A TCP-like Adaptive Contention Window Scheme for WLAN

Qixiang Pang, Soung Chang Liew, Jack Y. B. Lee,
Department of Information Engineering
The Chinese University of Hong Kong
Hong Kong

S.-H. Gary Chan
Department of Computer Science
Hong Kong University of Science and Technology
Hong Kong

***Abstract*—This paper proposes and investigates a simple self-adaptive contention window adjustment algorithm for 802.11 WLAN. We present simulation and analytical results showing 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 both when the number of active stations is large and small – that is, in both heavy and light contention cases. Furthermore, the adaptive window adjustment algorithm is simpler than previously proposed enhancement schemes in that no live measurement of the WLAN traffic activity is needed.**

Keywords - IEEE 802.11; Wireless Local Area Networks; WLAN; CSMA/CA; Congestion Control; Self-Adaptive Contention Window Adjustment

I. INTRODUCTION

802.11 wireless local area networks (WLAN) operating at the unlicensed ISM frequency bands [1] is one of the few highlights of communication technologies in recent years. The Media Access Control (MAC) and physical (PHY) layers for WLAN are defined by IEEE 802.11 [2].

Within the family of 802.11 standardized 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, has recently been finalized in June 2003.

Two types of MAC access protocols, Distributed Coordination Function (DCF) and Point Coordination Function (PCF) are defined in 802.11. However, 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 [3]-[8]. 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 original 802.11 algorithm, this algorithm dynamically adjusts the initial contention window to a near optimal point according to the traffic activity, thus avoiding bandwidth inefficiency due to improper contention window setting. Compared with other performance enhancement algorithms, our algorithm is effective both when there are many and few active stations. Furthermore, our algorithm does not require on-line measurement and computation.

The rest of the paper is organized as follows. Section II reviews the 802.11 standard and related work. Section III describes the proposed algorithm. Section IV presents the simulation results in various scenarios, and provides the performance comparisons of the new and standard algorithms. Section V concludes the paper.

II. 802.11 STANDARD AND RELATED WORK

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

For the transmission of each packet, a random backoff time (in slot) is selected uniformly between 0 and *cw*-1. The value of *cw* is called the contention window, and it depends on the number of previous transmission failures for that packet. At the first transmission attempt, *cw* is set to a value *CWmin* called the minimum (initial) 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 values of *CWmin* and *CWmax* are 32 and 1024 in 802.11b [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 excessive collisions and backoffs; on the other hand, when there are few active stations, too high a *CWmin* may lead to unnecessary idle airtime during which no station attempts to transmit. In either case, the channel is not used efficiently.

This work is sponsored by the Areas of Excellence scheme established under the University Grant Committee of the Hong Kong Special Administrative Region, China (Project Number AoE/E-01/99).

To solve the problem, one could employ a dynamic contention window adjustment algorithm with window size adjusted to reflect the number of active stations in the WLAN. Adaptive window adjustment algorithms different from the 802.11 WLAN standard have been studied in [3]-[8]. However, these methods suffer from a number of disadvantages that render them impractical: 1) they are effective only when there are a large number of stations [5][7]; 2) they require accurate active on-line measurement of the number of active stations [3][4][6][8]; 3) they assume constant packet size for the traffic over WLAN [3][6][8].

In real applications, it is quite common that there are only a few active stations with packets to send even when there are many stations in the WLAN. Web browsing, for example, will only cause a client station to transmit packets sporadically. In the home environment, the number of stations itself could be quite small, often just one.

On-line measurement of active stations and computation of the optimal contention window incur extra processing cost. Measurement and computation errors could even lead to worse performance than the original standard algorithm.

To overcome the problems in the original 802.11 and other enhancements algorithms, we propose in this paper a self-adaptive contention window algorithm – MIMLD that emulates the TCP window adjustment mechanism. As will be demonstrated, the algorithm performs well both when the number of active stations is large and small. And as with TCP, it does not require direct measurement of the traffic activity in the channel.

