Adaptive Contention Window Scheme for WLANs

Elwathig Elhag and Mohamed Othman
Faculty of Computer and Information Technology, University Putra Malaysia, Malaysia

Abstract: *In this paper, a new bakeoff algorithm is proposed to enhance the performance of the IEEE 802.11 Distributed Coordination Function (DCF) which employs Binary Exponential Backoff (BEB) algorithm. We present simulation results showing that the new algorithm outperforms the BEB algorithm and compared with the previously proposed enhancement algorithms, a salient feature of our algorithm is that it performs well 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 WLANs traffic activity is needed and don't assume constant packet sizes.*

Keywords: *IEEE 802.11, wireless local area networks, WLANs, DCF, binary exponential backoff, ACW.*

Received February 2, 2006; accepted April 30, 2006

1. Introduction

Since Wireless Local Area Networks (WLANs) can provide more flexible and convenient connections than wired networks, there is an increasing demand for wireless local area networks recently. To speed up the design of WLANs, the IEEE 802.11 study group proposes an international standard [6] for WLANs. The standard defines detail functions for both the Medium Access Control layer (MAC) and the Physical Layer (PHY). Within the family of 802.11WLANs, the most widely deployed version so far is 802.11b, which operates at 2.4GHz and provides up to 11Mbps data rate.

In the IEEE 802.11 standard, the Distributed Coordination Function (DCF) is defined for asynchronous data transmissions. The DCF shares access to the medium based on the CSMA/CA protocol. Note that collision detection is not used in WLANs, since a station is unable to detect the channel and transmit data simultaneously. To address this issue, CSMA/CA was developed instead. The basic idea of CSMA/CA is "listen before talk", where a station which desires to transmit must sense the medium before transmission to determine whether another user is transmitting. CSMA/CA employs an immediate positive acknowledgment scheme to make sure successful reception of packets. The receiving station sends the acknowledgment packet after a short time interval. If an acknowledgment is not received, the packet is considered lost and a retransmission is arranged.

The DCF mechanism is simple. However it has been shown by many references [2, 3, 4, 5, 9, 10, 12, 15, 16, 17, 18] that the standard DCF cannot efficiently utilize the limited wireless channel bandwidth when there are many stations in the WLANs accessing the same channel. The major reason is that the initial Contention Window (CW) size is kept fixed regardless of the traffic activity, whereas it should be large when the number of active stations is large and small when a number of stations small, all these literatures highlight the poor performance of the system as a whole in term of system throughput and to improve the fairness of 802.11 MAC because of low utilization of channel.

The main contribution of this paper is to propose new scheme termed as Adaptive Contention Window (ACW) unlike the standard BEB algorithm, this algorithm should automatically adjusts the CW to near optimal point according to the traffic activity, thus avoiding bandwidth wastage due to improper CW setting. Compared with other enhancement algorithms [2, 3, 4, 5, 9, 10, 12, 15, 16, 17, 18], 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 online measurement and computation and perform well for full range of packet sizes.

This paper is organized as follows. In section 2, we briefly review both the basic and RTS/CTS access mode of the DCF and the IEEE 802.11 MAC layer and related work. Section 3 introduces the ACW algorithm. Section 4 presents simulation results in various scenarios, and compares the performance of the ACW, BEB, EIED [12] and MIMLD [10] algorithms. Section 5 concludes the contribution of the paper.

2. Related Work

The IEEE 802.11 standard defines both MAC and PHY layers. One of the most important functions of the MAC layer of IEEE 802.11 is to coordinate the wireless medium access procedure. The fundamental access method in the IEEE 802.11 MAC protocol is the Distributed Coordination Function (DCF). Most commercial products only implement DCF. The DCF is a random access schema shares access to the medium based on (CSMA/CA) protocol. Retransmission of collided packets is managed according to binary exponential backoff rules, the IEEE 802.11 standard also provides an alternative access method, called the Point Coordination Function (PCF), which is an access method similar to a polling system and uses a point coordinator to determine which station to transmit.

