# The Modified Backoff Algorithm to reduce the number of collisions in the IEEE 802.11 Networks

Jaehyun Nam Author, *Member, KIMICS*

***Abstract***—In recent years, wireless ad hoc networks have become increasingly popular in both military and civilian applications due to their capability of building networks without the need for a pre-existing infrastructure. Recently, IEEE 802.11 *Task Group e* has been working on a new mechanism, the Enhanced Distributed Coordination Function (EDCF), to enhance the performance of 802.11 DCF. However, EDCF only reduces the internal collisions within a station, and external collisions between stations remain high in ad-hoc networks. In this paper, we propose to adopt an adaptive backoff window control technique, based on a dynamic value of the initial value of the range in which the backoff is chosen, so the backoff timer is randomly chosen in the range (*InitRng*, *CW*-1). We use *ns*-2 simulation to evaluate the throughput of our scheme. Results show that the throughput is improved for our scheme compared to the original DCF due to the reduced the number of collisions.

***Index Terms***—IEEE 802.11 DCF, backoff algorithm, collision, Wireless Local Area Networks (WLANs),

## I. INTRODUCTION

In recent years, the proliferation of portable and laptop computers has led to LAN technology being required to support wireless connectivity. Besides providing for computer mobility, wireless LANs (WLANs) are easier to install and save the cost of cabling.

The design of wireless LANs has to concentrate more on bandwidth consumption than wired networks. This is because wireless networks deliver much lower bandwidth than wired networks. However, the great improvement in channel modulation techniques makes the bandwidth of wireless medium large. It becomes feasible for real-time multimedia data to be transmitted via the wireless medium. Real time applications involving voice or video transmissions over a network have stringent requirements in terms of delay, bandwidth and other QoS parameters.

The IEEE 802.11 standard specifies a single MAC and multiple PHYs modes. The b and g modes operate in the 2.4 GHz band and provide data rates up to 11Mb/s and 54Mb/s, respectively, whereas the 802.11a PHY mode supports up to 54Mb/s in the 5 GHz band [1]. Therefore, the WLAN with high speed and low cost access to the Internet is a good platform to provide real –time services.

The IEEE 802.11 medium access control (MAC) employs a mandatory contention-based channel access function called distributed coordination function (DCF) and an optional centrally controlled channel access function called point coordination function (PCF) [1]. The DCF adopts a carrier sense multiple access with collision avoidance with binary exponential backoff.

Before initiating a transmission, each station is required to sense the medium and perform a binary exponential backoff. If the medium has been sensed idle for a time interval called DCF interframe space (DIFS), the station enters a backoff procedure. A slotted backoff time is generated randomly from a contention window (CW) size. The CW specifies the range of possible backoff time. At the first transmission attempt, CW is set equal to a minimum value, $CW_{min}$. When a collision occurs during the frame transmission, the CW becomes double and is bounded by a maximal value, $CW_{max}$. It is reset to $CW_{min}$ after successful transmission. The backoff time is decremented be each slot when the medium is sensed idle for that slot. It is frozen if the medium becomes busy, and resumes after the medium has been sensed idle again for another period of DIFS. When the backoff time is decreased to zero, the station transmits its frame immediately. The generating of random backoff time can be described as (1). The *rand*() function generates a random integer in the range from minimum to maximum uniformly.

$$Backoff\ Time = rand[0, min(CW_{min} \text{ X } 2^{\#\text{ of collision}}, CW_{max})] \text{ X } Slot\ Time \quad (1)$$

In the DCF scheme, all stations compete for the resources and channel with the same priorities. The number of collisions increases with the number of stations. Throughput degradation and high delays are caused by the increasing time needed by contending stations to access the channel. Recently, IEEE 802.11 *Task Group e* (TGe) has been working on a new mechanism, the Enhanced Distributed Coordination Function (EDCF), to enhance the performance of 802.11 DCF [2]. However, EDCF only reduces the internal collisions within a station, and external collisions between stations remain high in ad-hoc networks.