III. DESCRIPTION OF MIMLD ALGORITHM

Figure 1. shows our self-adaptive MIMLD contention window adjustment algorithm for WLAN. The algorithm is simple in principle. The major difference between our algorithm and the original 802.11 standard is that the contention window adjustment in our algorithm adapts to the contention intensity of the wireless channel.

At any one time, MIMLD can be in one of three possible phases: multiplicative increase phase, multiplicative decrease phase, and linear decrease phase. A new control parameter called *CWbasic* is introduced into this algorithm. *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 initial window size in the original 802.11 algorithm.

When collision occurs, as in the original algorithm, the contention window is doubled. A difference in our algorithm is that if the doubled value is still below *CWbasic*, *CWbasic* is adopted. 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 increases the probability that this might not be the case. This window increase phase is called "multiplicative increase".

- $CWnew \leftarrow max\ (CWnew/2, CWbasic)$, *if (succeeds and* $CWnew > CWbasic$*).*
- $CWnew \leftarrow max\ (\text{--}CWnew, CWmin)$, *if (succeeds and* $CWnew \leq CWbasic$*).*
- $CWnew \leftarrow max\ (2 \times CWnew, CWbasic)$, *if collides.*
- $CWnew \leftarrow CWnew$, *if retry limit is reached.*

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

Figure 1. Self adaptive window adjustment 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 the initial contention window size immediately, the contention window is halved but 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 avoided. 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 stations or the traffic is asymmetric (e.g., dominant 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 initial window size of the original 802.11 algorithm. The reason why we allow the contention window to be below *CWbasic* is based on the observation that in light contention cases (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 transmission attempts. However, although the contention is assumed to be light when the window 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 multiplicatively decreasing the window, it is linearly decreased. Only many consecutive successful transmissions can lower contention window to the minimum value *CWmin*.

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

contention window. The basic algorithms in TCP are Additive Increase and Multiplicative Decrease (AIMD) and slow start. 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.

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 improves the performance of the original 802.11 algorithm in both cases

IV. PERFORMANCE EVALUATION

In this section, the performance of MIMLD is evaluated and compared with the original algorithm. TABLE I. shows the protocol control parameters used in the original 802.11 algorithms. The control parameters used in MIMLD will be varied to study its behavior under different parameter values. We used the *NS-2* simulator [10] for our simulations.

A. Limit Analysis of Single-Active-Station Case

First, the throughput limit and delay when there is only one station are analyzed. The purpose is to study the improvement that can be obtained when there is no collision at all. With no collision, the improvement is solely due to the use of smaller *CWmin,* and not to the dynamic adjustment of *cw*.

Assume that the packet payload size (*Pk*) is constant and the station works under saturation condition (i.e., the station always has packets available for transmission [3]). The maximum throughput of one saturated station is given by:

$$S_{max} = \frac{Pk}{\left\{\begin{array}{l}(Pk + MAC)/\ DataRate + 2PHY + SIFS + DIFS \\ + ACK\ /\ BasicRate + T_{slot} \times (CW_{min} - 1)/2\end{array}\right\}} \quad (1)$$

The average backoff delay (not including queueing delay and ACK delay) of a packet is determined by:

$$D_{min} = \frac{Pk + MAC}{DataRate} + PHY + DIFS + T_{slot} \times \frac{(CW_{min} - 1)}{2} \quad (2)$$

where,

MAC is the MAC header in bits (28×8=224 bits);
DataRate is the physical layer data rate;
PHY is the overhead in physical layer;
SIFS is the time of SIFS;
ACK is the size of MAC ACK in bits (14×8=112 bits);
BasicRate is the rate for MAC ACK transmission;
T_{slot} is the time of one slot.

Two parameters determine the values of $S_{\max}$ and $D_{\min}$: the packet size, *Pk*, and the contention window size, $CW_{\min}$. Because in our new algorithm, the value of $CW_{\min}$ is smaller than that in the standard, the throughput can be increased and the delay decreased. Note that without losing accuracy, the propagation delay is ignored in (1) and (2).

