A Backoff Algorithm for Improving Saturation Throughput in IEEE 802.11 DCF

Kiyoshi Takahashi and Toshinori Tsuboi
School of Computer Science,
Tokyo University of Technology,
1404-1 Katakura, Hachioji, Tokyo, 192-0982 Japan
Telephone: +81-426-37-2501
E-mail: takahasi@cs.teu.ac.jp, tsuboi@cs.teu.ac.jp

***Abstract*— The medium access control (MAC) protocol is the main element which determines the system throughput in wireless local area networks. The MAC technique of the IEEE 802.11 protocol is called Distributed Coordination Function (DCF). However, in the DCF access procedure, the system throughput decreases when the number of stations is large. This paper proposes a simple backoff algorithm that uses finish tags in order to improve system throughput of a IEEE 802.11 network. Each station maintains the finish tag. It is updated when a packet reaches the front of its flow, and it is attached to the packet just prior to transmission. When a station hears packets which have older finish tags than the station, the station increases its backoff time counter. Therefore, the backoff time counter depends on the number of active stations in a wireless network; the more stations there are, the larger the backoff time counter becomes. This improves the throughput when a large number of stations are in a wireless network.**

I. Introduction

Wireless Local Area Networks (WLANs) have gained widespread acceptance. The IEEE 802.11 standard [1] is one of the recommended international standards for WLANs. The standard contains technical details for the Medium Access Control layer (MAC) and Physical layer (PHY) of the communication protocol. In the IEEE 802.11 MAC/PHY standard, two coordination functions are defined. One is the Point Coordination Function (PCF) and the other is the Distributed Coordination Function (DCF). In PCF, a polling technique is employed by the access points to query stations for any traffic they may have to send. In DCF, stations contend for the use of the channel in distributed manner via the use of the Carrier Sensing Multiple Access with Collision Avoidance (CSMA/CA) protocol. In the CSMA/CA protocol, each station sets up a backoff time according to a randomly selected interval from zero to the Contention Window (CW) for an additional deferral time before transmitting. The random backoff time is used to randomize moments at which stations try to access the wireless medium. However, in the random backoff mechanism, packet collisions are not completely eliminated, so the system throughput decreases as the number of active stations increases [2], [3], [4], [5].

To improve system throughput, several methods have been proposed. Bianchi *et al*. [6] proposed a method to determine the contention window based on the number of active stations. The number of stations is estimated by the number of slot times observed to be busy due to other stations' transmission. Cali *et al*. [7] proposed a method that estimates the number of stations via the number of empty slots and tunes CW. Yong *et al*. [8] proposed a method to determine CW by making the ratio between the idle period and period of collision be the optimum in terms of throughput. Those methods, however, need complex calculations and run-time estimations. By modifying the backoff mechanism, Haitao *et al*. [9] proposed a simple method that makes the value of CW to oscillate around a certain value. Simulations show an improvement in throughput, but saturation throughput still decreases as the number of station increases.

In this paper, we propose a simple backoff algorithm that uses finish tags. The finish tags have been proposed for fair queuing at the nodes in integrated services networks [10] and for fair queuing at mobile stations in a wireless network [11]. We modified finish tags, and use them to make the backoff time counter depend on the number of active stations. The proposed algorithm basically follows the IEEE 802.11 DCF random backoff algorithm. The main difference is that when a station hears a packet whose finish tag is older than that of the station, the station increases its backoff time counter by a constant amount. For this reason, the more stations there are, the larger the backoff time counter becomes. We show that the average length of the backoff time counter in the proposed algorithm is proportional to the number of active stations; simulations confirm that the proposed algorithm offers much higher system throughput than the current approach.

The rest of this paper is organized as follows. Section II describes the IEEE 802.11 DCF access procedure. Section III derives the proposed algorithm. Finally, we show the simulation results of the proposed algorithm in Section IV. Section V concludes this paper.

II. IEEE 802.11 DCF

In this section, we briefly show the DCF access procedure, its system throughput, and the optimum fixed contention window which has been given in [4].