In this paper, we propose to adopt an adaptive backoff window control technique, based on a dynamic value of the initial value of the range in which the backoff is chosen. In order to provide more successful packet

Manuscript received March 4, 2008; revised May 14, 2008. Jae-Hyun Nam is with the Department of IT, Silla University, Busan, 617-736, Korea (Tel: +82-51-999-5627, Fax: +82-51-999-5657,Email: jhnam@silla.ac.kr)

transmission, the backoff in our scheme is uniformly chosen in the range [*InitRng*, $CW_i$-1] instead of the range [0, $CW_i$-1]. We study the performance of an adaptive backoff window control technique through a simulation. Results show that our scheme provides a sufficiently small collision probability and offers an improvement in terms of the throughput over the IEEE 802.11 DCF.

The rest of this paper is organized as follows. Section II briefly summarizes related works. Section III describes the modified backoff algorithm in the IEEE 802.11 DCF. Section IV shows the simulation results of our scheme in comparison to the IEEE 802.11 DCF standard. Finally, Section V concludes the paper.

## II. RELATED WORKS

Many proposals addressing the DCF enhancements were discussed in the IEEE 802.11 task group e. In [3] , they summarized the features of these enhancements based on six functions: basic contention resolution approach, class differentiation, packet differentiation, averting packet aging, scheduling of competing traffic streams, and adaptation to traffic intensity.

Concerning the performance of the IEEE 802.11 DCF, it is reasonably well studied under different assumptions by a few researchers. The saturation throughput, for the finite number of stations under ideal channel conditions were studied using the analytical models based on the Markov chains in [4, 5, 6, 7]. [4] proposed a simple and accurate analytical model to compute saturation throughput. [5] improved Bianchi's model to derive a saturation delay. In [5], the busy medium condition for invoking the backoff procedure was taken into account, which was not considered in [4]. [6] presented closed form solutions for the probability of collision and the saturation throughput. [7] specifically considered the possibility of capture, the presence of hidden terminals and their effect on the throughput performance of the IEEE 802.11 MAC protocol.

An appropriate tuning of the IEEE 802.11 backoff algorithm can significantly increase the protocol capacity [8, 9]. In [8], they proposed to tune the backoff windows size on the number of active stations, this number being estimated by observing the channel status. [9] decreased the collision probability in the IEEE 802.11 network by modifying the backoff distribution to uniformly spread the channel access in a contention window. Both studies used simulative analyzes to show that significant improvements in protocol capacity can be achieved by modifying the backoff algorithm.

The Virtual DCF (VDCF) mechanism with the low implementation cost was discussed most popularly [10]. The idea of VDCF is to adjust the size of CW and IFS according to the traffic priority. The real-time traffic with a smaller setting obtains more opportunities to access the channel. Each flow has its own backoff time setting inside the station. However, the retransmission mechanism, which is also the binary exponential backoff, still cause large packet delays for real-time traffic and the situation would be worse when the channel condition is bad or the traffic load becomes heavy.

The IEEE 802.11 MAC uses DCF for media access among the participating network nodes. But, DCF alone is neither capable nor suitable for fulfilling the QoS requirements of real-time applications like voice and video. It does not provide any priority and there is no service differentiation between different flows. Generally, the proposed QoS schemes which are based on the IEEE 802.11 try to improve DCF functionality.

## III. THE MODIFIED BACKOFF ALGORITHM IN IEEE 802.11 DCF

The DCF adopts a binary exponential backoff scheme. At each packet transmission, the backoff time is uniformly chosen in the range (0, *CW*-1). The value CW is called Contention Window, and depends on the number of transmissions failed for the packet. At the first transmission attempt, $CW = CW_{min}$ called minimum contention windows. After each unsuccessful transmission, CW is doubled, up to a maximum value $CW_{max} = 2^m CW_{min}$. The values $CW_{min}$ and $CW_{max}$ reported in the standard [1] are summarized in Table 1.

