IEEE 802.11 DCF ENHANCEMENTS FOR NOISY ENVIRONMENTS

Tamer Nadeem, Ashok Agrawala
Department of Computer Science, University of Maryland,
College Park, MD 20742, USA,
{nadeem,agrawala}@cs.umd.edu

Abstract - In this paper, we analytically study the performance of the IEEE 802.11 DCF for infrastructure networks in noisy environments. We show that using the standard binary exponential backoff (BEB) mechanism in noisy environments results in a poor throughput performance due to its inability of differentiating between the causes of unsuccessful packet transmissions and verify the analytical model using the *ns-2* simulator. We propose an enhanced BEB mechanism that enhances the IEEE 802.11 with a capability of differentiating between different types of unsuccessful transmissions. We study the proposed mechanism analytically and verify it using *ns-2* and show that the new mechanism enhances the network performance up to order of magnitudes with respect to the network error rates.

I. Introduction

The IEEE 802.11 standard [1] for wireless local-area networks (WLANs) has been widely used in most commercial WLAN products available in the market. The 802.11 networks could be organized in two different frameworks: infrastructure mode and ad hoc mode. In infrastructure mode stations communicate with each other by first going through an Access Point (AP). On the other hand, in ad hoc mode stations communicate directly with each other, without the use of an access point (AP). Most corporate wireless LANs operate in infrastructure mode because they require access to the wired LAN. The IEEE 802.11 MAC specifies two different medium access control (MAC) mechanisms in WLANs: the contention-based Distributed Coordination Function (DCF) and the polling-based Point Coordination Function (PCF). At present, only the mandatory DCF is implemented in the 802.11-compliant products.

In DCF mechanism, the binary exponential backoff (BEB) mechanism is used for resolving packet collisions that occur as the uncoordinated stations (nodes) contend for the channel. To ensure packet transmission reliability, MAC acknowledgment (ACK) frames are used to indicate the correct reception of the data packets. When a station does not receive a corresponding ACK frame, it assumes the packet has been dropped due to a collision, and invokes the BEB mechanism for retransmission. We refer to such mechanism in this paper as $naive_{BEB}$. Applying $naive_{BEB}$ mechanism in environments that suffers from errors due to the noise in the wireless channels, results in a poor throughput performance because it *always* assumes that the packet corruptions are due to collisions only.

In this paper, we analytically study the performance of the IEEE 802.11 MAC for infrastructure networks using $naive_{BEB}$ mechanism in noisy environments. We show how $naive_{BEB}$ affects the network performance due to its inability of differentiating between the causes of unsuccessful packet transmissions. We verify the analytical model using the *ns-2* simulator. Then, we propose $smart_{BEB}$ that enhances the IEEE 802.11 MAC with a capability of differentiating between different types of corruptions that cause unsuccessful transmissions; collision corruptions and noise corruptions. We study the proposed mechanism analytically and verify it using *ns-2*. We show that $smart_{BEB}$ enhances the network performance up to order of magnitudes with respect to the network error rates (noise level).

II. IEEE 802.11 DCF Background

The IEEE 802.11 DCF access method is based on the Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) principle. The CSMA/CA mechanism requires a minimum specified gap/space between contiguous frame transmissions. Before a station starts transmission, it senses the wireless medium to ensure that the medium is idle for a period of time (DIFS Distributed Inter Frame Space), else the station waits until the end of the in-progress transmission before waiting for DIFS. In order to reduce the collision probability among multiple stations accessing the medium, the station waits for a random backoff interval after the DIFS deferral and then transmits if the medium is still free.

If the packet is correctly received, the receiving host sends an ACK frame after another fixed period of time (SIFS Short Inter Frame Space) which is smaller than DIFS. After receiving an ACK frame correctly, the transmitter assumes successful delivery of the corresponding data frame. Otherwise, the packet is assumed to be dropped because of a collision corruption. In addition to such *basic* transmission mechanism, the DCF defines an optional *RTS/CTS* mechanism, which requires that the transmitter and receiver exchange short Request-To-Send (RTS) and Clear-To-Send (CTS) control frames prior to the actual data frame transmission to eliminate the hidden nodes problem.

