Performance Evaluation of An Adaptive Backoff Scheme for WLAN

(Final version – Wiley Wireless Communications and Mobile Computing Journal – Special Issue on Emerging WLAN Technologies and Applications)

Qixiang Pang[1,2], Soung C. Liew[1], Jack Y. B. Lee[1], Victor C. M. Leung[2]

1. Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong

2. Department of Electrical and Computer Engineering, The University of British Columbia, Canada

Correspondence to:

Qixiang Pang
2356 Main Mall
Vancouver, BC, V6T 1Z4, CANADA
Email: qixiangp@ece.ubc.ca
Tel: 1-604-822-2668
Fax: 1-604-822-5949

*This work is sponsored by the Area of Excellence scheme established under the University Grant Committee of Hong Kong (Project Number AoE/E-01/99).

Performance Evaluation of An Adaptive Backoff Scheme for WLAN*

Qixiang Pang[1,2], Soung C. Liew[1], Jack Y. B. Lee[1], Victor C. M. Leung[2]

1. Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong

2. Department of Electrical and Computer Engineering, The University of British Columbia, Canada

Summary

In this paper, a simple self-adaptive contention window adjustment algorithm for 802.11 WLAN is proposed and analyzed. Numerical results show that the new algorithm outperforms the standard 802.11 window adjustment algorithm. Compared with the standard and previously proposed enhancement algorithms, a salient feature of our algorithm is that it performs well in both heavy and light contention cases regardless of the packet sizes and physical versions. Moreover, the adaptive window adjustment algorithm is simpler than previously proposed schemes in that no live measurement of the WLAN traffic activity is needed.

KEY WORDS: IEEE 802.11; WiFi; wireless local area networks; WLAN; CSMA/CA; self-adaptive contention window adjustment

* This work is sponsored by the Area of Excellence scheme established under the University Grant Committee of Hong Kong (Project Number AoE/E-01/99).

1. Introduction

The wide-spread adoption of 802.11 wireless local area networks (WLANs) operating at the unlicensed ISM frequency bands [1] is one of the few highlights of communication technologies in recent years. The IEEE 802.11 standard [2] specifies the medium access control (MAC) and physical (PHY) layers for so called 802.11 WLANs. Within the family of 802.11 WLANs, the most widely deployed version so far is 802.11b, which operates at 2.4 GHz and provides up to 11 Mbps data rate. Another standardized version is 802.11a, which operates at 5 GHz and provides up to 54 Mbps data rate. The newest version, 802.11g, which operates at 2.4 GHz as well but provides 54 Mbps data rate, has been finalized in June 2003.

Two types of MAC protocols, Distributed Coordination Function (DCF) and Point Coordination Function (PCF), are defined in 802.11. Most commercial products only implement DCF. The DCF mechanism is simple and robust. However, it has been shown by many that the standard DCF cannot efficiently utilize the limited wireless channel bandwidth when there are many stations in the WLAN accessing the same channel [3-9]. The major reason is that the initial contention window size is kept fixed regardless of the traffic activity, whereas ideally it should be large when the number of active stations is large, and vice versa.

The major contribution of this paper is a novel self-adaptive contention-window adjustment algorithm - MIMLD (Multiplicative Increase, Multiplicative/Linear Decrease) algorithm. Unlike the standard algorithm, this algorithm automatically adjusts the initial contention window to a near optimal point according to the traffic activity, thus avoiding bandwidth wastage due to improper contention window setting. Compared with other performance enhancement algorithms [3-8], our algorithm is effective not only when there are many active stations that contend with each other for channel access, but also when there are few stations. Furthermore, our algorithm does not need on-line measurement and computation.

The rest of the paper is organized as follows. Section 2 reviews the 802.11 standard and related works. Section 3 describes the proposed algorithm. Section 4 presents simulation and analytical results in various scenarios, and compares the performance of the new and standard algorithms. Section 5 concludes the paper.

2. 802.11 Standard and Related Work

DCF is the fundamental MAC layer function in 802.11 WLANs. DCF is based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). To resolve collisions of packets simultaneously transmitted by different stations, a slotted binary exponential backoff algorithm is employed in DCF.

At each packet transmission, a random backoff time (in slots) is selected uniformly between 0 and *cw*-1. The value of *cw* is called the contention window. At the first transmission attempt, *cw* is set to a value *CWmin* called the minimum contention window. After each unsuccessful transmission, *cw* is doubled, up to a maximum value *CWmax*. After a successful transmission, *cw* will be reset to *CWmin* for the next packet. The value of *CWmin* is 16 and 32, respectively, for 802.11a and 802.11b. The value of *CWmax* is 1024 in both cases[1] [2].

The contention-window adjustment algorithm defined in 802.11 has been proven to be robust in simulations as well as in real applications. However, with a fixed *CWmin*, the original algorithm neglects the possibility that the number of actively contending stations can change dynamically over time, leading to dynamically changing contention intensity. When there are many active stations, too small a *CWmin* may lead to too many collisions and backoffs; on the other hand, when there are few active stations, too high a *CWmin* may lead to excessive idle airtime during which no station attempts to transmit. In either case, the channel is not used efficiently.

To solve the problem, one could employ a dynamic contention window adjustment algorithm with contention window adjusted to reflect the number of active stations in the WLAN. These contention window adjustment algorithms are different from the 802.11

[1] In the 802.11 standard [2], 15, 31 and 1023 are actually used for *CWmin* and *CWmax*. The divergence comes from the different understanding of the value range of a new random backoff time, either [0, cw) or [0, cw]. For convenience here, we use 16, 32, and 1024 instead.

WLAN standard and have been discussed in [3-8]. Basically, these methods can be divided into two classes, measurement based and non-measurement based.

In the measurement based methods [3-6], a relationship between the number of stations and the optimal contention window is established first. Then an on-line sniffer is built inside each station to monitor the activities of all surrounding stations. When the number of active stations is estimated, the contention window is adjusted accordingly. However, on-line measurement of active stations and computation of the optimal contention window incur extra processing cost and are hard to implement. Measurement and computation errors could even lead to worse performance than the original algorithm in the 802.11 standard. To make matters worse, the optimal contention window does not only depend on the number of active stations but also on the packet sizes. In real networks, where the applications generate traffic with a wide variety of packet sizes, these measurement-based methods have their limitations.

Two non-measurement based methods were studied in [7] and [8]. These methods do not adjust the values of control parameters (e.g. *CWmin*) any more once they have been set up. They are easy to implement. But they only considered the case that there are a large number of stations. However, in real applications, it is quite common that there are only a few *active* stations with packets to send even if there are many stations in the WLAN. Web browsing, for example, will only cause a client station to transmit packets sporadically. Also in the home environment, the number of stations itself is normally quite small.

Our goal is to find a non-measurement based scheme that is simple to implement and able to provide near-optimal performance in both light and heavy contention cases. Moreover, the scheme should perform well for all versions of PHY and for the full range of packet sizes. To achieve the target, in this paper a self-adaptive contention window algorithm – *MIMLD* that emulates the TCP window adjustment mechanism is proposed.