There are two techniques used for packet transmitting in DCF. The default one is a two-way handshaking mechanism also known as basic access method. Acknowledgement (ACK) is transmitted by the destination station to prove the successful packet transmission. The other optional one is a four-way handshaking mechanism, which uses request-to-send/clear-to-send (RTS/CTS) technique to reserve the channel before data transmission

The BEB algorithm is widely used in MAC layer protocols due to its simplicity. In this algorithm, each node doubles its CW value up to Contention Window Maximum value (CWMax) after a collision and resets CW to Contention Window Minimum value (CWMin) after a successful transmission [6, 10, 12]. The standard BEB algorithm has two main problems discussed in the literature: first, the BEB in which the backoff is doubled after every collision and reduced to the minimal backoff after every successful transmission this does not provide an adequate level of fairness[2, 9] as example if one station has too high backoff counter and another station has small backoff counter it well reach the zero first and transmit it's packets and reset the CW to minimum value and may be can choose small backoff value again and transmit when the another station don't reach zero until now because it backoff value is too high. Second the BEB backoff calculation adjusts extremely rapidly [12]; it both backs off quickly when a collision is detected and it also reduces the backoff counter to CWMin value immediately upon a successful transmission. This produces rather large variations in the backoff counter because each new packet starts with the CWMin value, which can be too small for the heavy network load.

There are different algorithms and schemes in the literature proposed to resolve the problems of the BEB algorithm. We divided them in two methods: on-line methods and non on-line methods. In On-line Methods algorithms [2, 4, 9, 15, 18], a relationship between the number of stations and the optimal CW is first established then on-line sniffer is built inside each station to monitor the activities of all surrounding stations when the number of stations is estimated the CW is adjusted accordingly. We can note that the analysis in [2] showed the deterioration of the throughput when the number of nodes increase. In [9], the author presented a simple analytical model to compute the saturation throughput performance assuming a finite number of nodes and an ideal channel conditions; he demonstrated that the optimal CW size strongly depends on the number of contending nodes. In [18], They propose A Priority Backoff Algorithm (PBA), the basic idea of PBA is each station should collect statistical data of other stations transmission while sensing the channel and maintain a Sent Data Table (SDT) for all stations in network. When the station has data to transmit it will calculate CW based on the statistical data in SDT and its priority. In [15] the author proposed a new measurement-based algorithm to adaptively configure the optimal value of the initial CW value to improve the throughput and fairness. However, it also needs to compute current channel status at run time and adjusts the RTS/CTS message structure. As general on-line measurement of active stations and computation of the optimal CW incur extra processing cost and are hard to implement and most of them require exchange of information between nodes and complicated computation. 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 CW 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.

In non on-line methods [3, 5, 10, 12, 16, 17], these methods easy to implement and don't need exchange of information between nodes and don’t depend on packet sizes like on-line methods. But they only considered the case that there are a large number of stations, however, in real applications it quite common that there are only a few active stations in WLANs. In the home environment for example the number of stations is normally quit small. This paper [3] proposes CW resetting scheme to enhance the performance of IEEE 802.11 DCF and new analytical model based on Markov chain is introduced to compute the throughput of the proposed scheme. Increase and Linear Decrease (LMILD) backoff algorithm is proposed in [17], where the collision nodes increase their CW multiplicatively, while other nodes overhearing the collisions increase their CW linearly. After a successful transmission, all nodes decrease their CW linearly. They found that the optimum value for CW in a network with N active nodes is: CW = 5.1 * N. However this expression is calculated when using the access method RTS/CTS. And the number of nodes N is estimated by observing the channel status. Our proposed scheme ACW does not need to estimate network parameters such as competing node number and channel status in [3] and [17], where they require exchange of information between nodes and complicated computation.

In Exponential Increase Exponential Decrease (EIED) backoff algorithm [12], whenever a packet transmitted from a node is involved in a collision, the CW size for the node is increased by backoff factor r_I. The CW for the node is decreased by backoff factor r_D if the node transmits a packet successfully. The EIED backoff algorithm can be represented as follows.

$$CW = \min [r_I.CW, CW_{max}] \text{ on a collision}$$
$$CW = \max [CW/r_D, CW_{min}] \text{ on a success}$$

In Multiplicative Increase Multiplicative/Linear Decrease Backoff Algorithm (MIMLD) [10] they considered the small number of stations as ACW algorithm also but add new parameters (e. g., CWBasic) to standard to play the role of a threshold for distinguishing the contention intensity between heavy and light load of the wireless channel and it set to be close to the value of CWMin in the original 802.11 algorithm, so they changed the meaning of CWMin. Compare EIED and MIMLD algorithms with ACW algorithm we consider the small and large number of stations without any new parameters added to the standard. However ACW more flexible and over a wider range of scenarios give better performance. The MIMLD backoff algorithm can be represented as follows.