The DCF adopts a slotted binary exponential backoff mechanism to select the random backoff interval (in unit of tSlotTime). This random number is drawn from a uniform distribution over the interval [0, CW-1], where CW is the contention window size and its initial value is aCWmin. In the case of an unsuccessful transmission, indicated by no receiving of ACK frame, CW is doubled. Once CW reaches aCWmax, it will remain at this value. After a successful transmission, the CW value is reset to aCWmin before the random backoff

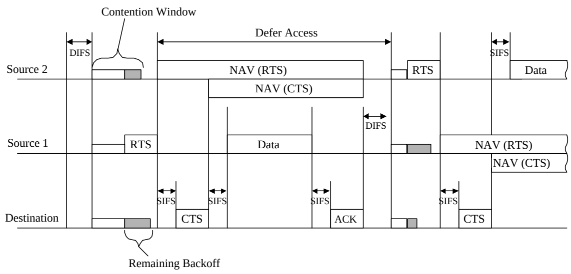

Fig. 1
IEEE 802.11 DCF Mechanism

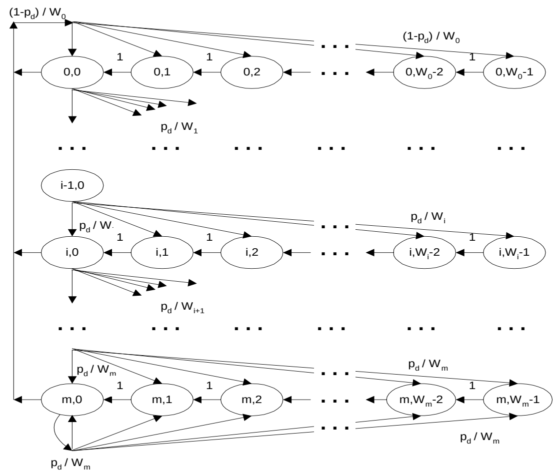

Fig. 5
Markov Chain model for the backoff window in noisy environments

interval is selected. Each station decrements its backoff counter every tSlotTime interval after the wireless medium is sensed to be idle for DIFS time. If the counter has not reached zero and the medium becomes busy again, the station freezes its counter. When the counter finally reaches zero, the station starts its transmission. Figure 1 illustrates such mechanism for case of two sources and a destination.

In noisy environment, the IEEE 802.11 binary exponential backoff mechanism that we refer to as $naive_{BEB}$ is forced to handle the noise corruptions as packet collision corruptions. Therefore, the network performance is degraded due to the additional undesired idle tSlotTime intervals because of using large values of CW.

III. Performance Analysis for IEEE 802.11 DCF in Noisy Environment

A. Markov Chain Model for Noisy Environment

Our model is based on the one proposed by [2] and we use the same assumption for our analysis. The contending stations are supposed to be a fixed number, n organized in a similar manner to the infrastructure mode. Let $b(t)$ be the stochastic process representing the back-off window size for a given station at slot time t[1]. Let m, maximum backoff stage, be the value such that $aCW_{max} = 2^m W_0$ where $W_0 = aCW_{min}$, and let us adopt the notation $W_i = 2^i W_0$, where $i \in (0, m)$ is called backoff stage. Let $s(t)$ be the stochastic process representing the backoff stage $(0, \ldots, m)$ of the station at time t. Similar to paper [2], the key approximation in this model is that the probability p_c that a transmitted packet collides is independent of the state s(t) of the station. Unlike paper [2] which used p_c to calculate the transition probabilities, we use p_d which captures the effect of the packet error rate, p_e, in the model in addition to the p_c. In the basic access mode, the transition probability because of packet corruption is:

$$p_d = p_c + (1 - p_c)\left(p_e + p_c^{ack} + (1 - p_c^{ack})p_e^{ack}\right) \tag{1}$$