Our algorithm is in essence the generalization of algorithms in [2,7,8]. The algorithms in [2,7,8] are all special cases of our MIMLD algorithm. Compared with those in [2,7,8], our algorithm is more flexible and provides better performance over a wider range of scenarios. As will be demonstrated, the algorithm performs well both when the number of active stations is large and when it is small. And, as with TCP, it does not require direct measurement of the traffic activity in the channel.

3. Description of MIMLD Algorithm

Figure 1 and Figure 2 demonstrate the new self-adaptive contention window adjustment algorithm for WLAN. The new algorithm is simple in principle. The major difference between the new algorithm and the original 802.11 standard is that the initial contention window in the new algorithm adapts to the contention intensity of the wireless channel. The *initial* contention window of a packet is the contention window used for its first transmission.

At any one time, the window adjustment algorithm, MIMLD, can be in one of three possible phases: multiplicative increase phase, multiplicative decrease phase, and linear decrease phase. In the new algorithm, a new control parameter called *CWbasic* is introduced and the meaning of *CWmin* is changed slightly. *CWbasic* plays the role of a threshold for distinguishing the contention intensity of the wireless channel. *CWbasic* is typically set to be close to the value of *CWmin* in the original 802.11 algorithm.

When the contention window $cw > CWbasic$, we assume that the contention intensity in the wireless channel is high. If a packet is successfully transmitted when $cw > CWbasic$, instead of going back to *CWmin* immediately, the contention window is halved but lower-bounded by *CWbasic*. By setting the contention window at a relatively high level (relative to the original 802.11 algorithm), potential collisions in the future can be minimized. This phase is called "multiplicative decrease".

When the contention window $cw \leq CWbasic$, after a successful transmission, the contention window is reduced by one rather than halved. Smaller contention window can yield better performance when there are a small number of active stations or the traffic is asymmetric (e.g., dominated by traffic from AP to clients). The intent of the linear reduction is to keep the contention window in the small regime as long as possible – reducing it too quickly may cause collisions to occur sooner, which in turn will cause *cw* to move out of this region. The linear reduction procedure is stopped when the contention window reaches its minimum value, *CWmin*. This phase is called "linear decrease".

Recall that *CWbasic* is typically set to be close to the *CWmin* value of the original 802.11 algorithm. The reason why we allow the contention window to go below *CWbasic* is based on the observation that when contention is light (e.g., when there are only one or two active stations) it is not necessary to wait an average (*CWbasic*-1)/2 time slots before each transmission attempt. However, although the contention is assumed to be light when *cw* is below *CWbasic*, this region is also regarded as critical, since the contention window is small. To be conservative and to avoid oscillations, instead of continually decreasing the window multiplicatively, it is linearly decreased. Only several consecutive successful transmissions can lower the contention window to the minimum value *CWmin*.

When a collision occurs, if $cw > CWbasic$, the contention window is doubled as in the original algorithm. If $cw < CWbasic$, the contention window is immediately increased to $CWbasic \times 2$. The reason for that is to escape the critical area below *CWbasic*, where *cw* is small, to avoid the potential for more collisions. The idea is that *cw* below *CWbasic* should be used only when the number of active stations is small. The occurrence of a collision reduces the probability of that case. This window increase phase is called "multiplicative increase".

By comparing MIMLD with the TCP congestion window adjustment procedure, similarities can be found. TCP throughput is proportional to its congestion window while 802.11 MAC throughput is proportional to the reciprocal of its contention window. The basic algorithms in TCP are Additive Increase and Multiplicative Decrease (AIMD) and "slow" start [10]. Our "multiplicative decrease" phase resembles the "slow start", the "linear decrease" phase resembles the "additive increase" in TCP, and the "multiplicative increase" phase resembles the "multiplicative decrease" algorithm in TCP (Table 1).

MIMLD is simple to implement in that it does not require on-line sniffing and measurement of traffic activity. In addition, the intrinsic operation of MIMLD takes into account both light and heavy contention cases. The next section shows that MIMLD algorithm improves the performance of the original 802.11 algorithm in both cases.

4. Performance Evaluation

In this section, the performance of MIMLD is analyzed and compared with the standard algorithm. The analytical analysis is verified by simulation using *NS-2* [11]. Table 2 shows the physical properties of 802.11a/11g and 802.11b in our study[2].

4.1. Throughput and Delay Analysis

We assume that there are n contending stations in a WLAN. The buffers of all stations are saturated and the stations always have packets to send. The analytical evaluation of the MIMLD algorithm is given as follows. For the analysis of the standard algorithm, please refer to [9].

Let p denote the probability of a packet colliding with another packet. Same as the model in [9], p is assumed to be constant and independent of the backoff stage of the packet (i.e., independent of how many collisions the packet has already suffered). Let $s(t)$ be the random process representing the backoff stage at time t. Let $b(t)$ be the random process representing the backoff counter during a particular backoff stage. Other definitions of parameters are as follows:

τ : the probability of a station transmitting during a slot time;

m : the maximum backoff stage beyond which the window size will be kept constant;

W : the basic window size, i.e., the value of *CWbasic;*

d : $-d$ is the minimum backoff stage; we have the relations $d = W - CWmin$ and $0 \le d < W$;

W_i : the window size at backoff stage i, $-d \le i \le m$;

$$W_i = \begin{cases} 2^i W & 0 \le i \le m \\ W + i & -d \le i < 0 \end{cases} \tag{1}$$

[2] While in 802.11g standard SIFS is 10 μs, in practice it is effectively 16 μs because of a 6 μs signal extension that gets tacked on to the end of every OFDM frame. Effectively, in *NS-2* simulation, not losing accuracy, a pure 802.11g environment is assigned the same parameters as that of 802.11a [12].

Consider the two-dimensional discrete-time Markov chain given by $\{s(t),\ b(t)\}$. By using the short notation, $P\{i_1,k_1 \mid i_0,k_0\} = P\{s(t+1)=i_1, b(t+1)=k_1 \mid s(t)=i_0, b(t)=k_0\}$, the only non-null one-step transition probabilities in the Markov chain are as follows (Figure 3):

$$\begin{cases} P\{i,k \mid i,k+1\} = 1 & -d \le i \le m, 0 \le k \le W_i - 2 \\ P\{i,k \mid i+1,0\} = (1-p)/W_i & -d \le i \le m-1, 0 \le k \le W_i - 1 \\ P\{i,k \mid i-1,0\} = p/W_i & 1 \le i \le m, 0 \le k \le W_i - 1 \\ P\{1,k \mid i,0\} = p/W_1 & -d \le i \le -1,\ 0 \le k \le W_1 - 1 \\ P\{m,k \mid m,0\} = p/W_m & 0 \le k \le W_m - 1 \\ P\{-d,k \mid -d,0\} = (1-p)/W_{-d} & 0 \le k \le W_{-d} - 1 \end{cases} \qquad (2)$$

The first line in (2) corresponds to the decrement of the backoff counter during the backoff process. The second line corresponds to the successful transmission of the previous packet and the new packet starts transmission. The third and fourth lines correspond to packet collision and the increment of the backoff stage. The fifth line corresponds to the fact that once the maximum stage is reached, there is no increment in subsequent transmissions. The sixth line corresponds to the fact that once the minimum stage is reached, there is no decrement in subsequent transmissions.