- *cw* ← *max* (*cw*/2, *CWbasic*), *if* (*succeeds and cw* > *CWbasic*).
- *cw* ← *max* (==*cw*, *CWmin*), *if (succeeds* and *cw* ≤ *CWbasic*).
- *cw* ← *min* (*2xmax* (*cw*, *CWbasic*), *CWmax*), *if collides*.
- *cw* ← *cw*, *if retry limit is reached*.

where:
$0 \leq CW_{min} \leq cw \leq CW_{max}$
$0 \leq CW_{min} \leq CWbasic \leq CW_{max}$

Our aim in this work is to find a non on-line method that is simple to implement and without adding any new parameters to standard, as well as our algorithm should provide near optimal performance in both light and heavy load. And the new algorithm should perform well for full range of packet sizes.

3. Description of ACW Algorithm

In 802.11 DCF, the value of CW has the minimal value CWMin. After each collision, the CW will be doubled until reaching the maximum CWMax. After each successful transmission, the CW will be reset to CWMin regardless of the network conditions such as the number of current competing nodes, this method tends to work well when there are only a few competing nodes. When the number of competing nodes increases, it will be shown to be ineffective since the new collisions can potentially occur and cause significant performance degradation. So even the number of nodes has increased to a very large value, the nodes will use the same initial CW. As a result, a lot of collisions occur and the throughput is deteriorated.

Since a node uses CW to control the backoff window, the optimal setting of CWMin will affect the performance. In 802.11 DCF, the CWMin is fixed (32 slots in DSSS 802.11b) regardless of the number of contending nodes. We think that for each number N of nodes, there is an optimal value of CWMin. Where, if we decrease CWMin to a value less than the optimal value, there will be more collisions, which will degrade the performance. At the same way, if we increase CWMin to a value greater than the optimal value, the packet transmitted will suffer from a longer delay, which will also degrade the performance.

In order to adapt the CWMin according to the number of nodes, we introduce the ACW algorithm. The main idea is that, in order to reduce the number of collisions, the CW size for each node is carefully selected according to the number of contending nodes; this CWMin will maximize the channel utilization and the throughput. CWMin should be increased linearly with the number of nodes N. We have found using simulation that the relation between the CWMin and the number of nodes takes the following equation:

$$CWMin = aN - b \tag{1}$$

where a, b is constant numbers, N the number of contending nodes and CWMin the minimum contention window size measured by slots.

3.1. Optimal CW Size for 802.11b

To prove our algorithm we have executed many scenarios for different number of stations in which we have changed the value of CWMin for many values (from 1 to 600 slots) we have found that the Optimal CW size for 802.11b is obtained by the following equation:

$$CWMin = 8N - 6 \tag{2}$$

This formula has been obtained by simulation; we applied this formula when CW size is doubled on a collision and halved when there is a successful transmission.

We can see the effect of changing CWMin for a different number of actives nodes in Figure 1. The throughput performance depends on the number of contending nodes and on the initial CW size. For a given CW, the throughput decreases drastically when the number of nodes increases. In the other hand, for a given number of nodes, the throughput has a maximum value that depends on the value of the CWMin. An initial small contention window size provides a sufficiently more of collision probability, especially when the number of nodes is large. Figure 1, shows the impact of the CWMin on the performance by measuring the throughput for different values of CWMin. The results of 5, 10, 20, and 30 contending nodes are shown in Figure 1. The throughput depends highly on CWMin and on the number of the contending nodes. For example, a low value of CWMin (35 slots) which is optimum for 5 nodes would not be suitable for a large number of nodes 30 for example and at the same time, a high value of CWMin (375 slots) which is optimum for 30 nodes would drastically penalize the throughput if the number of nodes is small 5 for example. As conclusion, to achieve an optimal throughput, the system parameters must be selected according to the traffic conditions, in particular, the optimal value of CWMin which depends on the number of the contending nodes.

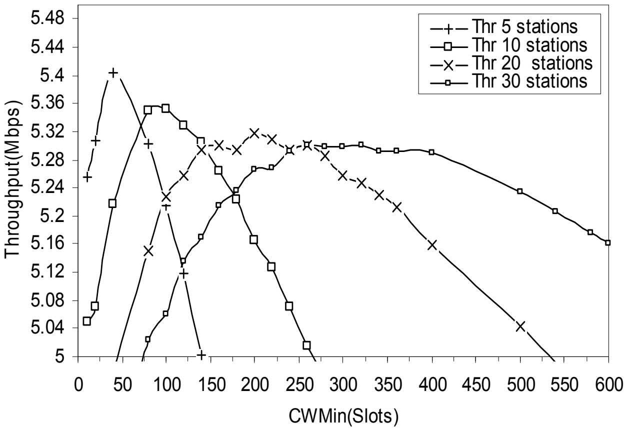

Figure 1. Throughput for different number of stations in function of CWMin using 802.11b.