and in case of using RTS/CTS access mode:

$$\begin{aligned} p_d \;&=\; p_c + (1-p_c)\Big(p_e^{rts} + p_c^{cts} + (1-p_c^{cts})\Big(p_e^{cts} + p_c^{data} \\ &\quad +(1-p_c^{data})\left(p_e + p_c^{ack} + (1-p_c^{ack})p_e^{ack}\right)\Big)\Big) \end{aligned} \tag{2}$$

where p_e^{rts}, p_e^{cts}, p_e^{ack} are the frame error probabilities (rates) of RTS, CTS, ACK respectively, while p_c^{cts}, p_c^{data}, p_c^{ack} are the colliding probabilities of CTS frame, data packet, ACK frame respectively. We can simplify such equations by neglecting the frames error probabilities because RTS, CTS, and ACK are short frames. Also, the colliding probabilities for CTS and ACK frames are negligible. Therefore, Equations 1 and 2 are approximated by:

$$p_d = p_c + p_e - p_c p_e \tag{3}$$

We model the bi-dimensional process s(t), b(t) as discrete-time Markov chain and show it in Figure 5 using p_d. The probability τ that a station transmits in a randomly chosen slot time is:

$$\begin{aligned} \tau \;&=\; \textstyle\sum_{i=0}^{m} b_{i,0} \\ &=\; \frac{2(1-2p_d)}{(1-2p_d)(W_0+1) + p_d W_0 (1-(2p_d)^m)} \end{aligned} \tag{4}$$

where $b_{i,k}$ is the stationary probability for state s(t)=i, b(t)=k, $i \in (0, m)$ and $k \in (0, W_i - 1)$. In steady state, p_d is expressed as:

$$p_d \;=\; 1 - (1-p_e)(1-\tau)^{n-1} \tag{5}$$

Equations 4 and 5 represent a nonlinear system in two unknowns τ and p_d (p_c) which can be solved using numerical techniques. A time slot will be either idle (id) where no station is transmitting, has transmission of only one station (tr) with probability of p_e of corrupting the packet, or has a collision (cl) because two or more stations are transmitting in the same time. The probabilities of such states are:

$$\begin{aligned} P_{id} &= (1-\tau)^n \\ P_{tr} &= n\tau(1-\tau)^{n-1} \\ P_{cl} &= 1 - (1-\tau)^{n-1}(1-\tau+n\tau) \end{aligned}$$

We define the saturation goodput of the network as:

$$\begin{aligned} G \;&=\; \frac{\text{E[successfully transmitted payload bytes in a slot time]}}{\text{E[length of a slot time]}} \\ &=\; \frac{(1-p_e)P_{tr}E[S]}{P_{id}\delta + (1-p_e)P_{tr}T_s + p_e P_{tr} T_f + P_{cl} T_c} \end{aligned} \tag{6}$$

[1]The slot time refers to the time interval between two consecutive backoff time counter decrements. This value is fixed (δ) in case of idle medium, or variable that includes a packet transmission when medium is busy.

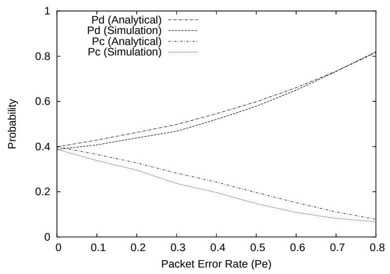

Fig. 2
The p_c and p_d values in noisy environments.

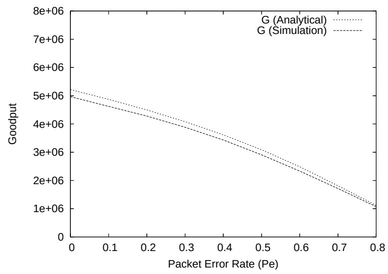

Fig. 3
Saturation goodput in noisy environments.