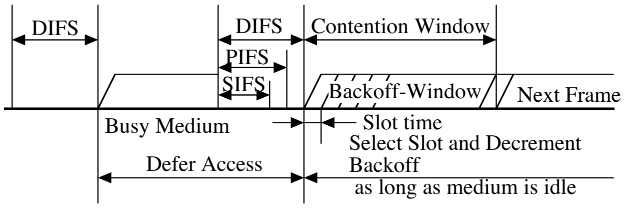

Fig. 1. Some IFS Relationships.

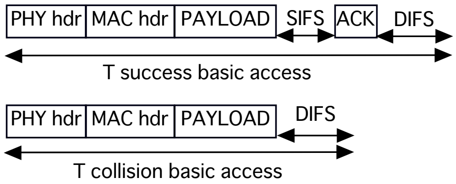

Fig. 2. *Ts* and *Tc* for basic access mechanism.

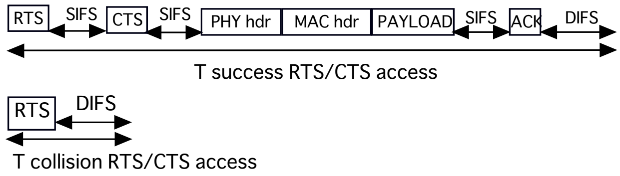

Fig. 3. *Ts* and *Tc* for RTS/CTS mechanism.

A. IEEE 802.11 DCF Access Procedure

In the IEEE 802.11DCF access procedure, each station maintains a CW and a backoff time counter. Initially, CW takes a value of CW_{min}, which is called the minimum contention window. At each transmission, the stations generate a random backoff period

$$Backoff\ Time = Random(\mathrm{CW}) \times aSlotTime, \tag{1}$$

where $Random(\mathrm{CW})$ is a pseudo-random integer drawn from a uniform distribution over the interval $[0, \mathrm{CW}]$, and $aSlotTime$ is the value of the correspondingly named PHY characteristic in the IEEE 802.11 standard [1]. This random backoff mechanism is used to randomize moments at which stations try to access the wireless medium. The set of CW values is sequentially ascending integer powers of 2 and minus 1 after each retransmission due to a collision, which occurs when two or more stations start transmission simultaneously in the same slot, until the number of retransmissions reaches the maximum backoff stage. Once the CW reaches the maximum value, it will remain at that value. After every successful transmission, the CW is reset to the CW_{min}. This contention resolution technique in the IEEE 802.11 standard is called binary exponential backoff.

A station which has a new packet monitors the channel activity for the specified interval shown in Fig.1. The station ensures that the channel is idle for the specified interval of DIFS before attempting to transmit a packet, and then begins to decrement its backoff time counter. When the backoff time counter expires and the channel is still idle, the station begins to transmit. If the channel becomes busy before the backoff time counter expires zero, the backoff time counter is frozen while the channel is busy.

The IEEE 802.11 DCF access procedure describes two alternatives for packet transmission: a basic access mechanism and an RTS/CTS mechanism. In the basic access mechanism, when backoff time counter reaches zero, the station initiates a packet transmission. After the destination station receives a packet correctly, an immediate positive acknowledgment (ACK) is sent to confirm the successful reception of the packet after a period of time called the short interframe space (SIFS). In the RTS/CTS mechanism, the station transmits a Request-to-Send (RTS) frame for the intended destination station. On receipt of the RTS frame, the destination station sends a Clear-to-Send (CTS) frame. On receipt of the CTS frame, the station transmits the data packet. On receipt of the data packet, the destination station sends an acknowledgment (ACK). The station can detect packet collision by the lack of only CTS response.