We have executed many simulations for different loads. We found that, for a certain number of nodes there is an optimal CWMin which maximizes the throughput. Figures 2-4, demonstrates the effect of CWMin on the throughput using 802.11b. We discuss the case of 5 stations in details and same goes to another number of stations. In Figure 2, we can distinguish three intervals of CWMin. For a CWMin less than 30 slots, the throughput is poor because there are many collisions. For CWMin greater than 50 slots, the throughput is also poor because of the long waiting time when using a big CWMin. Finally, for a CWMin more than 30 slots and less than 50 slots, we can observe that the throughput is maximized with a variation roughly of 5Kbit/s. Even through the load in the network increases, the value of agreed CWMin which maximizes the throughput is going to be in this interval (30 < CWMin < 50) and more precisely around of CWMin = 35. The throughput is approximately constant in this interval.

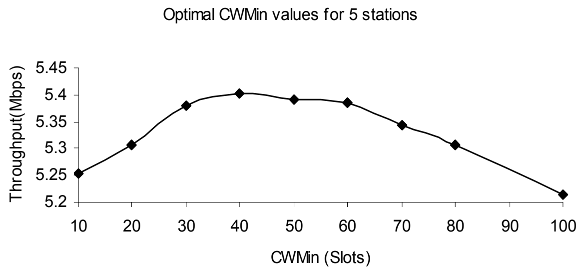

Figure 2. Optimal CWMin values for 5 stations.

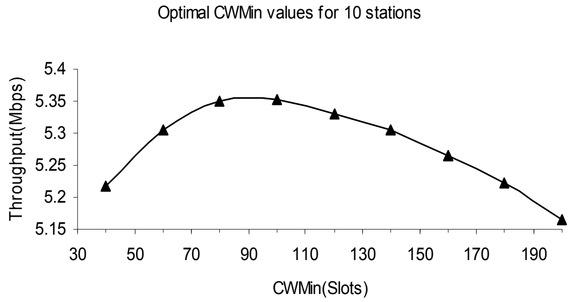

Figure 3. Optimal CWMin values for 10 stations.

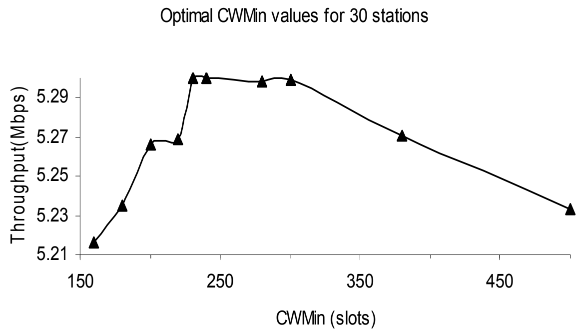

Figure 4. Optimal CWMin values for 30 stations.

3.2. Adaptive 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 BEB algorithm, we observe 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 an on-line method. However, as discussed in section 2, the on-line methods have some drawbacks that render them impractical. Therefore a Non on-line method is preferable.

Our proposed ACW algorithm and the standard, MIMLD, EIED algorithms all are categorize as Non on-line method. However ACW algorithm differ from those algorithms a sense that the initial contention window of a packet transmission is changed automatically with the optimal value for CWMin depends on the number of contending stations accessing the wireless channel. In Figure 5, we found the relation between the number of contending nodes and the optimal CW size depending on the results of many simulations showed in the Figures 1-4. For a given number of contending nodes, several scenarios were executed for different traffic load. From these figures and by undertaking an approximation of the experimental values of CWMin, we found that the value of CWMin whose maximizes the throughput performance is given by the formula (2).

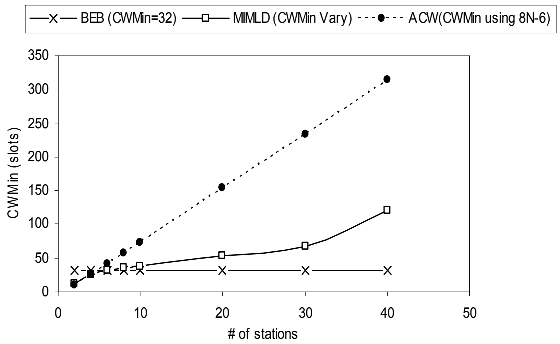

Figure 5. Comparison of initial contention windows using 802.11b packet size 1000 byte.