Let $b_{i,k} = \lim_{t\to\infty} \Pr\{s(t)=i, b(t)=k\}$ be the stationary distribution of the Markov chain. We have,

$$\begin{cases} b_{i,0} = \left(\dfrac{p}{1-p}\right)^{i-1} b_{1,0} & 1 \le i \le m \\ b_{i,0} = (1-p)^{-i+1} b_{1,0} & -d < i \le 0 \\ b_{-d,0} = \dfrac{(1-p)^{d+1}}{p} b_{1,0} & \end{cases} \qquad (3)$$

and,

$$b_{i,k} = \frac{W_i - k}{W_i} b_{i,0},\ -d \le i \le m,\ 1 \le k \le W_i - 1 \qquad (4)$$

By imposing the normalization condition for stationary probability distribution, we have,

$$1 = \sum_{i=-d}^{m} \left(b_{i,0} \sum_{k=0}^{W_i - 1} \frac{W_i - k}{W_i} \right) = \sum_{i=-d}^{m} \left(b_{i,0} \frac{W_i + 1}{2} \right)$$

$$=\frac{b_{1,0}}{2}\left[\sum_{i=1}^{m}\left(\frac{p}{1-p}\right)^{i-1}(W_i+1)+\sum_{i=-(d-1)}^{0}(1-p)^{-i+1}(W_i+1)+\frac{(1-p)^{d+1}}{p}\left(W_{-d}+1\right)\right]$$

$$=\frac{b_{1,0}}{2}\left[\left(\sum_{i=1}^{m}2W\left(\frac{2p}{1-p}\right)^{i-1}+\sum_{i=1}^{m}\left(\frac{p}{1-p}\right)^{i-1}\right)+\left(\sum_{i=-(d-1)}^{0}(W+i+1)(1-p)^{-i+1}+(W-d+1)\frac{(1-p)^{d+1}}{p}\right)\right]$$

$$=\frac{b_{1,0}}{2}\left[\Delta_1+\Delta_2\right] \tag{5}$$

Δ_1 and Δ_2 in (5) are derived as follows.

For simplicity, let $q=\frac{p}{1-p}$,

$$\Delta_1=\left(\sum_{i=1}^{m}2W\left(2q\right)^{i-1}+\sum_{i=1}^{m}q^{i-1}\right)=2W\left(\frac{1-(2q)^m}{1-2q}\right)+\frac{1-q^m}{1-q}$$

$$=\frac{2W(1-(2q)^m)(1-q)+(1-2q)(1-q^m)}{(1-2q)(1-q)} \tag{6}$$

$$\Delta_2=\sum_{i=-(d-1)}^{0}(W+i+1)(1-p)^{-i+1}+(W-d+1)\frac{(1-p)^{d+1}}{p}$$

$$=\sum_{j=0}^{d-1}(W-j+1)(1-p)^{j+1}+(W-d+1)\frac{(1-p)^{d+1}}{p}$$

$$=\sum_{j=0}^{d-1}(W+1)(1-p)^{j+1}-(1-p)^2\sum_{j=0}^{d-1}j(1-p)^{j-1}+(W-d+1)\frac{(1-p)^{d+1}}{p}$$

$$=(W+1)\sum_{j=0}^{d-1}(1-p)^{j+1}+(1-p)^2\frac{\partial}{\partial p}\left(\sum_{j=0}^{d-1}(1-p)^{j}\right)+(W-d+1)\frac{(1-p)^{d+1}}{p}$$

$$=\frac{(W+1)(1-p)(1-(1-p)^d)}{p}+\frac{(1-p)^2[dp(1-p)^{d-1}-1+(1-p)^d]}{p^2}+(W-d+1)\frac{(1-p)^{d+1}}{p}$$

$$=\frac{(1-p)(Wp+2p-1+(1-p)^{d+1})}{p^2} \tag{7}$$

Then from (5), (6) and (7), we get

$$b_{1,0}=2/\left[\Delta_1+\Delta_2\right]$$

$$=2\Big/\left[\frac{2W(1-(2q)^m)(1-q)+(1-2q)(1-q^m)}{(1-2q)(1-q)}+\frac{(1-p)(Wp+2p-1+(1-p)^{d+1})}{p^2}\right] \tag{8}$$

We can now express the probability that a station transmits in a randomly chosen slot time. As a transmission occurs when the backoff time counter is equal to zero, regardless of the backoff stage, thus,

$$\tau = \sum_{i=-d}^{m} b_{i,0} = \frac{1-q^m}{1-q} b_{1,0} + \frac{1-p}{p} b_{1,0} = \frac{1-q^m}{1-q} b_{1,0} + \frac{1}{q} b_{1,0} = \frac{1-q^{m+1}}{(1-q)q} b_{1,0} \quad (9)$$

The collision probability is then determined by,

$$p = 1-(1-\tau)^{n-1} \quad (10)$$

Note that (8), (9) and (10) represent a nonlinear system in the two unknowns τ and p, which can be solved using numerical techniques. Once τ is solved, the throughput can be derived accordingly (neglecting the over-the-air propagation delay):

$$S = \frac{P_s P_{tr} Pk}{(1-P_{tr})\sigma + P_{tr} P_s T_s + P_{tr}(1-P_s)T_c} \quad (11)$$

where,

S is the system throughput of MAC layer payload;

P_s is the probability that a transmission occurring on the channel is successful,

$$P_s = \frac{n\tau(1-\tau)^{n-1}}{1-(1-\tau)^n}; \quad (12)$$

P_{tr} is the probability that there is at least one transmission in a considered slot time,

$$P_{tr} = 1-(1-\tau)^n; \quad (13)$$

Pk is the MAC payload size (in bits);

σ is the duration of an empty slot time;

T_s is the time the channel is sensed busy because of a successful transmission;

T_c is the time wasted by a collision;

For basic access, $T_s = T_c = DIFS + DATA + SIFS + ACK$;

For RTS/CTS access, $T_s = DIFS + RTS + SIFS + CTS + SIFS + DATA + SIFS + ACK$;

$T_c = DIFS + RTS + SIFS + CTS$.

The average delay can be estimated as follows. Assume the total throughput given by (11) is fairly shared by the n contending stations, thus the average packet delay is given by,

$$D = \frac{n \cdot Pk}{S} = \frac{n \cdot \left((1 - P_{tr})\sigma + P_{tr} P_s T_s + P_{tr}(1 - P_s) T_c \right)}{P_s P_{tr}} \tag{14}$$

Numerical results are presented in Figure 4 to Figure 6. For 802.11b, the control parameters of MIMLD algorithm, *CWmin*, *CWbasic*, and *CWmax* are, 2, 32, and 1024, respectively. For 802.11a/11g, they are 2, 16, and 1024, respectively.