Table 1 Slot Time, minimum, and maximum contention window values specified by the 802.11 standard: Frequency Hopping Spread Spectrum (FHSS), Direct Sequence Spread Spectrum (DSSS), Infrared (IR)

| PHY | Slot Time ($\sigma$) | $CW_{min}$ | $CW_{max}$ |
|---|---|---|---|
| FHSS | 50 $\mu s$ | 16 | 1024 |
| DSSS | 20 $\mu s$ | 32 | 1024 |
| IR | 8 $\mu s$ | 64 | 1024 |

The backoff time counter is decremented as long as the channel is sensed idle, "frozen" when a transmission is detected on the channel, and reactivated when the channel is sensed idle again for more than a DIFS. The station transmits when the backoff time reaches 0.

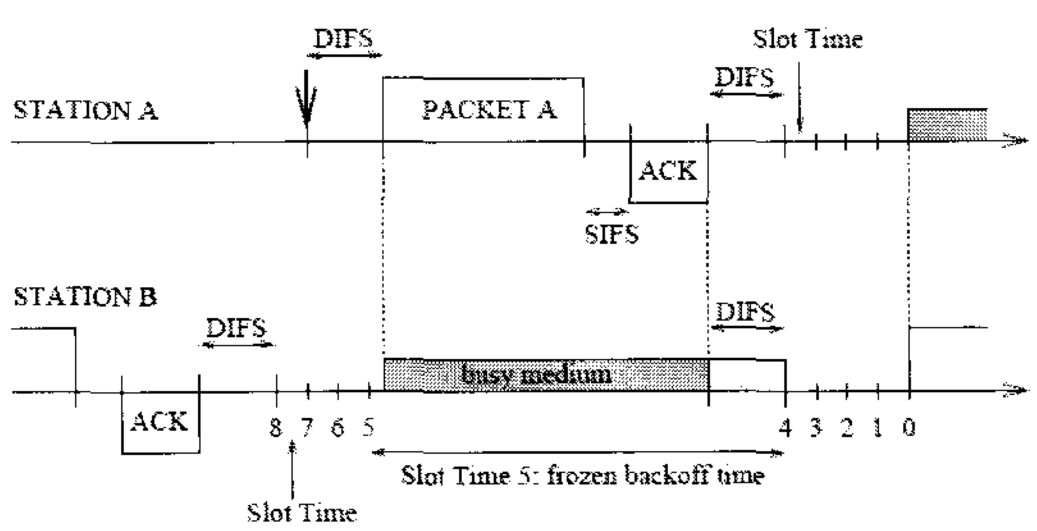



Fig. 1 Example of Basic Access Mechanism

Fig. 1 is an example of basic access mechanism in the IEEE 802.11 DCF. Two stations A and B share the same channel. At the end of the packet transmission, station B waits for a DIFS and then chooses a backoff time equal to 8, before transmitting the next packet. Note that the transmission of packet A occurs in the middle of the Slot

Time corresponding to a backoff value, for station B, equal to 5. As a consequence of the channel sensed busy, the backoff time is frozen to its value 5, and the backoff counter decrements again only when the channel is sensed idle for a DIFS.

To evaluate the performance of the DCF access scheme, [8] had measured the system throughput and the access delay. In [8], as the offered load increases, the throughput reaches a saturation value which depends on the number of contending stations: the higher the number of stations, the lower the throughput. Moreover, the saturation throughput highly depends on the number of contending stations and on the values of the contention window limits. Specially, they had observed that, regarding of the effects of the contention window limits, the most critical parameter is $CW_{min}$. In fact, especially for a large number of stations, an initially small contention window size did not provide a sufficiently small collision probability. Large values of $CW_{min}$ may strongly limit the throughput of a single station. Of course, the average access delay increases with the initial contention window size.