The ACW algorithm has more throughputs compare with the MIMLD, EIED and BEB algorithms. In ACW algorithm the initial CW adjusts automatically with the change of number of stations. It can be seen from Figure 5 the initial CW in ACW algorithm is smaller than the standard and MIMLD when the number of stations is small and large when the number of stations is large. This explain the reason why ACW algorithm perform better than the standard and MIMLD algorithms especially when the number of stations large (i. e., >10).

Table 1 compares the initial contention windows in the algorithms when 1000-byte data packets are transmitted over 802.11b by saturated stations. In Table 1 for the standard algorithm BEB and the ACW 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 BEB algorithm always uses CWMin (which is 32) as its initial contention window regardless of the number of stations. In contrast, in the ACW algorithm, the initial contention window is adjusting automatically with the change of the number of stations depending on the formula (2). It is clear from Table 1 that the initial contention window in ACW algorithm is smaller than that in the standard BEB algorithm when the number of stations is small and larger when the number of stations is large. When compared with MIMLD the CWMin in ACW is large than MIMLD especially when the number of stations is large (i. e., >10). This explains the reason why the proposed algorithm ACW performs better than the standard BEB algorithm and MIMLD algorithm.

Table 1. Comparison of initial contention windows using 8N - 6 in 802.11b.

No of Stations	2	4	6	8	10	20	30	40
BEB Algorithm	32	32	32	32	32	32	32	32
MIMLD Algorithm	11	25	32	35	38	53	68	120
ACW Algorithm	10	26	42	58	74	154	234	314

4. Simulation

In this section, we prove the validity of our scheme ACW algorithm by compare our results with the standard BEB, the MIMLD [10] and the EIED [12] algorithms by simulation using NS-2 tool [13] in various scenarios. The parameters we used in the simulation and the physical properties of 802.11b in our study are displayed in Table 2. We will compare throughput and fairness indices using basic access and RTC/CTS mechanisms.

Table 2. Physical properties of 802.11b.

Slot Time	20 ms
CCA Time	15 ms
RxTx Turnaround Time	5 ms
SIFS Time	10 ms
PHY Overhead	198 ms
Data Rate	11Mbps
Basic Data Rate	2Mbps

As described previously, there are two problems in standard BEB algorithm. First, the CW is fixed (CWMin) with the fixed CWMin the BEB 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. Secondly, the BEB suffers from fairness problem [10, 2, 9]; we measure the fairness of our algorithm and BEB, MIMLD algorithms. The duration for simulation runs is 300 seconds. Moreover, simulation runs for greater than 300 seconds on 50 stations under high loads are extremely time consuming, and not change the results.

4.1. Throughput 802.11b Packet Size 1000 Basic Access

To study the performance of the ACW algorithm, we compare the results of ACW algorithm with the standard BEB, EIED and the obtained results for MIMLD algorithms in various scenarios. The throughput is measured in our simulation by calculating how many bytes have been received by the traffic sinks for each station in a simulation time in Mbps and gets the total of throughputs for all stations in network.

Simulation results are presented in Figure 6 for 802.11b, the control parameters of our scheme and the standard CWMin, CWMax are 32 and 1024 respectively. For MIMLD algorithm, CWMin, CWBasic (new parameter introduce in the MIMLD algorithm), and CWMax are, 2, 32, and 1024, respectively.

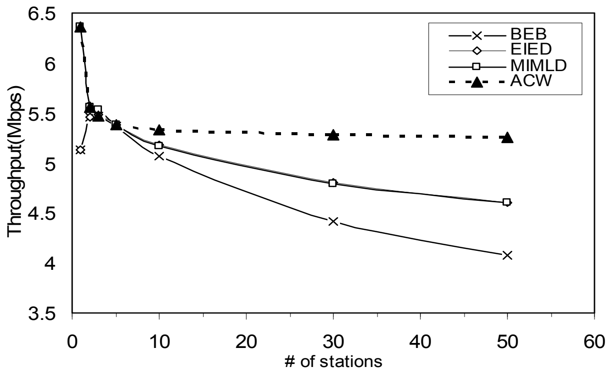

Figure 6. Throughput Comparisons 802.11b packet size 1000 bytes basic access mode.

In Figure 6, 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 by using basic access feature in 802.11, all parameters kept fixed when we change the number of stations 1,2,3,5,10,30,50 the simulation time is 300ms. It can be concluded that the ACW algorithm yields improvements whether the number of active stations is large or small over the EIED, MIMLD and the standard algorithms.