Figure 4 and Figure 5 present the throughput comparisons of the two algorithms. It can be concluded that MIMLD produces throughput improvements for both 802.11b and 11a/11g, whether the number of active stations is large or small, and whether the packet size is large or small. As for the packet delay, equation (14) shows that the improvement of the delay is same as that of the throughput.

In particular, when the number of stations is large (say, >10) or when it is small (say, <3), MIMLD gives significant improvements over the standard algorithm. For instance, in basic access mode, the percentages of improvements for one single station are 24% (802.11b, 1000 bytes packet size), 50% (802.11b, 100 bytes packet size), 24% (802.11a/g, 1000 bytes), and, 48% (802.11a/g, 100 bytes). In the case of 60 stations, the improvements are 14% (802.11b, 1000 bytes packet size), 14% (802.11b, 100 bytes packet size), 20% (802.11a/g, 1000 bytes), and, 18% (802.11a/g, 100 bytes).

RTS/CTS access is an optional feature in the 802.11 standard and is helpful to overcome the hidden terminal problem. Figure 6 shows that the new backoff algorithm improves the performance of RTS/CTS access as well, although the amount of improvement is not as high as that of basic access.

4.2. Self-Adaptation of the Initial Contention Window

The initial contention window of a packet is the contention window for its first transmission and it should change along with the change of the number of active stations to achieve optimal throughput. However, in the standard backoff algorithm, for every packet, the initial contention window is always set rigidly to *CWmin*, which does not take into account the current contention intensity. In order to dynamically change the initial contention window, one may come up with a measurement-based scheme. However, as discussed in Section 2, the measurement-based schemes have some drawbacks that render them impractical. Therefore a non-measurement based scheme is preferable.

Our MIMLD and the standard algorithms are both non-measurement based. They are common in that once the control parameters (*CWmin*, *CWbasic*, *CWmax* for *MIMLD* and *CWmin*, *CWmax* for the standard algorithm) are configured in the system, no further changes are necessary whatever the number of active stations is. However, different from the standard algorithm, in the MIMLD algorithm, the initial contention window of a packet transmission is changed automatically. It is not always *CWmin* as in the standard algorithm.

Table 3 compares the initial contention windows in the two algorithms when 1000-byte data packets are transmitted over 802.11b by saturated stations. For the standard algorithm, *CWmin* is 32 and *CWmax* is 1024. For the MIMLD algorithm, the values for *CWmin*, *CWbasic* and *CWmax* are 2, 32 and 1024, respectively. All of the parameters are not changed when the number of stations changes. The standard algorithm always uses *CWmin* (which is 32) as its initial contention window regardless of the number of stations. In contrast, in MIMLD algorithm, the initial contention window adjusts automatically with the change of the number of stations. It can be seen in Table 3 that the initial contention window in MIMLD is smaller than that in the standard algorithm when the number of stations is small and larger when the number of stations is large. This explains the reason why the MIMLD algorithm performs better than the standard algorithm.

Figure 7 compares the performance of MIMLD with the standard algorithm with different *CWmin* values (16, 32 and 64). It is not difficult to conclude that MIMLD performs overall the best among the four cases over a wide range of scenarios.

4.3. Fairness and Robustness of MIMLD Algorithm

One may question the fairness of our algorithm by observing that in our algorithm the linear reduction part further reduces the contention window size when the transmission of one station is successful and the window is already below the value of *CWbasic*. Same as in the standard algorithm, it benefits the successful station. By simulation, we compare the fairness of our algorithm and the standard algorithm quantitatively.

The well-known Jain's fairness index [13] is used. The definition of Jain's index is given by,

$$f(x_1, x_2, x_3, \ldots, x_n) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n\sum_{i=1}^{n} x_i^2} \tag{15}$$

where, x_i is the throughput of contending flow (station) i; and, n is the number of contending flows (stations).

Table 4 gives the measured fairness indices. In the scenarios, saturated stations sending data packets (MAC payload is 1000 bytes) are simulated for 100 seconds. The results show that there is only a slight fairness impact by the new algorithm.

We then design a simulation scenario in which the number of active stations changes dynamically to examine the robustness of the MIMLD algorithm. The purpose is to demonstrate how our algorithm adapts to the changes in contention intensity. In the scenario being studied, the number of active stations ramps up in steps from 2 to 4, 6, 8, 10, 20, 30, and finally reaches 40. Then the number decreases in steps to 30, 20, 10, 8, 6, 4, and finally back to 2. Each active station operates at the 802.11b physical layer and attempts to send data packets (1000 bytes) in a saturated manner one after another. Two step sizes, 1 sec and 50 sec, are applied.

Figure 8 shows the sampled initial contention windows. From the figure, when the number of stations is high, the initial contention window tends to be large, and when the number of stations is small, the initial contention window tends to be small. This verifies that our algorithm effectively adapts the contention window according to the contention intensity. Figure 9 shows another scenario in which the number of stations abruptly changes from 5 to 40. The contention windows of two stations are sampled and shown in Figure 9. Within seconds of the change in the number of active stations, the initial contention window stabilizes to a higher average value.

4.4. Effect of Different Control Parameters in MIMLD Algorithm

We now study the effect of assigning different control parameter values to the MIMLD algorithm. Numerical results in Figure 10 show the effects on system throughput when the control parameters in MIMLD, <*CWmin, CWbasic*>, are assigned different values. In the legend, standard-16 and standard-64 stand for the standard algorithm with *CWmin*=16 and 64, respectively. MIMLD-<2,16> stands for MIMLD algorithm with *CWmin*=2 and *CWbasic*=16. MIMLD-<2,64> stands for MIMLD algorithm with *CWmin*=2 and *CWbasic*=64. 802.11b physical layer and 1000 bytes data packets are used.

By comparing the results of MIMLD and the standard algorithm, we conclude that the MIMLD algorithm is less sensitive to the contention window settings. The throughput difference between MIMLD-<2,16> and MIMLD–<2,64> is much smaller than that between standard-16 and standard-64. The insensibility to the values of *CWmin* and *CWbasic* in MIMLD algorithm implies that there is no strong need to adjust them and the values of <2, 32> will work well.

In Figure 11, we examine the performance of different multiplicative decrease factors (*mdf*). Two different values of multiplicative decrease factor are studied, 1.5, and the original 2. In both cases, the contention windows are, *CWbasic* =32, and *CWmin* =2.

Figure 11 shows that by assigning 1.5 to the multiplicative decrease factor, further improvement is obtained when the number of stations is large. But the improvement is less significant than that from the standard algorithm to MIMLD with *mdf*=2.