Parameter	*Value*	*Comments*
PHY header	24 octets	PHY layer overhead
MAC header	28 octets	MAC layer overhead
ACK	38 octets	ACK frame length + PHY header
RTS	44 octets	RTS frame length + PHY header
CTS	38 octets	CTS frame length + PHY header
Slot time	20 μs	idle slot time (δ)
SIFS	10 μs	SIFS time
DIFS	50 μs	SIFS + 2 * $delta$
aCWmin	31	minimum contention window
m	5	backoff levels

Fig. 4
MAC and PHY system parameter.

where $E[S]$ is the average packet length and δ is the duration of an empty (idle) slot time. The T_s, T_f, and T_c are the average time the channel is sensed busy because of a successful transmission, failure (corrupted) transmission, or a collision respectively. In the case of using basic access mode we have:

$$\begin{aligned} T_s &= PHY_{hdr} + MAC_{hdr} + S + SIFS + ACK + DIFS \\ T_f &= PHY_{hdr} + MAC_{hdr} + S + DIFS \\ T_c &= PHY_{hdr} + MAC_{hdr} + S + DIFS \end{aligned}$$

and in the RTS/CTS access mode we have:

$$\begin{aligned} T_s &= RTS + SIFS + CTS + SIFS + PHY_{hdr} + MAC_{hdr} \\ &\quad +S + SIFS + ACK + DIFS \\ T_f &= RTS + SIFS + CTS + SIFS + PHY_{hdr} + MAC_{hdr} \\ &\quad +S + DIFS \\ T_c &= RTS + DIFS \end{aligned}$$

where MAC_{hdr}, PHY_{hdr}, and S are the MAC layer overhead, the PHY layer overhead, and data payload respectively. Note that all terms are expressed in time units (seconds).

B. Model Validation

We validated our model by comparing the analytical results with the results from *ns-2* simulator. Each station has enough data to transmit at any time during the simulation time to obtain the saturation goodput performance. We vary the channel noise to see the effect of system under different packet error rates p_e. To simplify the analytical model, we assume all stations experience the same p_e. All the parameters used in analytical model and our simulations follow the parameters of DSSS [1], and are summarized in Figure 4. Note that PHY header, RTS frame, and CTS frame are sent at the basic access rate. Different scenarios using different number of stations, channel bit rates, payload sizes, and using both basic and RTS/CTS access modes were conducted to validate the model. In this paper we show the results for the configuration of 20 nodes in addition to the access point node to model the infrastructure mode using RTS/CTS access mode, 11 Mbps as the channel rate, and data payload is 1000 bytes in addition to IP and UDP headers of 20 and 8 bytes respectively.

Figure 2 plots the p_c and p_d values. The p_c is calculated as the number of missing CTS frames over the total number of transmitted RTS frames, and p_d as the summation of the number of missing CTS and the number of missing ACK frames over the total number of transmitted RTS frames. The saturation goodput of the network using the basic access mode is showed in Figure 3. Comparing our approximated Markov model with the simulation results for runs of different configuration scenarios, we observe that analysis results match the simulation results closely which validates our model in Section III-A.

IV. Enhanced IEEE 802.11 MAC ($smart_{BEB}$)

The problem of the current IEEE 802.11 standard mechanism is that it does not differentiate between the causes of corrupting packets. It assumes the *only* cause for dropping packets is collision corruption.

In this section, we propose the $smart_{BEB}$ which is a mechanism to enhance the IEEE 802.11 with a capability to differentiate between different causes for packet corruptions. In case a packet is dropped because of collision corruption, the IEEE 802.11 standard BEB mechanism is followed and the contention window (CW) is doubled. If the cause of dropping a packet is noise (error) corruption, $smart_{BEB}$ handles the transmission as successful one and resets the CW to W_0. In addition, $smart_{BEB}$ handles the retransmission of the dropped packet as a new packet transmission.