In the IEEE 802.11, a random value is selected from a range (0, *CW*-1) to calculate backoff time. However, it is possible to select a small backoff time regardless of a large CW. It is due to the fact that the backoff timer is randomly chosen in the range (0, *CW*-1). To address this problem, we substitute the initial value of range (0, *CW*-1) with *InitRng*, so the backoff timer is randomly chosen in the range (*InitRng*, *CW*-1). It chooses a large backoff time, and also provides a very low collision probability over the IEEE 802.11 standard. *InitRng* is calculated as follows.

$$InitRng_i = \begin{Bmatrix} 0 & i = 0,1 \\ i \times CW_{\min} & i \geq 2 \end{Bmatrix} \quad (2)$$

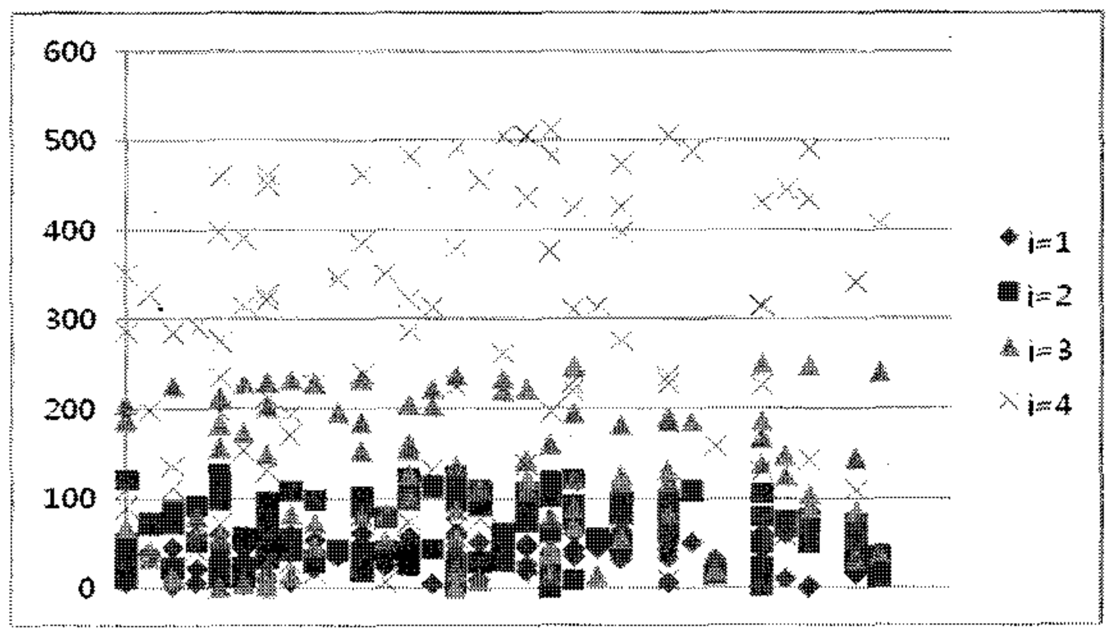
Therefore, the generating of random backoff time can be modified as (3).

$$BackoffTime = rand[InitRng, min(CW_{min} \text{ X } 2^{\# \text{ of collision}}, CW_{max})] \text{ X } SlotTime \quad (3)$$