In particular, when the number of stations is very large (i. e., >= 10) or very small (i. e., <=3), the ACW algorithm exhibits improvement. For instance, when compare ACW algorithm with standard BEB algorithm, the percentage of improvements for the ACW algorithm for one single station is 23.3%, and for 50 stations the percentage of improvements is 29%. Then compared ACW algorithm with EIED algorithm, the percentage of improvements for one single station is 23.8%, and for 50 stations the percentage of improvements is 14%.

4.2. Throughput 802.11b Packet Size 100 Basic Access

Simulation results are presented in Figure 7 for 802.11b, the control parameters of ACW and the BEB standard algorithm CWMin, CWMax are 32 and 1024 respectively. For MIMLD algorithm, CWMin, CWBasic and CWMax are, 2, 32, and 1024, respectively. In Figure 7 we measure the throughput for ACW algorithm and compare it with the standard BEB, EIED and MIMLD algorithms but in case of the packet size is 100 bytes to prove ACW can get more improvement not only in case that the contention intensity in wireless channel is high but also in lightly load even with the packet size is small (100 bytes).

We get improvement over standard and EIED and MIMLD especially when the load of wireless channel is high (the number of contending stations >= 10). For instance compare with standard BEB algorithm, the percentage of improvements for ACW for one single station is 50% (802.11b 100 bytes packet size). In the case of 50 stations, the improvements are 26% (802.11b 100 bytes packet size).

Compare ACW with EIED algorithm, the percentage of improvements for ACW for one single station is 50% (802.11b 100 bytes packet size). In the case of 50 stations, the improvements are 12%. Comparing ACW with MIMLD algorithm for 50 stations the improvement is 2%. In particular, when the number of stations is very large (i. e., >= 10) or very small (i. e., <= 3), ACW exhibits more improvement and get more improvement over MIMLD algorithm also especially when the number of stations is very large (>= 10).

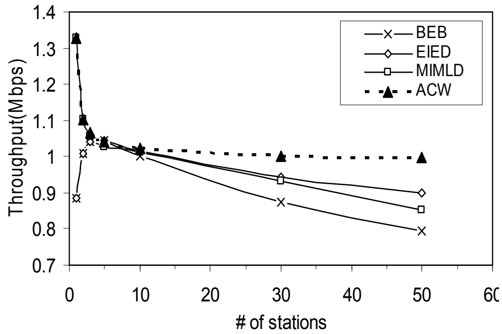

Figure 7. Throughput Comparisons 802.11b packet size 100 bytes basic access mode.

As conclusion from the Figure 6 and Figure 7, we can observed that the throughput of standard BEB algorithm which use constant value of CWMin equal to 32 in case of 802.11b it suffer from low throughput ,where the probability of packet collisions will increase. We observe that the EIED and MIMLD algorithms don't choose the optimal CWMin especially when the number of stations is large because increasing the value of CWMin when N increases, will improve the throughput but when N is too high, some stations may have an extremely small chance to access to the channel successfully. We can see that the choice of CWMin has a large influence on the network performance and this choice is strongly depend on the number of contending stations.

4.3. Throughput 802.11 b Packet Size 1000 Using RTS/CTS Access Mode

RTS/CTS access is an optional feature in the 802.11 standard and is helpful to overcome the hidden terminal problem. In NS-2 tool we can turn on/off this feature. Simulation results are presented in Figure 8, for the physical version 802.11b, the control parameters of ACW algorithm and the standard CWMin, CWMax are 32 and 1024 respectively. For MIMLD algorithm, CWMin, CWBasic, and CWMax are, 2, 32, and 1024, respectively. Each active station attempts to send UDP packets (1000 bytes) in a saturated manner one after another using RTS/CTS access feature in 802.11 all parameters kept fixed when we change the number of stations 1, 2, 3, 5, 10, 30, 50 the simulation time is 300ms.

Figure 8 shows the saturation throughput of RTS/CTS access mode, the X-axis is the number of stations and Y-axis is the saturation throughput in Mbps. We can observe that the ACW algorithm improves the performance of RTS/CTS access as well, although the amount of improvement is not as high as that of basic access. The ACW has a higher throughput since it can decrease the chance of collision by adaptively adjusting the contention window. Comparing with standard BEB algorithm the percentage of improvement for one single station is 17%. In case of 50 contending stations the percentage is 12%. When we compare with EIED algorithm for one station the improvement is 17% and for 50 stations is 6%. Comparing the ACW algorithm with MIMLD algorithm the percentage of improvement for 50 stations is 5.4%.