To model $smart_{BEB}$, we need to replace p_d of Markov model in Section III-A by $\acute{p}_d = \acute{p}_c$ where $\acute{p}_c$ is the conditional collision probability. The probability $\acute{\tau}$ in the new model is estimated by solving Equations 4 and 5, substituting p_d with $\acute{p}_d$ and τ with $\acute{\tau}$. The $\acute{P}_{id}$, $\acute{P}_{tr}$, and $\acute{P}_{cl}$ are calculated similar to the Equations 6. The goodput, $\acute{G}$ for this model is calculated using similar equation to Equation 6. We define the percentage of the goodput enhancement of $smart_{BEB}$ over $naive_{BEB}$ as:

$$\nabla G = \frac{\acute{G} - G}{G} \times 100 \quad (7)$$

Figure 6 shows analytical results of the ∇G for different configuration of data rates, number of stations, and access modes in noisy environments. Using $smart_{BEB}$ mechanism enhances the system goodput significantly because it limits the contention window size that reduces the number of unnecessary idle time slots.

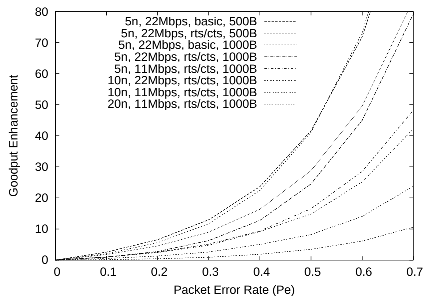

Fig. 6
Analytical goodput enhancement of $smart_{BEB}$ mechanism.

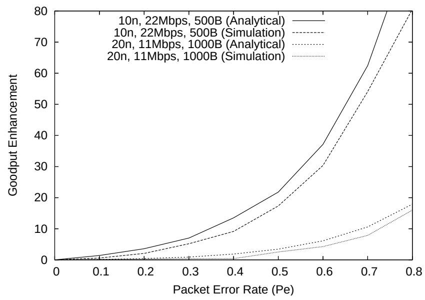

Fig. 7
Goodput enhancement for $smart_{BEB}$ implementation in RTS/CTS mode.

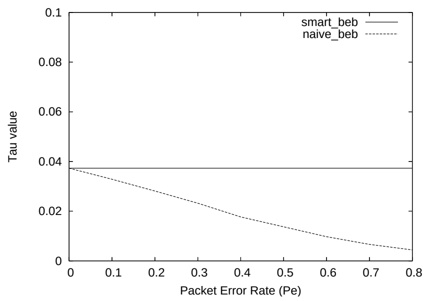

Fig. 8
The τ values in noisy environments.

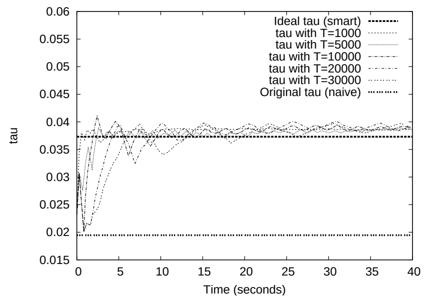

Fig. 9
Measured τ_{actual} for different T time slots when $p_e = 0.4$.

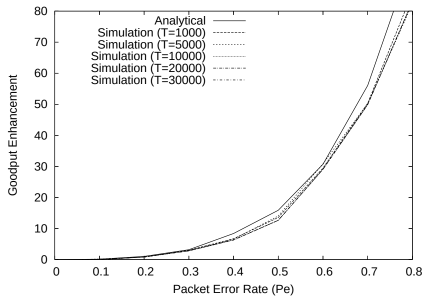

Fig. 10
Goodput enhancement for $smart_{BEB}$ implementation in basic mode.

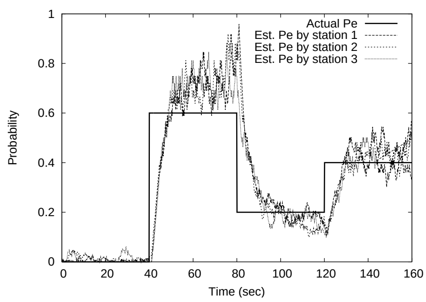

Fig. 11
The estimated p_e for variable error rates over time.