B. System Throughput with Random Backoff Mechanism

The normalized system throughput, S, is given in [4]. It is defined as the fraction of time the channel is used to successfully transmit payload bits:

$$S = \frac{P_{tr} P_s E[P]}{(1 - P_{tr})\sigma + P_{tr} P_s T_s + P_{tr}(1 - P_s) T_c}, \tag{2}$$

where $P_{tr} = 1 - (1 - \tau)^n$, $P_s = n\tau(1 - \tau)^{n-1}/P_{tr}$, n is the number of stations, $E[P]$ is the average packet payload size, σ is the duration of an empty slot time, τ is the probability that a station transmits in a randomly chosen slot time, T_s is the average time that the channel is busy because of a successful transmission, and T_c is the average time that the channel is busy because of a collision. With the basic access mechanism, see Fig.2, these values are given as

$$\begin{cases} T_s^{bas} = & \mathrm{PHY_{hdr}} + \mathrm{MAC_{hdr}} + E[P] \\ & + \mathrm{SIFS} + \delta + \mathrm{ACK} + \mathrm{DIFS} + \delta \\ T_c^{bas} = & \mathrm{PHY_{hdr}} + \mathrm{MAC_{hdr}} + E[P] \\ & + \mathrm{DIFS} + \delta, \end{cases} \tag{3}$$

where δ is the propagation delay. The parameters, except for $E[P]$, are specified for the PHY layer. With the RTS/CTS mechanism, the corresponding values are

$$\begin{cases} T_s^{rts} = & \mathrm{RTS} + \mathrm{SIFS} + \delta + \mathrm{CTS} + \mathrm{SIFS} \\ & + \delta + \mathrm{PHY_{hdr}} + \mathrm{MAC_{hdr}} + E[P] \\ & + \mathrm{SIFS} + \delta + \mathrm{ACK} + \mathrm{DIFS} + \delta \\ T_c^{rts} = & \mathrm{RTS} + \mathrm{DIFS} + \delta, \end{cases} \tag{4}$$

where RTS and CTS are the length of the RTS and CTS frames, respectively, as shown in Fig.3.

C. Optimum Fixed Contention Window and Average Length of Backoff Period

The optimum fixed contention window, CW_{opt}, is given in [4]. It maximizes the saturation throughput, which is the system throughput assuming that each station always has a packet available for transmission, when the maximum backoff stage, m, is zero. To fine the CW_{opt}, (2) is rearranged as follows:

$$S = \frac{E[P]}{T_s - T_c + \dfrac{\sigma(1-P_{tr})/P_{tr} + T_c}{P_s}}. \tag{5}$$

S is maximized when the following quantity is maximized:

$$\frac{P_s}{(1-P_{tr})/P_{tr} + T_c/\sigma} = \frac{n\tau(1-\tau)^{n-1}}{T_c^* - (1-\tau)^n(T_c^*-1)}, \tag{6}$$

where $T_c^* = T_c/\sigma$. Taking the derivative of (6) with respect to τ, rewriting it under the condition $\tau \ll 1$, and forcing it to equal 0, the approximate solution is $\tau \approx 1/(n\sqrt{T_c^*/2})$, and the optimum fixed contention window is $CW_{opt} \approx n\sqrt{2T_c^*}$. Moreover, in [4], the maximum saturation throughput, which is the saturation throughput with CW_{opt}, is shown to be independent of the number of stations.

The average length of the backoff period, in this case, $Backoff\ Time_{av} = (CW_{opt}/2) \times aSlotTime$, because the backoff period is a pseudo-random integer drawn from a uniform distribution over the interval $[0, CW_{opt}]$. Thus, since $CW_{opt} \approx n\sqrt{2T_c^*}$, it becomes

$$Backoff\ Time_{av} \approx \left(n\sqrt{T_c^*/2}\right) \times aSlotTime. \tag{7}$$

It depends on the number of stations.

III. Proposed Backoff Algorithm

In this section, we propose a new backoff algorithm and show that the average length of the backoff time counter is proportional to the number of stations.

A. Computation of Backoff Time Counter

Each station maintains a finish tag and a local virtual clock in addition to the backoff time counter and CW, and transmits its packet with a copy of the finish tag. Let (F, d) and v denote the finish tag and virtual clock at a given station. At the beginning of the transmission of a flow, the finish tag and virtual clock are set to $(0, 0)$ and zero, respectively.

When a packet reaches the front of its flow, the finish tag is calculated as follows:[1]

$$(F, d) := (L + v, 0), \tag{8}$$

where L is the length of the packet.