We would like to leave thorough study of different parameters for future work. In addition, recently the 802.11 standardization committee is actively involved in a new MAC standard of 802.11, 802.11e [14]. In 802.11e, the values of the control parameters, such as *CWmin*, inter-frame space, transmission opportunity limit, are differentiated for different traffic categories. By applying the same principle, we can extend our new algorithm to its enhanced version – enhanced MIMLD algorithm. As shown in Figure 12, for different traffic categories, different *CWmin*, *CWbasic*, *CWmax*, multiplicative increase factor (*mif*), multiplicative increase factor (*mdf*), and linear decrease factor (*ldf*) can be applied. A study of the performance of enhanced MIMLD algorithm when multiple traffic categories with different QoS requirements co-exist in one WLAN is left for the future as well.

5. Conclusions

In this paper, we have proposed and studied a new self-adaptive contention window adjustment algorithm, MIMLD, for DCF in 802.11 WLANs. Our algorithm has the following advantages over the original and other enhancement algorithms:

- Effective and Flexible

  Compared with previously proposed enhancement algorithms, MIMLD exhibits performance improvements over the original algorithm over a full range of number of active stations. In addition, our algorithm also does not need to assume constant packet size in its optimization procedure.

- Simple to implement

  Because MIMLD does not need on-line measurements, it can be easily implemented by minor modification of the 802.11 firmware.

- Robust and Responsive

  The simulation results also show that the new algorithm is less sensitive to the initial parameter settings than the standard algorithm. Similarity exists between our MIMLD algorithm and TCP congestion window adjustment algorithm, which has been widely proven robust. Our simulations show that the MIMLD algorithm adjusts to changes in number of active stations within seconds.

In this paper, we mainly target the performance improvement of a single-hop wireless network (either infrastructure mode or ad hoc mode of WLAN), which is currently the dominant application of 802.11 [1,2]. Another potential application of 802.11 products is multi-hop wireless networks. As widely known, there are severe problems in multi-hop wireless networks [15-17], such as hidden terminal, exposed terminal, self interference, capture effect, QoS, etc. However, the consideration of these problems is outside the scope of this paper.

Acknowledgement

The authors would like to thank the three anonymous reviewers for providing valuable comments to improve the quality of this paper.

References

1. Gast MS. 802.11 Wireless Networks: The Definitive Guides, O'Reilly, 2002.
2. IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, ISO/IEC 8802-11:1999E, Aug. 1999.
3. Bianchi G, Fratta L, Oliveri M. Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs. *IEEE PIMRC* 1996; 392-396.
4. Cali F, Conti M, Gregori E. IEEE 802.11 protocol: design and performance evaluation of an adaptive backoff mechanism. *IEEE Journal on Selected Areas in Communications* 2000; 18(9): 1774-1786.
5. Peng Y, Wu H, Cheng S, Long K. A new self-adapt DCF algorithm. *IEEE GLOBECOM* 2002; 87-91.
6. Chen Y, Zeng Q, Agrawal DP. Performance analysis and enhancement for IEEE 802.11 MAC protocol. *International Conference on Telecommunications* 2003; 860-867.
7. Wu H, Cheng S, Peng Y, Long K, Ma J. IEEE 802.11 distributed coordination function (DCF): analysis and enhancement. *IEEE ICC* 2002; 605-609.
8. Song N, Kwak B, Song J, Miller LE. Enhancement of IEEE 802.11 distributed coordination function with exponential increase exponential decrease backoff algorithm. *IEEE VTC* 2003 Spring; 2775-2778.
9. Bianchi G. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications* 2000; 18(3): 535-547.
10. Stevens WR. TCP/IP illustrated, Addison-Wesley, 1994.
11. NS-2, URL http://www.isi.edu/nsnam/ns.
12. URL http://www.tkn.tu-berlin.de/research/802.11e_ns2.

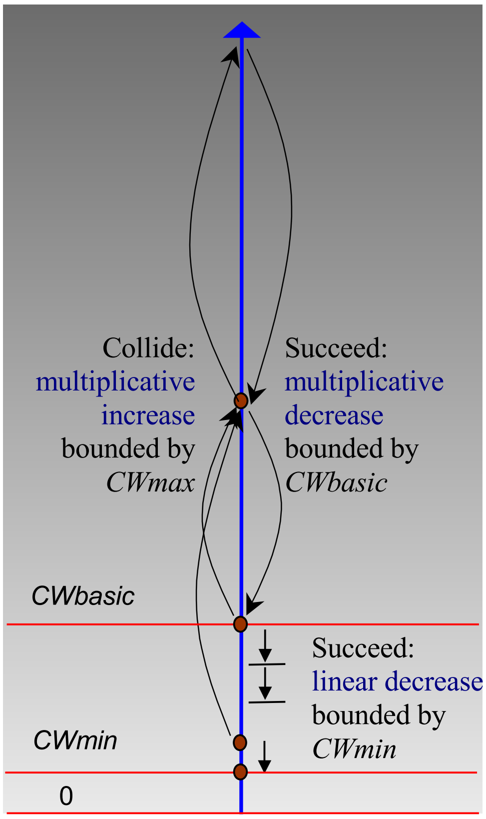
13. Jain R. Throughput fairness index: an explanation. *ATM Forum Contribution* 99-0045, February 1999.
14. IEEE WG, "802.11e D6," November 2003.
15. Issariyakul T, Hossain E, Kim DI. Medium access control protocols for wireless mobile ad hoc networks: issues and approaches. *Wireless Communications and Mobile Computing* 2003; 3(8): 935-958.
16. Acharya A, Misra A, Bansal S. High-performance architectures for IP-based multihop 802.11 networks. *IEEE Wireless Communications* 2003; 10(5): 22-28.
17. Chakrabarti S, Mishra A. Quality of service challenges for wireless mobile ad hoc networks. *Wireless Communications and Mobile Computing* 2004; 4(2): 129-153.

- *cw* $\leftarrow$ *max (cw/2, CWbasic), if (succeeds and cw > CWbasic).*
- *cw* $\leftarrow$ *max (--cw, CWmin), if (succeeds and cw* $\leq$ *CWbasic).*
- *cw* $\leftarrow$ *min {2×max (cw, CWbasic), CWmax}, if collides.*
- *cw* $\leftarrow$ *cw, if retry limit is reached.*