V. Implementation of $smart_{BEB}$ Mechanism

A. RTS/CTS Access Mode

In RTS/CTS mode, a station starts its transmission sequence by transmitting RTS frame. When it receives the CTS frame, it knows that the medium is reserved for its transmission. Then it transmits the data packet and waits for an ACK frame to verify a successful transmission. Since RTS and CTS are short frames, the probability of corrupting those packets due to noise errors is small and the only reason for their corruptions is because of a collision. On the other hand, once a station receives CTS, the probability of a collision corruption to the data packet is negligible. Therefore, in $smart_{BEB}$ mechanism when a station does not receive a CTS, it assumes a collision and follows the IEEE 802.11 backoff mechanism in doubling the CW size. On the other hand, when a station does not receive a ACK, it assumes the loss of the data packet due to a noise corruption and reset CW to W_0. Figure 7 shows the goodput enhancement for different configuration using the $smart_{BEB}$ mechanism. The simulation parameters are as in Table 4. From Figure 7, the simulation results match the analytical results which verifies the correctness of this implementation mechanism.

B. Basic Access Mode

In basic mode, there is no hints similar to the RTS/CTS mode to help in guessing the cause of packet corruption. Therefore, a hypothesis is needed to help identify the cause of the packet corruption in the basic access mode. The key idea of the hypothesis is that when a station doesn't receive the ACK frame, it assumes the packet is dropped because of noise corruption with probability p, or because of collision corruption with probability $(1 - p)$. Estimation of p is based on the observation from Markov model, with the knowledge of the number of active stations, that the τ value for each client is decreased with the increasing of p_e in $naive_{BEB}$ mechanism, while it is constant with different values of p_e in $smart_{BEB}$ mechanism. Figure 8 shows the τ values for scenario of 10 active stations.

We propose two methods to estimate the number of active stations: passive and active methods. In passive method, each station keeps sensing the channel and monitoring the activities on the wireless medium when it is not transmitting to count the number of different active stations. In active mode, the access point of the infrastructure network cooperates by estimating the number of active stations associated with it and broadcasting this information within the beacon frames or in a separate control messages. We summarize our mechanism mode as follows:

- Each station, initially, set its p to zero assuming all the packet losses are due to collision corruptions.
- With the knowledge of the number of active stations, each station calculates the constant goal τ (τ_{ideal}) when p_e is zero using the Markov model in Section III-A.
- Each station, during its life time, measures its actual τ value (τ_{actual}) each T time slots.
- If τ_{actual} is larger than τ_{ideal}, then the station is transmitting too frequently and needs to slow down by increasing its idle slots. Therefore, p is decreased by δ to increase the probability of collisions and subsequently increasing the CW more frequently.
- If τ_{actual} is lower than τ_{ideal}, then the station seldom tries

to transmit and needs to increase the trials by reducing the number of idle slots. Hence, the station increases p by δ to assign more of the dropping packets to noise corruptions that results in decreasing (resetting) CW more frequently.

- The δ values are assigned with respect to the value of T. For example, for $T = 1000$, we let δ be 0.01, while in case of $T = 10000$, δ is equal to 0.05.
- When an ACK frame is missing, the station resets its CW to W_0 with probability p, and increase CW to $max(2 * CW, aCWmax)$ with probability $(1 - p)$.

To validate our implementation, we ran ns-2 for different scenario configuration. In this section, we show the results for scenario of 10 active stations in addition to the access point transmitting data packets of size 500 bytes at data rate 22Mbps. We used the active method to estimate the number of active stations. Figure 9 plots the average τ_{actual} for a single station over the simulation duration for different T time slots when $p_e = 0.4$, and the goodput enhancement is plotted in Figure 10. From the figures, the effect of T is not significant. Therefore, choosing small value for T would allow $smart_{BEB}$ to adapt to the noise level faster.