The backoff time counter is initialized with the binary exponential backoff scheme using CW. Let CW_j denote the CW after the successful transmission of a packet $(j = 0)$ or at the *j*-th retransmission due to a collision $(j \geq 1)$. CW_j is calculated as follows:

$$CW_j := \begin{cases} (CW_{min}+1)\times 2^j - 1 & \text{if } m > j \geq 0 \\ (CW_{min}+1)\times 2^m - 1 & \text{if } j \geq m, \end{cases} \tag{9}$$

[1]In this paper, we use := as the *assignment operator*. The expression on the right side of the assignment operator is evaluated using the current value of any variables referenced in the expression.

where *m* is the maximum backoff stage. The backoff time counter is determined with CW_j. Let BT denote the backoff time counter. After a successful transmission or at the *j*-th retransmission, it is initialized as follows:

$$BT_j := Random\,(CW_j) \times aSlotTime. \tag{10}$$

As in IEEE 802.11 DCF, the backoff time counter in the proposed backoff algorithm decreases while the channel is idle, and it is frozen while the channel is busy. If BT_j expires and the channel is still idle, the station begins to transmit. However, BT_j is increased when the given station hears a packet with an older finish tag. Upon hearing a packet with finish tag ($\acute{F}$, $\acute{d}$) from one of the other stations, even if it is not addressed to the station, own finish tag and virtual clock are updated as follows:

$$(F, d) \;:=\; (F, d+1) \tag{11}$$

$$v \;:=\; max\left[v, \acute{F}\right], \tag{12}$$

and BT_j is updated as follows:

$$\begin{aligned} &\text{if } ((F > \acute{F}) \text{ or } (F = \acute{F} \text{ and } d < \acute{d})) \\ &\qquad BT_j := BT_j + B \times aSlotTime \\ &\text{else} \\ &\qquad BT_j := BT_j, \end{aligned} \tag{13}$$

where B is a constant. This means that BT_j is increased, when ($\acute{F}$, $\acute{d}$) is older than (F, d).

When the station, which has (F, d), transmits a packet successfully, its virtual clock is calculated as follows:

$$v := max\,[v, F]\,. \tag{14}$$

The finish tag is set by using (8) again when the next packet reaches the front of its flow, and CW_0 and BT_0 are determined by using (9) and (10) again.

In our algorithm, a station increases its backoff time counter by a constant amount when it hears a packet with an older finish tag. For this reason, the more stations there are, the larger the backoff time counter becomes; system throughput improves even when the number of stations is large.

B. Average Backoff Period with Proposed Algorithm

The backoff time counter increases as many times as the station hears packets which have an older finish tag until the station transmits its own packet. Therefore, the average length of the backoff time counter is proportional to the average number of those packets.

We assume the following:

1) The length of the packets is the same.
2) Collision is the only cause of packet loss.
3) No hidden stations are considered.
4) The network consists of a finite number of stations, n.
5) Each station always immediately has a packet available for transmission (saturation conditions).

From these assumptions, after one station completes a successful transmission, all other stations have older finish tags. After hearing *i*-1 packets with older finish tags, *n*-*i* stations have an older finish tag than the station. Assuming that each station has equal probability of transmission success, the probability is equal to $1/n$; the probability that a station hears at least i packets with older finish tags from its last successful transmission until its next attempt to transmit a packet is given as follows:

$$p_i = \frac{n-1}{n}\sum_{l=0}^{\infty}\left(\frac{1}{n}\right)^l \cdots \frac{n-i}{n}\sum_{l=0}^{\infty}\left(\frac{i}{n}\right)^l \frac{1}{n} = \frac{1}{n}. \quad (15)$$

From (10) and (13), the average of the backoff time counter of the station at $j = 0$ is given as follows:

$$\begin{aligned} BT_{av} &= \left(B\sum_{i=1}^{n-1} i \times p_i + \frac{CW_{min}}{2}\right) \times aSlotTime \\ &= \left(\frac{(n-1)B}{2} + \frac{CW_{min}}{2}\right) \times aSlotTime. \quad (16) \end{aligned}$$

Under the condition $n \gg 1$,

$$BT_{av} \approx \left(n\frac{B}{2} + \frac{CW_{min}}{2}\right) \times aSlotTime. \quad (17)$$

Thus, the average of the backoff time counter, BT_{av}, depends on the number of stations[2], as is true in IEEE 802.11 DCF with the optimum fixed contention window in (7). However, in the proposed algorithm, the contention window size is not fixed (i.e., $m > 0$). BT_{av} depends on the values of CW_{min} and CW_j as well as B.