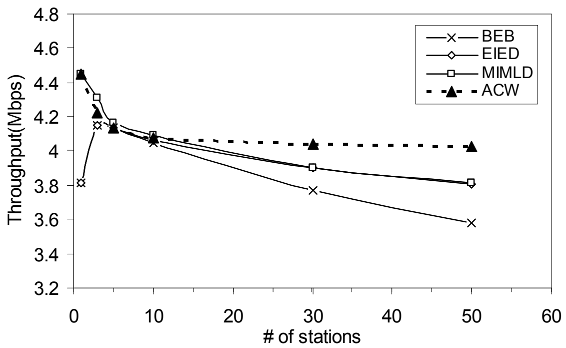

Figure 8. Throughput Comparisons 802.11b packet size 1000 bytes RTS/CTS mode.

4.4. Fairness of ACW Algorithm

The standard algorithm (BEB) in which the backoff is doubled after every collision and reduced to the minimal contention window after every successful transmission this does not provide an adequate level of fairness [2, 9, 10] we mention that in section 3. By simulation, we compare the fairness of our algorithm and the MIMLD algorithm and the standard algorithm. The well-known Jain's fairness index [7] is used.

The definition of Jain's index is given by:

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

where:

- x_i: The throughput of contending flow (station) i.
- n: The number of contending flows (stations).

Fairness measures how evenly the connections share the scarce resource amongst themselves. The fairness index always lies between 0 and 1 for non-negative throughputs like what we see in Table 3, and is equal to (k / n) if k of the n stations receive equal throughput and the remaining $(n - k)$ none. Thus, fairness cannot be less than $1/n$ in a network with n contending stations. The results shown in Table 3 prove that our proposed algorithm ACW is fairness comparing with the standard or MIMLD algorithms when multiple connections (stations) are simultaneously active and contending to access the wireless channel.

Table 3 gives the measured fairness indices for ACW algorithm and the MIMLD and the standard algorithms. In the scenarios, saturated stations sending data packets (packet size is 1000 bytes) are simulated for 100 s. The results show that the fairness of my algorithm is better than these algorithms especially when N is large.

Table 3. Comparison of fairness indices using 802.11b.

	Basic Access Mode			RTS/CTS Access Mode		
No of Stations	BEB	MIMLD	ACW	BEB	MIMLD	ACW
5	0.9998	0.9995	0.9999	0.9999	0.9965	0.9997
10	0.9994	0.9984	0.9990	0.9992	0.9980	0.9997
30	0.9952	0.9922	0.9995	0.9968	0.9890	0.9994
50	0.9937	0.9825	0.9992	0.9940	0.9814	0.9992

5. Conclusions

The major contribution in this paper is an ACW algorithm; we have proposed and studied a new adaptive contention window adjustment algorithm, ACW, for DCF in 802.11 Wireless Local Area Networks (WLAN). Simulation results show that the new algorithm outperforms the standard 802.11 window adjustment algorithm. We have found that the throughput performance is strongly dependent on the number of actives nodes and the total load offered to the system. We have modified the initial contention window size to be relative to the number of contending nodes, which can be easily calculated from the routing table in each node. We found via the simulation results the expression of the optimal CWMin in function of the active stations in order to maximize the throughput of the system. The simulation results show that ACW has much better performance than the traditional DCF and the obtained results in [10, 12].,

The ACW algorithm has the following advantages over the standard BEB, MIMLD and EIED algorithms:

- A performance improvement over a full range of number of active stations and it does not need to assume constant packet size.
- Simple to implement because ACW algorithm does not need on-line method or calculation, it can be easily implemented by minor modifications of IEEE 802.11 standard firmware.

- Through a simulation it has been proved that ACW algorithm is more fairness in terms of load distribution compared with other proposed algorithms.

Due to time and space limits, 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. So in future investigation we try to enhanced version of ACW algorithm by found the optimal CWMin in function of active stations for 802.11a/11g physical versions to prove ACW algorithm can get more performance than other proposed backoff algorithms not in heavy and light load only but for different physical versions even if the packet size not constant. How much improvement relative to the standard algorithm can be achieved with ACW algorithm when RTS/CTS are turned on using 802.11a/11g remains to be investigated.

References

[1] Altman E. and Jimenez T., "NS Simulator for Beginners," *Lecture Notes*, University de Los Andes, Merida, Venezuela and ESSI, Sophia-Antipolis, France, 2004.

[2] Bianchi G., Fratta L., and Oliveri M., "Performance Evaluation and Enhancement of the CSMA/CA MAC Protocol for 802.11 Wireless LANs," *in Proceedings of IEEE Conference Personal Indoor Mobile Radio Communications (PIMRC)*, Taipei, Taiwan, vol. 1, no. 1, pp. 407-411, 1996.