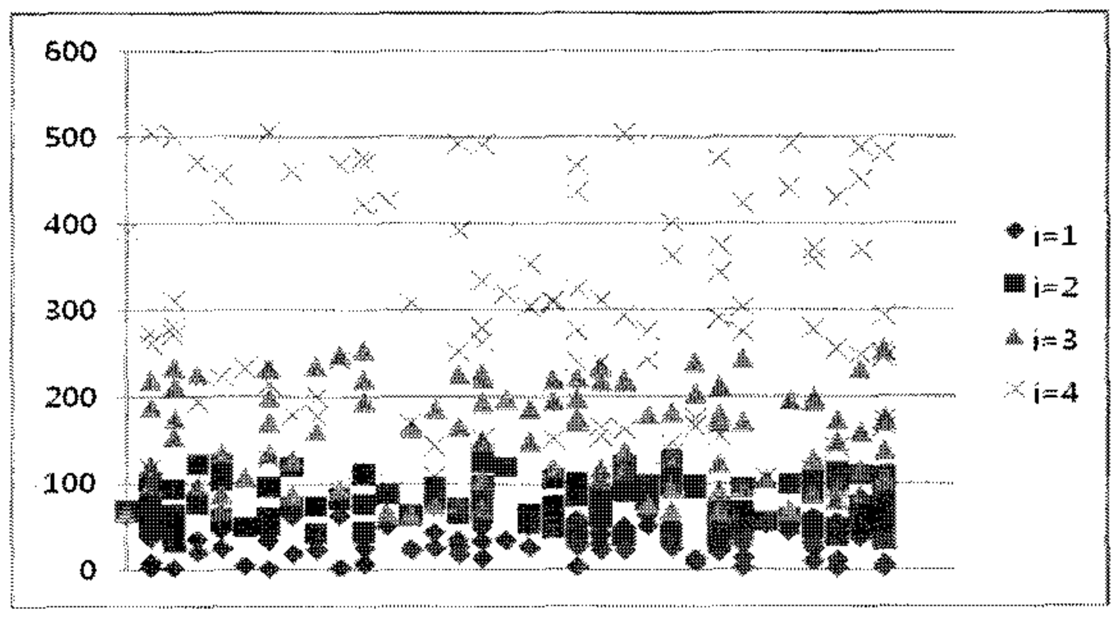
After each successful transmission, CW and *InitRng* are reset to $CW_{min}$ and 0 respectively.

$$\begin{Bmatrix} CW = CW_{\min} \\ InitRng = 0 \end{Bmatrix} \quad (4)$$

Fig. 2 represents the distribution of the backoff time, which is used in the IEEE 802.11 standard DCF and our scheme. We assume that $CW_{min}$ is 32, the number of retransmission attempts is 4 ($i$ = 4), and the number of simulations is 100 respectively. Fig. 2 (a) is the distribution of the backoff time that is used in the IEEE 802.11 standard DCF. The figure indicates that the backoff time highly depends on the number of retransmission attempts, but there is a small backoff time than $CW_{min}$ due to the random selection at the number of retransmission attempts equal to 3 or 4. However, unlike Fig. 2 (a), Fig. 2 (b) represents a large backoff time than $CW_{min}$ at the number of retransmission attempts equal to 3 or 4. This is because the backoff timer is randomly chosen in the range (*InitRng*, *CW*-1).




(a) 802.11 standard DCF




(b) Our scheme DCF

Fig. 2 Distribution of the backoff time

```
inc_cw()
{
    cw = (cw << 1) + 1;
    collisionNo = collisionNo +1;
    If (collisionNo >= 2)
        InitRng = collisionNo * phymib_.getCWMin();
    else
        InitRng = 0;
    if (cw > phymib_.getCWMax())
        cw = phymib_.getCWMax();
}
rst_cw()
{
    cw = phymib_.getCWMin();
    collisionNo = 0;
    InitRng = 0;
}
BackoffTimer::start(int cw, int InitRng, int idle, double difs)
{
    Scheduler &s = Scheduler::instance();
    stime = s.clock();
    rtime = ((Random::random() % cw) +InitRng)
            * mac->phymib_.getSlotTime();
    difs_wait = difs;
      ...............
}
```

Fig. 3 Modified backoff algorithm

Fig. 3 summarizes the modified backoff algorithm that is used in the IEEE 802.11 DCF. Fig. 3 codes modified the code of MAC/mac-802_11.h and MAC/mac-timers.cc in *ns*-2. Function *inc_cw*() doubles CW after each unsuccessful transmission and changes *InitRng* value to ($i$ * $CW_{min}$). Function *rst_cw*() resets CW and *InitRng* to $CW_{min}$ and 0 respectively.