In the following section, we show their impact on saturation throughput.

IV. Simulation Results

The system throughput of the proposed algorithm depends on the values of parameters such as CW_{min} and B. In this section, we show the characteristics of the proposed algorithm in terms of these parameters.

A. Simulation Program

In our simulation program, which is written in the Java programming language, we simulate the backoff mechanism at each station and count the number of slot times the system sees the idle state, successful transmission, and collisions up to the given number of slot times; the assumptions are those of III-B.

B. Saturation Throughput

We computed the values of P_{tr} and P_s from the results of the simulation program, and system throughput using (2) and the values of T_s and T_c, which were calculated from Table I. For simplicity, we neglected the size of the finish tag, which should be included in the header or payload of a packet.

[2]After the station has heard k packets with an older finish tag, the probability that the station hears i more packets with an older finish tag until its own transmission is $p_i^k = \frac{n-k-1}{n}\sum_{l=0}^{\infty}\left(\frac{k+1}{n}\right)^l \cdot \frac{n-k-2}{n}\sum_{l=0}^{\infty}\left(\frac{k+2}{n}\right)^l \cdots \frac{n-k-i}{n}\sum_{l=0}^{\infty}\left(\frac{k+i}{n}\right)^l \cdot \frac{1}{n} = \frac{1}{n}$. Thus, after the j-th retransmission, the station has $BT_{av} = \left(\left(\frac{1}{2} - k + \frac{k}{2n} + \frac{k^2}{2n} + \frac{n}{2}\right)B + \frac{CW_j}{2}\right) \times aSlotTime$. Under the condition $n \gg 1$, $BT_{av} \approx \left(n\frac{B}{2} + \frac{CW_j - 2kB}{2}\right) \times aSlotTime$. That is, BT_{av} still depends on the number of stations.

TABLE I
System parameters for MAC and DSSS PHY Layer

Packet payload	$8191 bits$
MAC header	$272 bits$
PHY header	$192 bits$
ACK	$112 bits$ + PHY header
RTS	$160 bits$ + PHY header
CTS	$112 bits$ + PHY header
Channel bit rate	$1 Mbps$
Propagation delay	$1 \mu s$
aSlotTime	$20 \mu s$
SIFS	$10 \mu s$
DIFS	$50 \mu s$

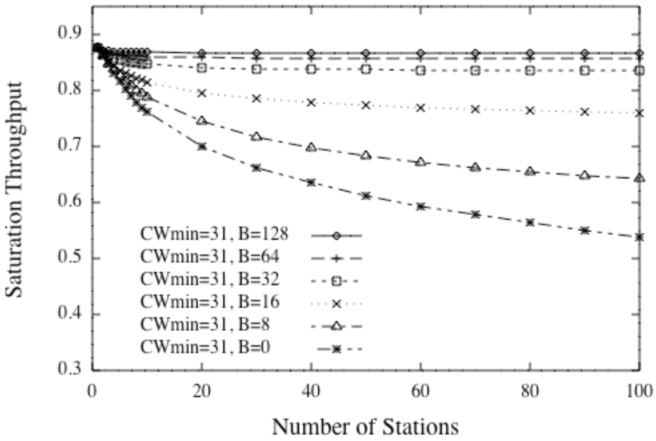

Fig. 4. Saturation throughput with basic access mechanism.

