambda\left[(2\gamma+\tau)(\rho\pi R^2-1)+\tau\rho R(\pi R+l\sin\theta-2R\theta)\right]}$$

We define a busy period as the period between two consecutive idle periods. There are two types of busy periods: busy periods with successful data transmissions and those with collisions. The expected time of a busy period is then:

$$\overline{B} = T_s P_s(l) + T_f\,(1 - P_s(l))$$

where T_s is the expected time of successful transmission period, and T_f is the expected time of unsuccessful period. A successful transmission period consists of an RTS packet transmission time followed by τ, a CTS packet transmission time followed by τ, and a data packet transmission time (δ) followed by τ. Thus the length of a successful cycle is:

$$T_s = 2\gamma + 3\tau + \delta$$

An unsuccessful transmission period consists of an RTS packet transmission time followed by τ and a collision time before the channel becomes idle again [6]:

$$T_f = \gamma + 2\tau - \frac{1 - e^{-\tau\lambda\rho\pi R^2}}{\lambda\rho\pi R^2}$$

The average utilization period can be expressed as:

$$\overline{U} = \delta \cdot P_s(l)$$

An idle period is the time between two consecutive busy periods. According to the property of a Poisson process, we can express the expected time of an idle period as:

$$\overline{I} = \frac{1}{\lambda\rho\pi R^2}$$

The channel throughput of a single transmission area for some value of l is then:

$$S(l) = \frac{\overline{U}}{\overline{B} + \overline{I}}$$

To calculate the channel throughput, we need to average the $S(l)$ over all the values of l by integration.

We multiply it by M_c to obtain the network utilization of the coverage area:

$$\begin{aligned} S &= M_c \cdot \frac{1}{\rho\pi R^2}\int_0^R 2\pi l\rho S(l)dl \\ &= \frac{S_{PRN}}{2\sqrt{3}R'^2}\cdot\frac{2}{R^2}\int_0^R lS(l)dl \end{aligned} \quad (1)$$

Combining previous equations into Eq. (1), we obtain the network utilization of DBTMA. Numerical integration is required to calculate the result.

B. 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 starts to receive a data 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] J. Deng and Z. J. Haas, "Dual Busy Tone Multiple Access (DBTMA): A New Medium Access Control for Packet Radio Networks," in *IEEE ICUPC'98*, Florence, Italy, October 5-9, 1998.

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

[5] F. Talucci, M. Gerla and L. Fratta, "MACA-BI (MACA By Invitation): A Receiver Oriented Access Protocol for Wireless Multiple Networks," in *PIMRC '97*, Helsinki, Finland, September 1-4, 1997.

[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.