Fig. 4 codes modified the code of MAC/mac-802_11.cc in *ns*-2. Function *recv* () is generally the entry of most network protocols (handling packets from both uplayer and downlayer). For outgoing packets, it will call *send* () function that is the entry of CSMA/CA .

```
void recv(Packet *p, Handler *h) {
    struct hdr_cmn *hdr = HDR_CMN(p);
    //handle outgoing packets
    if(hdr->direction() == hdr_cmn::DOWN) {
            send(p, h); //CSMA/CA
            return;
    }
    ...
    //else, handle incoming packets
}
void send(Packet *p, Handler *h) {
    ...
    if(mhBackoff_.busy() == 0) {
        if(is_idle()) {
            if (mhDefer_.busy() == 0) {
                /* If we are already deferring, there is no
                 * need to reset the Defer timer. */
                rTime = ((Random::random() % cw_) + InitRng)
                        * (phymib_.getSlotTime());
                mhDefer_.start(phymib_.getDIFS() + rTime);
            }
        } else {
                /* If the medium is NOT IDLE, then we start
                 * the backoff timer. */
                BackoffTimer.start(cw_, InitRng, is_idle());
                }
    }
}
```

Fig. 4 Modified CSMA/CA module in *ns*-2

## IV. SIMULATION EXPERIMENTS

In this section, we present the simulation results for the number of collisions and throughput over the number of stations. We run our simulation on the *ns*-2 simulator [11]. The DCF basic access method with modified backoff procedure and the original DCF protocol are simulated and compared with the following assumptions: ideal channel conditions, no hidden terminal effects, finite number of stations and equal average packet arrival rate for all the stations. The arrival process is assumed to be Poisson. Table 2 shows the parameter values used in our simulation.

Table 2 Simulation Parameters

| | |
|---|---|
| Packet Payload | 8184 bits |
| MAC Header | 272 bits |
| PHY Header | 128 bits |
| Channel Bit Rate | 1Mbit/s |
| Propagation Delay | 1 μs |
| Slot Time | 50 μs |
| SIFS | 28 μs |
| DIFS | 128 μs |
| ACK_Timeout | 300 μs |

Fig. 5 and 6 show the mean number of collisions per packet and the throughput versus the number of stations for five different network scenarios with 10, 20, 30, 40 and 50 stations. The formulae for the calculation of the throughput is

$$Throughput = \frac{received\_Packet \times 8000}{Runtime}$$

where *runtime* is the difference between the end time and the start time of the node.