1) Basic Access Mechanism: Figure 4 plots the saturation throughput of the proposed algorithm with the basic access mechanism. CW_{min} was 31 and m was 5. B was 0, 8, 16, 32, 64, and 128. The proposed algorithm with $B = 0$ is the same as the IEEE 802.11 DCF access procedure. The saturation throughput increases as B increases, and the proposed algorithm offers a significant improvement over IEEE 802.11 DCF (Fig.4). When B is 32 and the number of stations is 30 or more, the throughput of the proposed algorithm is practically independent of the number of stations.

We also plotted the saturation throughput when CW_{min} was 7, 15, and 31 in Fig.5. It shows that the value of CW_{min} impacts on the throughput when the value of B is 8. However, when B is 32, the value of CW_{min} has a negligible impact on throughput.

2) RTS/CTS Mechanism: We plotted the saturation throughput with the RTS/CTS mechanism (Fig.6). CW_{min} was 31 and m was 5. Figure 6 shows that the throughput is practically independent of the number of stations when B is 32 and the number of stations is 30 or more; the peak of throughput decreases as B increases. However, when B exceeds 32, the entire saturation throughput curve falls.

We plot the throughput in Fig.7 for CW_{min} values of 7, 15, and 31. It shows that CW_{min} also impacts on the entire saturation throughput curve, and that B influences the peak and the improvement in saturation throughput. When B is 32,

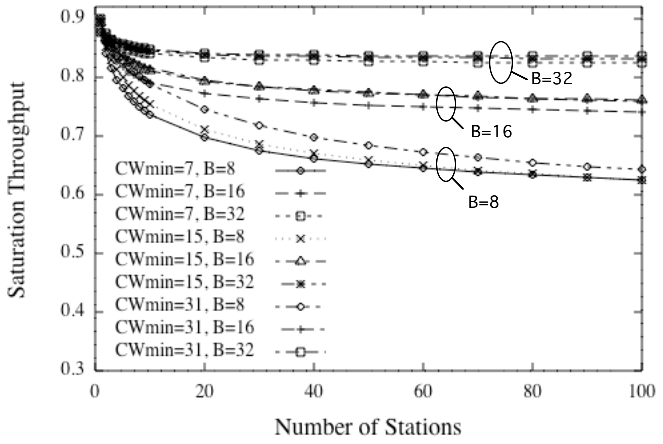

Fig. 5. Saturation throughput with basic access mechanism.

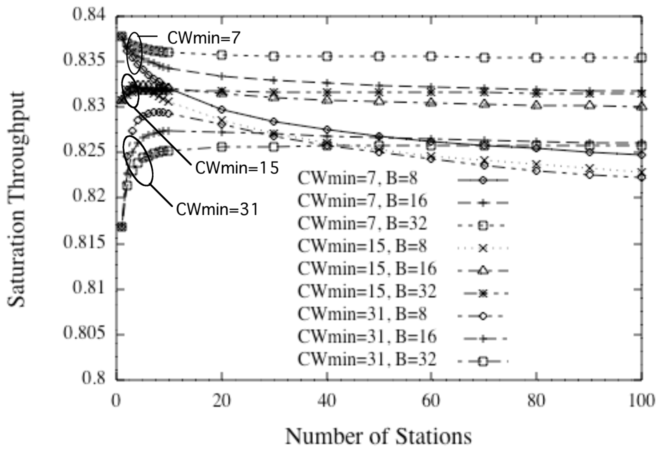

Fig. 7. Saturation throughput with RTS/CTS mechanism.

the throughput is practically independent of the number of stations regardless of the value of CW_{min}.

The difference in the throughput with the basic access mechanism and the RTS/CTS mechanism is the difference in effects of collisions on throughput. In the basic access mechanism, system throughput can be significantly improved by reducing packet collisions because that the collisions may occur on a packet. In the RTS/CTS, the performance can be improved but it is less than in the basic access mechanism because that the collisions may occur only on the RTS frame.