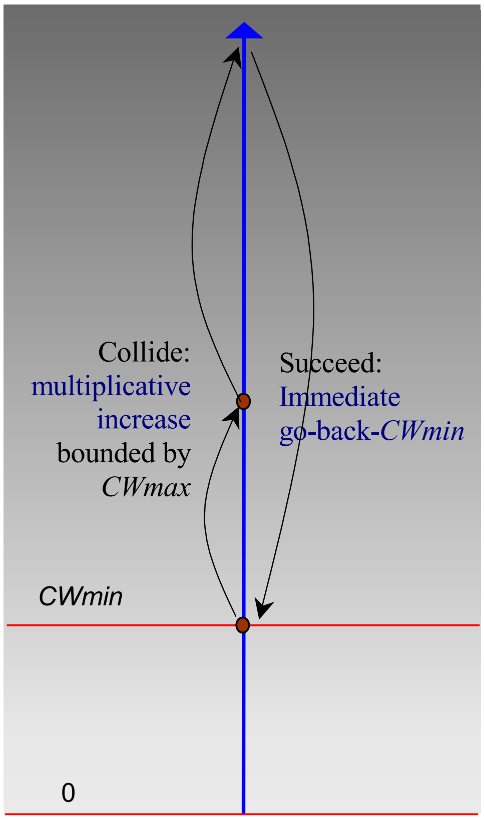
$0 \leq CWmin \leq cw \leq CWmax$

$0 \leq CWmin \leq CWbasic \leq CWmax$

Figure 1. MIMLD backoff algorithm



(a) MIMLD algorithm



(b) Standard algorithm

Figure 2. Comparison of the contention window evolution processes in the two algorithms

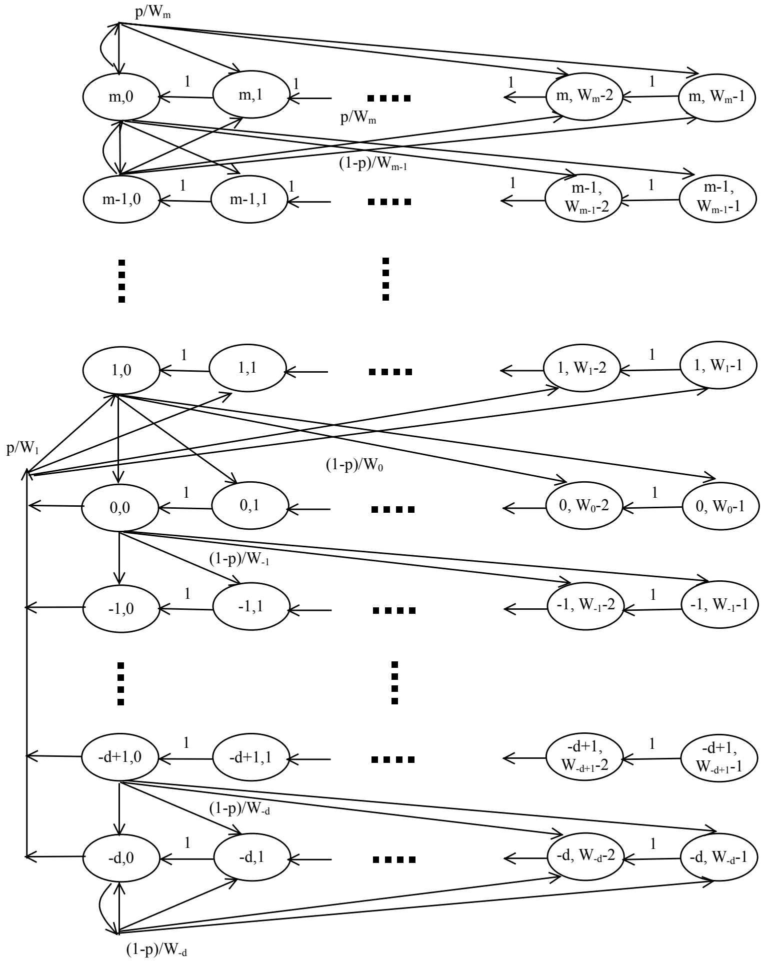
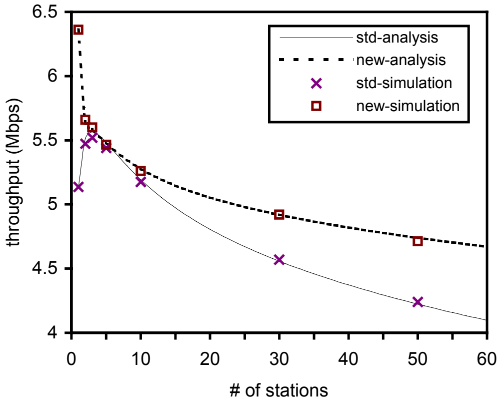
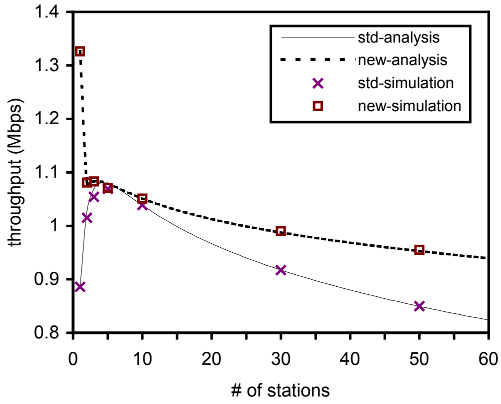


Figure 3. State transition diagram of the new algorithm

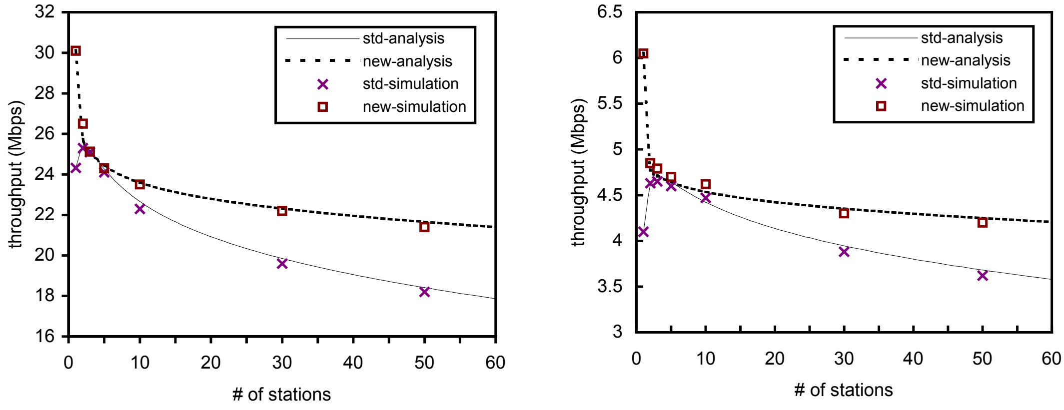


(a) packet size =1000 bytes



(b) packet size =100 bytes

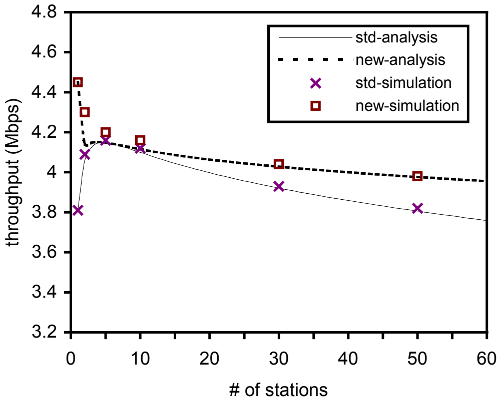
Figure 4. Throughput comparisons - 802.11b