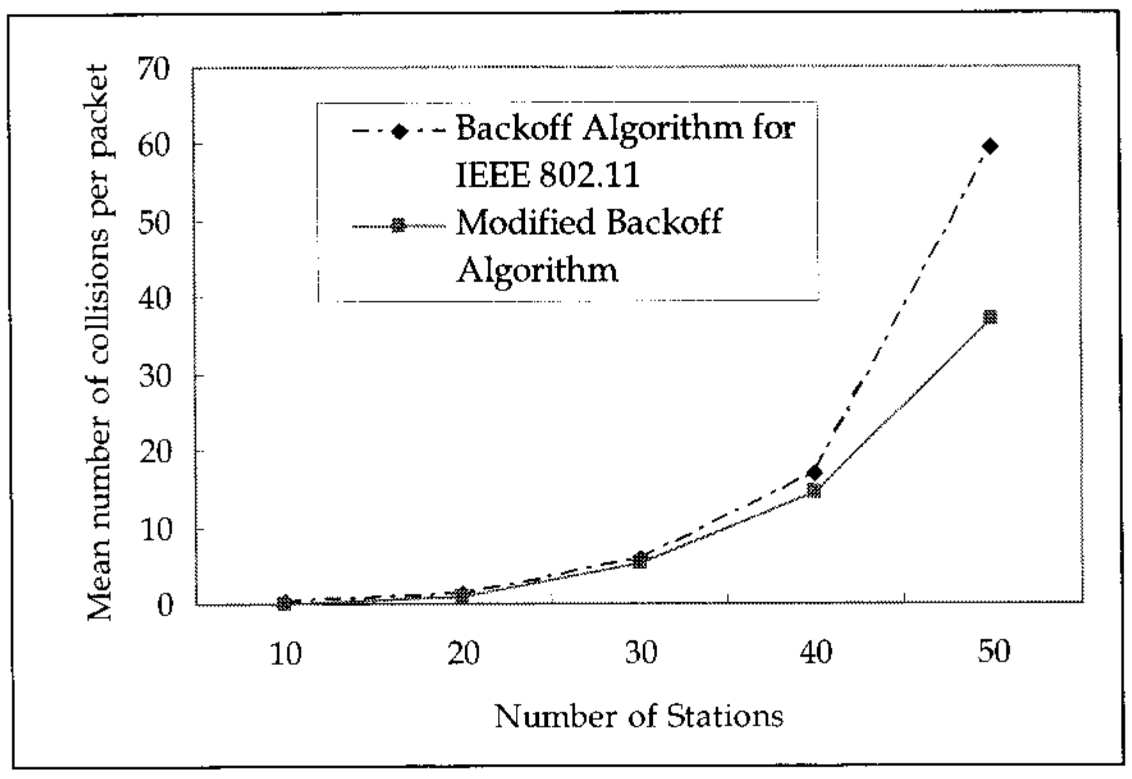



Fig. 5 Average number of collisions per packet versus the number of stations

As shown in Fig. 5, as the number of stations increases the collision probability increases. However, our scheme improves 20% collision probability more than the IEEE 802.11 DCF. The reason is that our scheme chooses a large backoff time, and also provides a very low collision probability over the IEEE 802.11 standard.

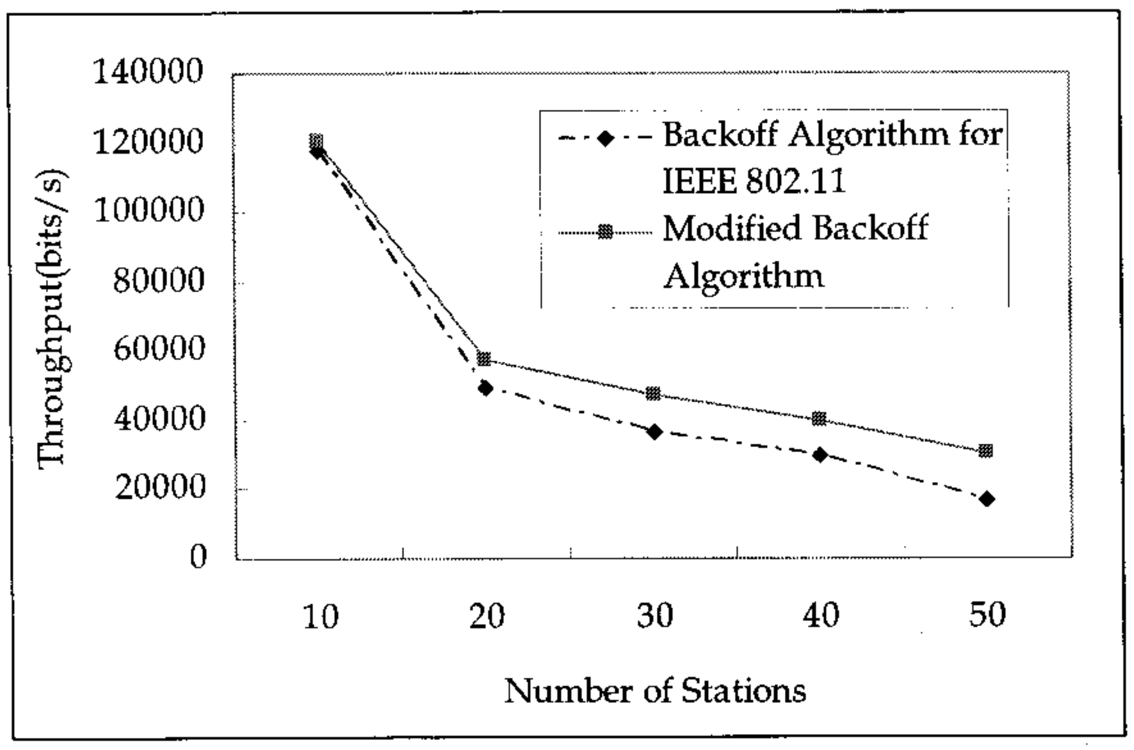



Fig. 6 Throughput versus the number of stations

Fig. 6 shows the average throughput as function of the number stations for minimum contention window size of 32. As shown in Fig. 6, as the number of stations increases the throughput decreases. However, the plot shows that the throughput is improved for our scheme compared to the original DCF due to the reduced the number of collisions. The effectiveness of our scheme is due to its capability to keep a very low collision probability.