Since p is the percentage of the dropped packet assigned to the noise corruptions only, p is expressed as:

$$p = \frac{(1 - p_c)p_e}{p_c + p_e - p_c p_e} \tag{8}$$

Using such equation, a station could estimate the packet error rate p_e it experiences. Figure 11 plots the estimated p_e by the first three stations for our scenario. In this simulation, p is incremented or decremented by $\delta = 0.01$ each 1000 time slots. As in the figure, the p_e estimations follow the actual p_e value as it changes over time.

VI. Related Work

One of the issues in the analysis of the IEEE 802.11 protocol is to devise an analytical model which can predict the collision probability and its effect on the performance metrics. Paper [3] analyzes the throughput and fairness issues of the DCF function and paper [4] gives the theoretical throughput limit of 802.11 based on a p-persistent variant. However, none of these captures the effect of the Contention Window(CW) and binary slotted exponential back-off procedure used by DCF in 802.11. Paper [2] uses Markov process to analyze the saturation throughput of 802.11 and show that the Markov analysis works well. The model is extended in [5] to consider the frame retransmission limits. While these studies use the stochastic analysis, TC model [6] uses the mathematical approximations with average values.

The models mentioned so far assume ideal channel conditions, where packet error does not occur. Qiao and Choi [7], [8] assume additive white Gaussian noise channel (AWGN) and calculate packet error probability, then derive the goodput performance of PHY/MAC protocol analytically. However they assume that there are only two stations (one sender and one receiver) therefore no collisions occur. In our model we consider both packet errors and the collisions among stations. To our knowledge, neither of the previous works addressed the effect of environment noises of the network performance, nor the fairness between stations suffering from different noise values.

VII. Conclusion and Future Work

In this paper, we analyzed the network performance in noisy environments. We showed how the standard BEB of IEEE 802.11 degrades significantly the network performance in such environments analytically and by simulation. We proposed an enhanced BEB, $smart_{BEB}$, that enhances the network performance by order of magnitudes in noisy environments. We showed how to implement the $smart_{BEB}$ in basic access mode and in the RTS/CTS access mode with minimal modification requirement to the IEEE 802.11.

As future work, we will study the effect of the noise on the network fairness and how $smart_{BEB}$ will guarantee the network fairness. We will examine different fairness criteria based on pricing and performance models that allow the system to choose the optimum model.

VIII. Acknowledgment

This work was supported in part by the Maryland Information and Network Dynamics (MIND) Laboratory, its founding partner Fujitsu Laboratories of America, and by the Department of Defense through a University of Maryland Institute for Advanced Computer Studies (UMIACS) contract.

References

[1] "IEEE 802.11b, Part ll: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer Extension in the 2.4GHz Band," *IEEE 802.11 Standard*, Sep. 1999.

[2] G. Bianchi, "Performance Anaylsis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal on selected areas in communications*, vol. 18, March 2000.

[3] H. Chhaya and S. Gupta, "Performance modeling of asynchronous data transfer methods of IEEE 802.11 MAC protocol," *Wireless Networks*, vol. 3, 1997.

[4] F. Cali, M. Conti, and E. Gregori, "Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit," *IEEE/ACM Transactions on Networking*, vol. 8, December 2000.

[5] H. Wu, Y. Peng, K. Long, S. Cheng, and J. Ma, "Performance of Reliable Transport Protocol over IEEE 802.11 Wireless LAN: Analysis and Enhancement," in *Proc. of IEEE INFOCOM*, (Nevada, USA), June 2002.

[6] Y. Tay and K. Chua, "A Capacity analysis for the IEEE 802.11 MAC protocol," *Wireless Networks*, pp. 159–171, January 2001.

[7] D. Qiao and S. Choi, "Goodput enhancement of IEEE 802.11a wireless LAN via link adaptation," in *Proc. of IEEE International Conference on Communications*, 2001.

[8] D. Qiao, S. Choi, and K. G. Shin, "Goodput Analysis and Link Adaptation for IEEE 802.11a Wireless LANs," *IEEE Transaction On Mobile Computing*, vol. 1, no. 4, 2002.