TABLE I. PARAMETERS OF STANDARD 802.11b IN SIMULATION

CWMin	32
CWMax	1024
SlotTime	20 us
CCATime	15 us
RxTxTurnaroundTime	5 us
SIFSTime	10 us
PHY overhead	192 us
MaxPropagationDelay	2 us
DataRate	11 Mbps
BasicRate	2 Mbps

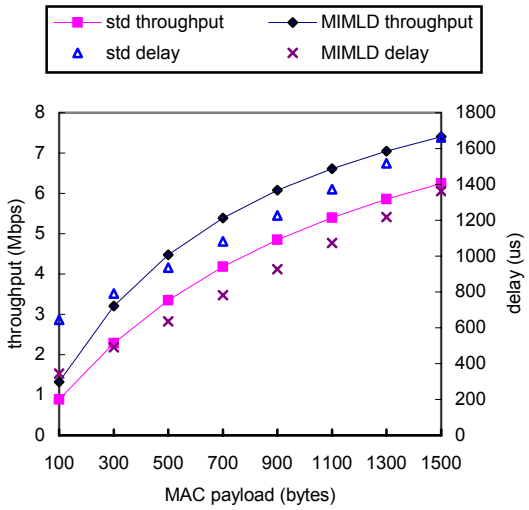

Figure 2. Throughput and delay limits of one station.

Figure 2. compares the throughputs and delays. In the standard algorithm, *CWmin* are 32. In MIMLD, *CWmin* is set to be 2. The value of *CWbasic* here does not matter since there is no contention. Clearly, when there is no contention, the use of a smaller *CWmin* can yield improved performance. The next few sections show that MIMLD can also achieve better performance when there are more active stations.

B. Multiple-Active-Station Case

We first assume the number of active stations is fixed and does not vary dynamically over time. Section D will consider the dynamic case. The buffers of all stations are saturated and the stations always have packets to send. In MIMLD, the control parameters, *CWmin*, *CWbasic*, and *CWmax* are, 2, 32, and 1024, respectively. For the sake of comparison, we assign *CWbasic* and *CWmax* in our algorithm the values of *CWmin* and *CWmax* defined in the 802.11 standard.

From the simulations results (Figure 3. and Figure 4.), it can be concluded that MIMLD yields improvements whether the number of active stations is large or small. In particular, when the number of stations is very large (say, >=10) or it is very small (say, <=3), MIMLD exhibits more improvement. For instance, the percentages of improvement for one single station are 24% (1000 bytes packet size), 50% (100 bytes packet size) respectively. In the case of 90 stations, the improvements are 21% (1000 bytes packet size), 22% (100 bytes packet size) respectively.

C. Dynamic Changing of Number of Active Stations

We design a simulation scenario with dynamic changing of number of active stations. 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 from 2 to 4, 6, 8, 10, 20, 30, and finally reaches 40. Then the number decreases 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 UDP packets (1000 bytes) in a saturated manner one after another. The size of changing step is 1 sec.

Figure 5. shows the sampled initial contention windows. The sampling interval is 0.1 sec. From the diagram, when the number of stations is high, the initial contention window tends to be at a high level. And when the number of stations is small, the initial contention window is at a low level. This is because in the case of heavy contention, our algorithm allows the contention window to be at a high level while in the case of light collision, the window goes to a lower level.

D. Effect of the Initial Parameter Settings

We now study the effects of control parameter values. Figure 6. shows the throughput differences when the control parameters in MIMLD, <*CWmin CWbasic*>, are assigned different initial values. Each active station operates at the 802.11b physical layer and sends packets (1000 bytes) in a saturated manner in the simulations.

By comparing the results of MIMLD and the standard algorithms with different initial window sizes, we conclude that the new algorithm is robust and less sensitive to the initial contention window settings.

Figure 6. also includes the 'optimal' throughput by applying the optimal contention window [11] calculated from equation (3):

$$CW_{min} = (n-1)\sqrt{\frac{\frac{Pk+MAC}{DataRate} + PHY + SIFS + \frac{ACK}{BasicRate} + PHY + DIFS}{T_{slot}}} \quad (3)$$

In (3), *n* denotes the number of contending stations; other notations are same as those in (1) and (2).

The results show that when there are not many stations, our algorithm, MIMLD, can yield more throughput than the "dynamically optimized" standard algorithm.