## V. CONCLUSIONS

In this paper, we proposed the modified backoff algorithm based on that of the IEEE 802.11 DCF. In the IEEE 802.11, it is possible to select a small backoff time regardless of a large CW. It is due to the fact that the backoff timer is randomly chosen in the range (0, *CW*-1). To address this problem, we substitute the initial value of range (0, *CW*-1) with *InitRng*, so the backoff timer is randomly chosen in the range (*InitRng*, *CW*-1). It chooses a large backoff time, and also provides a very low collision probability over the IEEE 802.11 standard. We have compared the performance of our scheme with the original DCF protocol. Because the objective of our scheme is to keep a very low collision probability, it is found from the simulation results that the throughput can be substantially enhanced if the backoff timer is randomly chosen in the range (*InitRng*, *CW*-1).

However, large values of the backoff time increases the propagation delay time. We need the future work about decreasing the delay time.

## REFERENCES

[1] IEEE standard 802.11-1999, *Part II : Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) specifications*, 1999

[2] IEEE 802.11e WG, *Medium Access Control (MAC) Enhancements for Quality of Service*, IEEE 802.11e/D2.0, Nov. 2001

[3] M. Benveniste, "Summary of Features Proposed for the Enhanced-DCF Wireless MAC Protocol," *IEEE Document, 802.11-01/003*, January 2001

[4] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[5] E. Ziouva and T. Antonakopoulos, "CSMA/CA performance under high traffic conditions: Throughput and delay analysis," *Comput. Commun.*, vol. 25, no. 3, pp. 313–321, Feb. 2002.

[6] Y.C. Tay and K.C. Chua, "A capacity analysis for the IEEE 802.11 MAC Protocol", *Wireless Networks* 7, pp 159-171, 2001 Kluwer Academic Publishers.

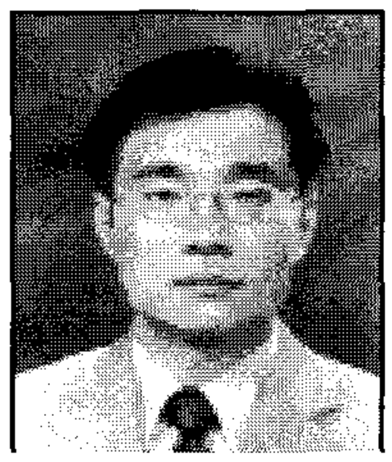
[7] H. S. Chhaya and S. Gupta, "Performance modeling of asynchronous data transfer methods of IEEE 802.11 MAC Protocol", *Wireless Networks* 3, pp 217-234, 1997.

[8] G. Bianchi, L. Fratta, and M. Oliveri, "Performance evaluation and enhancement of the CSMA/CAMAC protocol for 802.11 wireless LANs," in *Proc. PIMRC 1996*, Taipei, Taiwan, Oct. 1996, pp. 392–396.

[9] J. Weinmiller, H. Woesner, J. P. Ebert, and A. Wolisz, Analyzing and tuning the distributed coordination function in the IEEE 802.11 DFWMAC draft standard," in *Proc. MASCOT'96*, San Jose, CA, Feb. 1996.

[10] G. Chesson, A. Singla, W. Diepstraten, and H. Tenniusen, et al, "VDCF Proposed Draft Text," *IEEE Document*, 802.11-01/131, March 2001

[11] NS-2 simulator, http://www.isi.edu/nsnam/ns/, April 2002



**Jaehyun Nam**
Received the B.S., M.S., and Ph.D. degree in Computer Engineering from Pusan National University, Busan, Korea in 1989, 1992, and 2002. Since 2002, he has been an assistant professor in the Department of IT, Silla University, Busan, Korea. His research interests in the areas of wireless communications, mobile communications, and sensor networks.