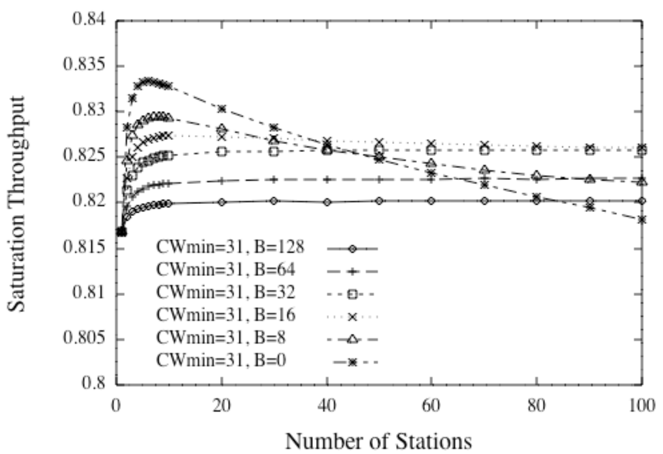

Fig. 6. Saturation throughput with RTS/CTS mechanism. The proposed algorithm with $B = 0$ is equivalent to IEEE 802.11 DCF.

V. CONCLUSIONS

In this paper, we proposed a backoff algorithm that improves the system throughput of a WLAN operating under saturation conditions. The proposed algorithm uses the binary exponential backoff scheme to initialize the backoff time counter, and finish tags to control it. Each station increases its backoff time counter by a constant amount, B, upon hearing a packet with an older finish tag from one of the other stations. The simulation results show a big improvement in terms of the saturation throughput over the IEEE 802.11 DCF access procedure when its parameters are optimized. In the basic access mechanism, CW_{min} has a small impact on the saturation throughput, but it increases as B increases. In the RTS/CTS mechanism, CW_{min} has an impact on the entire saturation throughput curve, and B has an impact on the peak and throughput. In both access mechanisms, when B is 32, the throughput is practically independent of the number of stations.

Future work can include evaluation of the proposed algorithm, e.g. fairness and delay, with the network simulator: *ns-2*.

REFERENCES

[1] IEEE std 802.11-1999 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.

[2] F. Cali, M. Conti, and E. Gregori, ”Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit,” IEEE/ACM Trans. Networking, vol.8, pp.785-799, Dec. 2000.

[3] G. Bianchi, ”IEEE 802.11 - Saturation Throughput Analysis,” IEEE Commun. Lett., vol.2, no.12, pp.318-320, Dec. 1998.

[4] G. Bianchi, ”Performance Analysis of the IEEE 802.11 Distributed Coordination Function,” IEEE J. Select. Areas Commun., vol.18, no.3, pp.535-547, Mar. 2000.

[5] P. Chatzimisios, V. Vitsas, and A. C. Boucouvalas, ”Throughput and delay analysis of IEEE 802.11 protocol,” in Proc. IEEE 5th International Workshop on Networked Appliances, Oct. 2002.

[6] G. Bianchi, L. Fratta, and M. Oliveri, ”Performance Evaluation and Enhancement of the CSMA/CA MAC Protocol for 802.11 Wireless LANs,” in Proc. the 17th International Symposium on Personal, Indoor, and Mobile Communication (PIMC), 1996.

[7] F. Cali, M. Conti, and E. Gregori, ”IEEE 802.11 Protocol: Design and Performance Evaluation of an Adaptive Backoff Mechanism,” IEEE J. Select. Areas Commun., vol.18, pp.1774-1786, Sept. 2000.

[8] Yong Peng, Haitao Wu, Shiduan Cheng, and Keping Long, ”A New Self-Adapt DCF Algorithm,” in Proc. IEEE GLOBCOM, pp.78-91, 2002.

[9] Haitao Wu, Shiduan Cheng, Yong Peng, Keping Long, and Jian Ma, ”IEEE 802.11 Distributed Coordination Function (DCF): Analysis and Enhancement,” in Proc. IEEE ICC, Apr. 2002.

[10] A.K. Parekh and R.G. Gallager, ”A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-node Case,” IEEE/ACM Trans. on Networking, vol.1, pp.344-357, Jun. 1993.

[11] N.H. Vaidya, P. Bahl, and S. Gupta, ”Distributed Fair Scheduling in a Wireless LAN,” in Proc. of ACM MobiCom 2000, pp.167-178, Boston, MA USA, Aug. 2000.