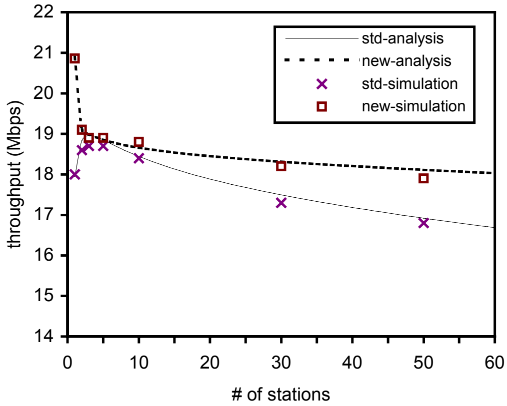
(a) packet size =1000 bytes

(b) packet size =100 bytes

Figure 5. Throughput comparisons - 802.11a/11g



(a) 802.11b



(b) 802.11a/11g

Figure 6. Throughput of RTS/CTS access (packet size=1000 bytes)

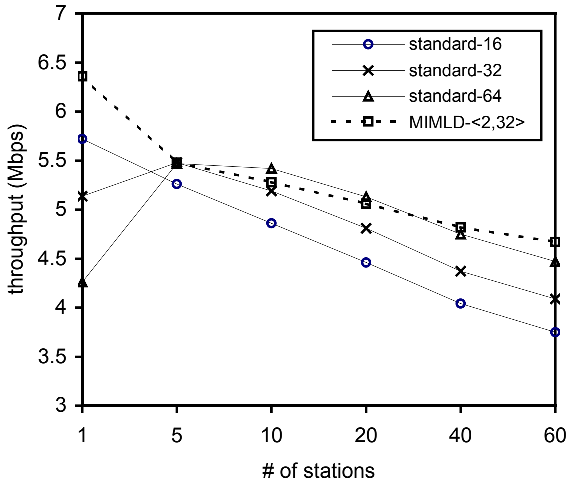
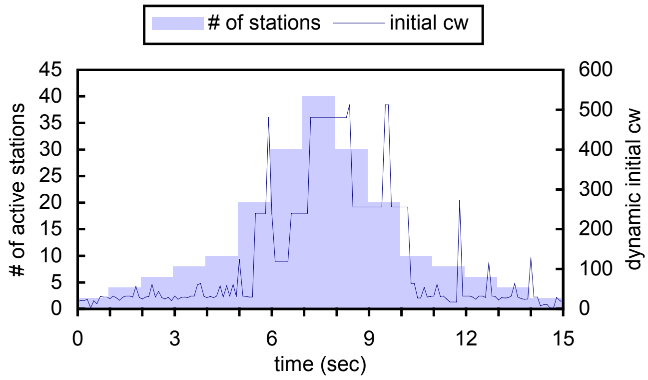
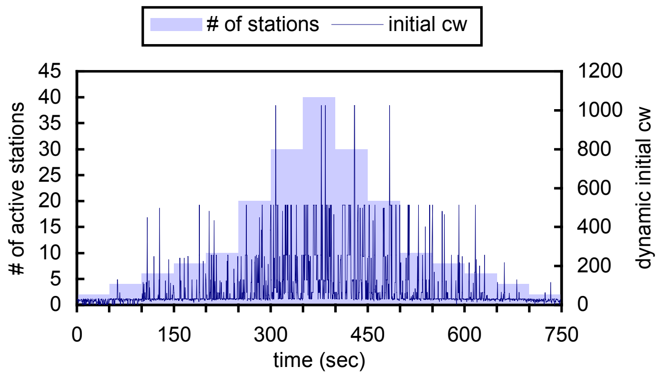


Figure 7. Comparison of the standard algorithm and the MIMLD algorithm



(a) step = 1sec, sampling interval = 0.1sec.



(b) step = 50sec, sampling interval = 0.5sec

Figure 8. Dynamics of the sampled initial contention windows in MIMLD

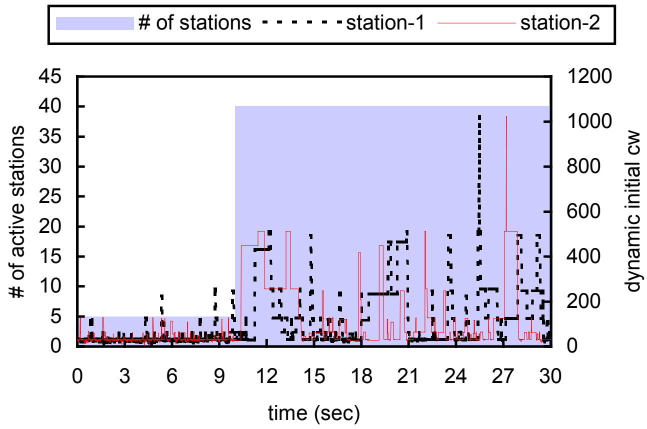


Figure 9. Reaction of contention window to abrupt change in number of active stations

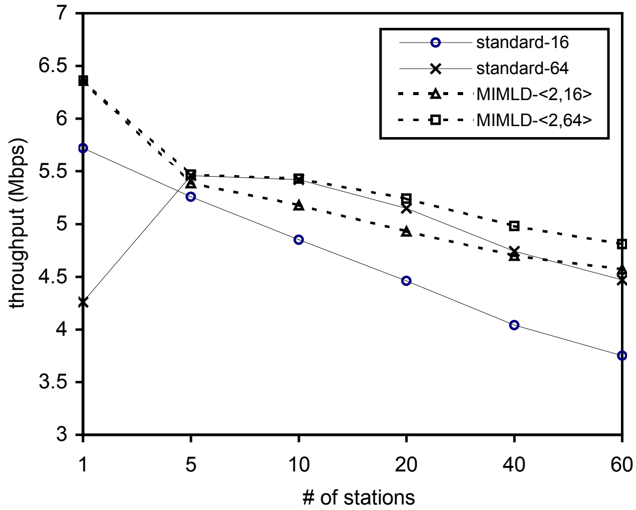


Figure 10. Effect of MIMLD parameters <*CWmin*, *CWbasic*>

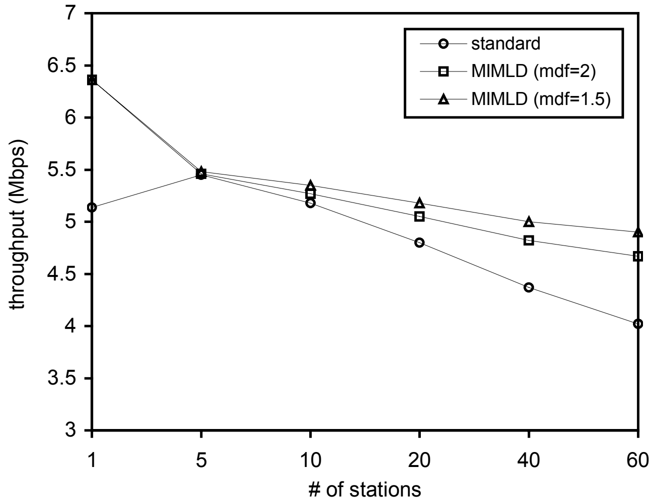


Figure 11. Effect of different multiplicative decrease factors in MIMLD

- $cw_i \leftarrow max\ (cw_i\ /\ mdf_i,\ CWbasic_i)$, *if (succeeds and* $cw_i > CWbasic_i$*).*
- $cw_i \leftarrow max\ (cw_i - ldf_i,\ CWmin_i)$, *if (succeeds and* $cw_i \leq CWbasic_i$*).*
- $cw_i \leftarrow min\ \{mif_i \times max\ (cw_i,\ CWbasic_i),\ CWmax\}$, *if collides.*
- $cw_i \leftarrow cw_i$, *if retry limit is reached.*