On the other hand, it also shows that when there are many stations, there is room for improvement for MIMLD. We present some preliminary results on how to further improve MIMLD by adjusting the multiplicative decrease factor in Figure 7. Three different values of multiplicative decrease factor are studied, 1.25, 1.5, and the original 2. In all the three cases, the contention windows are, *CWbasic* = 32, and *CWmin* = 2. When the contention window decrease factor is 1.25, the throughput of MIMLD becomes quite close to that of dynamically optimized standard algorithm when the number of stations is large. And the throughput is not degraded compared to decrease factor of 2 and 1.5 when the number of stations is small.

V. CONCLUSIONS

In this paper, we have proposed and studied a new self-adaptive contention window adjustment algorithm, MIMLD, for DCF in 802.11 WLAN.

Compared with previously proposed enhancement algorithms, MIMLD exhibits performance improvements over the original algorithm for all range of number of active stations. In the case of very few stations and very many stations, significant improvement is obtained. In addition, our algorithm also does not need to assume constant packet size in its optimization procedure.

Because MIMLD does not need on-line measurement and computation, it boasts simplicity. MIMLD can be easily implemented by minor modifications of the 802.11 firmware.

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 that has been widely proven robust.

Possible future investigations are as follows:

1) The performance of 802.11a/11g has not been studied in this paper. Although it remains to be confirmed, we believe similar results can be expected for 802.11a/11g.

2) RTS/CTS has not been considered in this paper due to time and space limits. RTS/CTS is an optional mechanism in 802.11 to overcome the hidden-terminal problem and can potentially increase throughput when the transmitted packet size is large and the contention intensity is high. How much improvement relative to the standard algorithm can be achieved with MIMLD when RTS/CTS is turned on remains to be investigated.

3) In the new MAC standard of 802.11, 802.11e, an enhanced DCF mechanism is defined to provide service differentiation over WLAN [12]. In 802.11e, the values of the control parameters, such as contention window, inter-frame space, transmission opportunity, can be set differently. By applying the same principle, we can extend our new algorithm to its enhanced version – enhanced MIMLD algorithm. For different traffic categories, different values of *CWmin*, *CWbasic*, multiplicative increase/decrease factor, and linear decrease factor can be adopted. The selection of proper values for different traffic classes remains to be investigated.

REFERENCES

[1] M. S. Gast, 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] G. Bianchi, L. Fratta, M. Oliveri, "Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs," IEEE PRIMRC 1996, pp.392-396.

[4] F. Cali, M. Conti, E. Gregori, "IEEE 802.11 protocol: design and performance evaluation of an adaptive backoff mechanism," IEEE J. Select. Areas Commun., vol. 18, no. 9, pp. 1774-1786, Sep. 2000.

[5] H. Wu, S. Cheng, Y. Peng, K. Long, J. Ma, "IEEE 802.11 distributed coordination function (DCF): analysis and enhancement," IEEE ICC 2002, pp.605-609.

[6] Y. Peng, H. Wu, S. Cheng, K. Long, "A new self-adapt DCF algorithm," IEEE GLOBECOM 2002, pp.87-91.

[7] N. Song, B. Kwak, J. Song, and L. E. Miller, "Enhancement of IEEE 802.11 distributed coordination function with exponential increase exponential decrease backoff algorithm," IEEE VTC 2003 Spring, pp.2775-2778.

[8] Y. Chen, Q. Zeng, D. P. Agrawal, "Performance analysis and enhancement for IEEE 802.11 MAC protocol," ICT 2003, pp. 860-867.

[9] Y. Xiao, J. Rosdahl, "Throughput and delay limits of IEEE 802.11," IEEE Commun. Lett., vol. 6, no. 8, pp. 355-357, Aug. 2002.

[10] NS-2, URL http://www.isi.edu/nsnam/ns.

[11] Y. C. Tay, K. C. Chua, "A capacity analysis for the IEEE 802.11 MAC protocol," Wireless Networks, 7, pp. 159-171, 2001.

[12] IEEE WG, "802.11e D6", Nov. 2003.