[3] Chatzimisios P., Boucouvalas A. C., Vitsas V., Vafiadis A., Oikonomidis A., and Huang P., "A Simple and Effective Backoff Scheme for the IEEE 802.11 MAC Protocol," *in Proceedings of the International Conference on Cybernetics and Information Technologies, Systems and Applications (CITSA 2005)*, USA, July 2005.

[4] Chen Y., Zeng Q., and Agrawal D. P., "Performance Aalysis and Enhancement for IEEE 802.11 MAC Protocol," *in Proceeding of IEEE International Conference on Telecommunications (ICT)*, vol. 1, no. 3, pp. 860-867, 2003.

[5] Deng J., Varshney P. K., and Hass Z. J., "A new Backoff Algorithm for the IEEE 802.11 Distributed Coordination Function," *in Proceedings of Communication Networks and Distributed Systems Modeling and Simulation*, CNDS 04, San Diego, CA, USA, 2004.

[6] IEEE Standard for wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, ISO/IEC 8802-11,1999E (R2003).

[7] Jain R., "Throughput Fairness Index: An Explanation," *ATM Forum Contribution*, pp. 99-1045, 1999.

[8] NS by Example, Tutorial, available at: http://nile.wpi.edu/NS/, November 2005.

[9] Peng Y., Wu H., Cheng S., and Long K., "A New Self-Adapt DCF Algorithm," *in Proceedings of IEEE Global Telecommunications Conference (GLOBECOM02)*, vol. 1, no. 1, pp. 87-91, 2002.

[10] Qixiang P., Soung C., Jack Y., and Victor C., "Performance Evaluation of an Adaptive Backoff Scheme for WLAN," *Wireless Communications and Mobile Computing*, vol. 4, no. 1, pp. 867-879, 2004.

[11] Rizzoli A. E., "A Collection of Modelling and Simulation Resources on the Internet," http:// www.idsia.ch/~andrea/simtools.html, November 2005.

[12] Song N., Kwak B., Song J., Miller L. E., "Enhancement of IEEE 802.11 Distributed Coordination Function with Exponential Increase Exponential Decrease Backoff Algorithm," *in Proceedings of IEEE Vehicular Technology Conference (VTC)*, vol. 5, no. 1, pp. 2775-2778, 2003.

[13] The Network Simulator NS-2 homepage, http://www.isi.edu/nsnam/ns/, October 2005.

[14] The NS2 Manual, http://www.isi.edu/nsnam/ ns/ns-documentation.html, November 2005.

[15] Wang C., Li B., and Lemin L., "A New Collision Resolution Mechanism to Enhance the Performance of IEEE 802.11 DCF," *in Proceedings of IEEE Transactions on Vehicular Technology (VTC)*, vol. 53, no. 4, pp. 1235-1246, 2004.

[16] Wen-Kuang K. and Kuo C. C. J., "Enhanced Backoff Scheme in CSMA/CA for IEEE 802.11," *in Proceedings of IEEE Vehicular Technology Conference (VTC)*, vol. 5, no. 1, pp. 2809-2813. 2003.

[17] Wu H., Cheng S., Peng Y., Long K., and Ma J., "IEEE 802.11 Distributed Coordination Function (DCF): Analysis and Enhancement," *in Proceedings of IEEE International Conference on Communications (ICC)*, NY, USA, vol. 1, no. 1, pp. 605-609, 2002.

[18] Yan S., Zhou Y., Wu S., Guo W., "Priority Backoff Algorithm for IEEE 802.11 DCF," *in Proceedings of IEEE Vehicular Technology Conference (VTC)*, vol. 4, no. 1, pp. 423-427, 2004.

Elwathig Elhag received his bachelor degree in computer science, University of Khartoum, Sudan, in 1999, and master of computer science, University Putra Malaysia (UPM) in 2006. He worked as programmer and lecturer in AL-Alamiah Electronic Company, Kingdom of Saudi Arabia from 2000 to 2004. His research interests include computer and data networks, design and analysis, computer languages, specifically object oriented programming and design, distributed database, and data mining techniques.

Mohamed Othman received his BSc from UKM, MSc from UTM, PhD from UKM in 1999, Best PhD thesis in Malaysia. Currently, he is the head of Department of Communication Technology and Network, University Putra Malaysia, Malaysia. His research interests include parallel and distributed algorithms (multigrid algorithm on SMP and cluster platforms, parallel and sequential implementation of LUC cryptography algorithm), scientific computing (multigrid algorithm, points and block iterative algorithms), high speed network (dynamic bandwidth allocation algorithm, chordal ring networks).