$0 \leq CWmin_i \leq CWbasic_i \leq CWmax$

mdf_i: *multiplicative decrease factor of class i;*

mif_i: *multiplicative increase factor of class i;*

ldf_i: *linear decrease factor of class i.*

Figure 12. Service differentiation using the MIMLD algorithm

Table 1. Similarity between MIMLD and TCP

MIMLD contention window evolution	TCP congestion window evolution
Multiplicative Decrease	“Slow Start”
Linear Decrease	“Additive Increase”
Multiplicative Increase	“Multiplicative Decrease”

Table 2. Physical properties of 802.11b, 802.11a and 802.11g

	802.11b	802.11a/802.11g
SlotTime	20 µs	9 µs
CCATime	15 µs	3 µs
RxTxTurnaroundTime	5 µs	2 µs
SIFSTime	10 µs	16 µs
PHY overhead	192 µs	20 µs
DataRate	11 Mbps	54 Mbps
BasicDataRate	2 Mbps	6 Mbps

Table 3. Comparison of initial contention windows

# of stations	2	4	6	8	10	20	30	40
Standard algorithm	32	32	32	32	32	32	32	32
MIMLD algorithm (average)	11	25	32	35	38	53	68	120

Table 4. Comparison of Fairness Indices

# of stations	802.11b				802.11a/g			
	Basic		RTS/CTS		Basic		RTS/CTS	
	standard	MIMLD	standard	MIMLD	standard	MIMLD	standard	MIMLD
5	0.999	0.999	0.999	0.996	0.999	0.999	0.999	0.999
50	0.994	0.986	0.993	0.984	0.998	0.993	0.998	0.992

Authors' Biographies

Qixiang Pang is a Research Engineer in the Department of Electrical and Computer Engineering at the University of British Columbia (U.B.C.). Before he joined U.B.C. in July 2004, he worked as Research Fellow in the Department of Information Engineering at the Chinese University of Hong Kong from January 2003 to July 2004. His research on WLAN and ad hoc networks at the Chinese University was supported by the Area of Excellence in Information Technology (AoE-IT) of Hong Kong. From April 2000 to November 2002, he was a Senior Systems Engineer engaging in 2.5G/3G GPRS design and development in the Architecture and System Engineering Group at Motorola Canada. His main research interests are in the areas of wireless networking including WPAN, WLAN, WMAN, 3G and 4G.

Soung C. Liew received his S.B., S.M., E.E., and Ph.D. degrees from the Massachusetts Institute of Technology. From 1984 to 1988, he was at the MIT Laboratory for Information and Decision Systems, where he investigated Fiber-Optic Communications Networks. From March 1988 to July 1993, Soung was at Bellcore (now Telcordia), New Jersey, where he engaged in Broadband Network Research. Soung is currently Professor at the Chinese University of Hong Kong.

Soung's research interests include wireless networks, Internet protocols, multimedia communications, optical networks, and broadband packet switch design. He has over 80 publications in these areas combined. Most recently, he has been working on 1) applications and communications protocols over wireless networks, focusing on performance, security, and mobility issues; and 2) wireless LAN and ad-hoc network designs.

Soung is currently co-Director of the Area of Excellence in Information Technology (AoE-IT) of Hong Kong, a multimillion-dollar joint project with participations from the Chinese University, Hong Kong University of Science and Technology, and University of Hong Kong. Currently, two main themes for research at AoE-IT are 1) Pervasive Multimedia Content Delivery of Heterogeneous Internet and 2) Intrusion Detection on Internet.

Besides academic activities, Soung is also active in the industry. He co-founded two technology start-ups in Internet Software and has been serving as consultant to many companies and industrial organizations. He is currently consultant for the Hong Kong Applied Science and Technology Research Institute, helping to formulate R&D directions and strategies in the areas of Wireless Internetworking, Applications, and Services.

Soung is the holder of three U.S. patents and Fellow of IEE and HKIE. He is listed in Marquis Who's Who in Science and Engineering. He is the recipient of the first Vice-Chancellor Exemplary Teaching Award at the Chinese University of Hong Kong.

Jack Y. B. Lee received his BEng and PhD degrees in Information Engineering from the Chinese University of Hong Kong in 1993 and 1997 respectively. He is currently Associate Professor of the Department of Information Engineering at the Chinese University of Hong Kong and directs the Multimedia Communications Laboratory (http://www.mcl.ie.cuhk.edu.hk) to conduct research in distributed multimedia systems, fault-tolerant systems, peer-to-peer systems, multicast communications, and Internet computing. Jack is a Senior Member of IEEE and a Professional Member of ACM.

Victor C. M. Leung received the B.A.Sc. (Hons.) degree in electrical engineering from the University of British Columbia (U.B.C.) in 1977, and was awarded the APEBC Gold Medal as the head of the graduating class in the Faculty of Applied Science. He attended graduate school at U.B.C. on a Natural Sciences and Engineering Research Council Postgraduate Scholarship and obtained the Ph.D. degree in electrical engineering in 1981.

From 1981 to 1987, Dr. Leung was a Senior Member of Technical Staff at Microtel Pacific Research Ltd. (later renamed MPR Teltech Ltd.), specializing in the planning, design and analysis of satellite communication systems. He also held a part-time position as Visiting Assistant Professor at Simon Fraser University in 1986 and 1987. In 1988, he was a Lecturer in the Department of Electronics at the Chinese University of Hong Kong. He joined the Department of Electrical Engineering at U.B.C. in 1989, where he is a Professor, Associate Head of Graduate Affairs, holder of the TELUS Mobility Industrial Research Chair in Advanced Telecommunications Engineering, and a member of the Institute for Computing, Information and Cognitive Systems. His research interests are in the areas of architectural and protocol design and performance analysis for computer and telecommunication networks, with applications in satellite, mobile, personal communications and high speed networks.

Dr. Leung is a Fellow of IEEE and a voting member of ACM. He is an editor of the IEEE Transactions on Wireless Communications, and an associate editor of the IEEE Transactions on Vehicular Technology. He has served on the technical program committee of numerous conferences, and is serving as the Technical Program Vice-Chair of IEEE WCNC 2005.