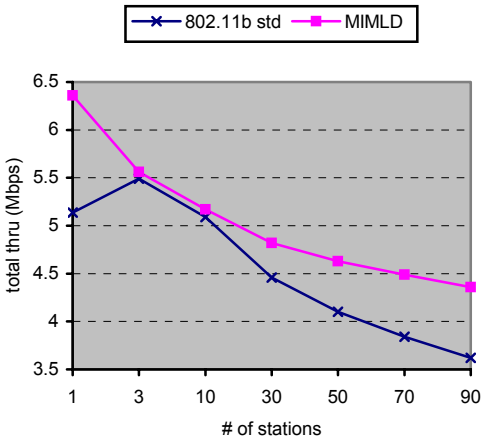

Figure 3. Throughput comparison (pk size =1000 bytes)

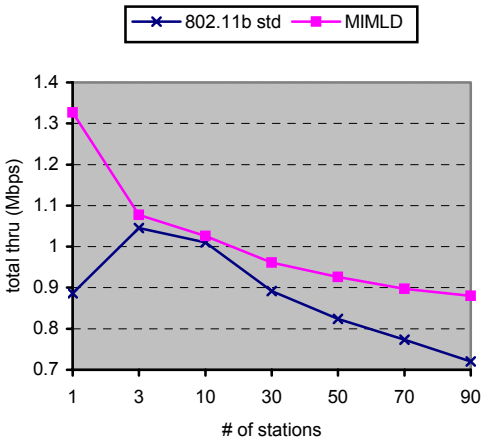

Figure 4. Throughput comparison (pk size =100 bytes)

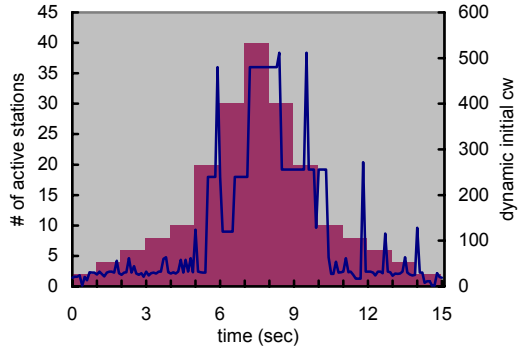

Figure 5. Dynamics of the sampled initial contention window

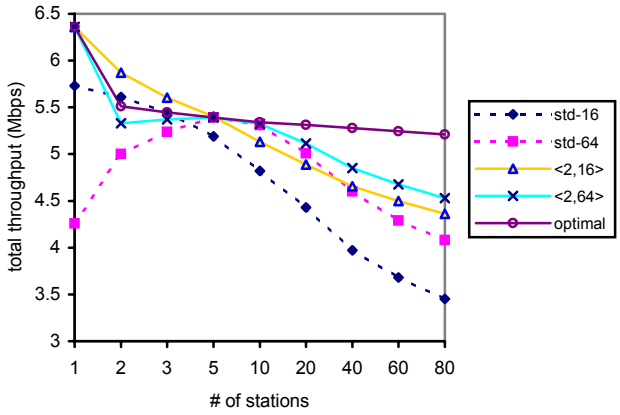

Figure 6. Effect of initial settings of contention window

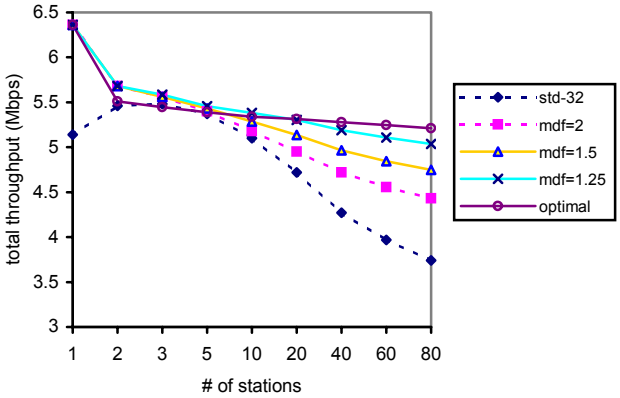

Figure 7. Effect of decrease factor