Ubiquitous Computing and Communication Journal
(ISSN 1992-8424)

A NEW BACKOFF ALGORITHM FOR IEEE 802.11 DCF MAC PROTOCOL IN MOBILE AD HOC NETWORKS

Wail Mardini, Muneer Bani Yassein, Zainab AbuTaye'

Department of Computer Science
Jordan University of Science and Technology
Email: {mardini | masadeh}@just.edu.jo, zoabutaye08@cit.just.edu.jo

ABSTRACT

Backoff algorithms are one class of collision resolution algorithms used in the medium access control protocol in mobile ad hoc networks. When there are different nodes competing to access a shared channel at the same time, the possibility of collision is highly probable, especially in high traffic load networks. Collision is considered as the major problem in wireless networks, so the backoff mechanism should be applied in order to decrease the collision and to achieve an efficient use of the shared channel. This paper aims to propose and evaluate a new backoff algorithm called "Dynamic Backoff Algorithm (DBA)" which is a combination of two increment methods (exponential, linear) that applied for N phases and it also has one decrement method (linear).

Keywords: MANETs, IEEE 802.11, backoff, Contention Window.

1 INTRODUCTION

Mobile Ad hoc Networks (MANETs) [3, 23] work without requiring any preexisting communication infrastructure. These types of networks gain a high importance and attract attention due to the need of rapid deployment in emergency cases such as military operations, search and rescue operations and disaster recovery that do not have enough time to build an infrastructure. A MANET is an autonomous system of wireless nodes connected by wireless links. Each node not only acts as a sender or receiver but also as a router in order to convey the packet via intermediate nodes until reach the desired destination (multi-hop technique). These nodes have mobility characteristic that allow forming a dynamic network topology which is highly changeable and random.

The occurrence of packets collision is considered as the major problem in wireless networks and this problem is usually tackled by applying backoff mechanisms. Once the collision has occurred, the collided nodes are needed to defer for a period time which usually refer to as retransmission delay (or backoff) which is usually selected randomly from bounded contention window that has a predetermined lower and upper values. The number of active nodes and traffic load in the network has a direct impact on these values. As an example CW_{min} and CW_{max} are usually set to 31 and 1023 respectively in IEEE 802.11 DCF, and set to 2 and 1024 respectively in Ethernet [1, 13, 18, 22].

This paper is outlined as follows. Section 2 presents an overview of IEEE 802.11 standard. Section 3 displays related works on backoff algorithms. Section 4 describes the proposed algorithm, how it works and presents our experiments that we made in order to choose suitable threshold values for our proposed algorithm. The simulation results for our proposed algorithm are displayed in Section 5. Conclusion and Future Work are drawn in Section 6.

2 OVERVIEW of IEEE 802.11 STANDARD

IEEE 802.11 MAC protocol is a widely used standard in Wireless Local Area Network (WLAN) in its both types: Infrastructure-based and Infrastructure-less (Ad hoc) networks [5, 15, 24].

The MAC protocol, positioned in data link layer (layer 2 in the ISO OSI reference model) [2, 6, 7], plays a vital role in controlling the access to the shared medium and thus reducing the collision as can as possible. This protocol defines two medium access coordination functions called Distributed Coordination Function (DCF) which is used mandatory and Point Coordination Function (PCF) which is used optionally [8, 9].

In ad hoc network, IEEE 802.11 MAC Protocol uses the DCF to access shared channel. DCF offers asynchronous feature, contention based, and a distributed access to the channel [10]. In addition, it offers a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism and Request to Send/Clear to Send (RTS/CTS) techniques [7].

IEEE 802.11 DCF uses a Binary Exponential

Backoff (BEB) algorithm [8, 11, 12, 21] which uniformly chooses the backoff value from the Contention Window (CW). The BEB algorithm is used in IEEE 802.11 due to its simplicity and good performance in many used scenarios [13, 14], but it suffers from fairness and stability problems. It also achieves low throughput under high traffic load [24]. It works as follows: when a node attempts to transmit the packet, it listens to the channel before that, using a carrier sensing technique. If the channel is idle for duration more than the Distributed Inter-Frame Space (DIFS), the node has a privilege to access the channel and start transmitting. Otherwise, the node waits for a time period of DIFS and the backoff algorithm will be applied. Each node selects a random amount of time denoted as Backoff time (BO) in the range [0, CW-1] which is uniformly chosen from the contention window.

The following equation is used to calculate the backoff time (BO):

$$BO = Random\ (0, CW-1) * SlotTime \qquad (1)$$

Where Random( ) is pseudorandom integer taken from a uniform distribution over the interval [0, CW-1], CW is an integer within the range of values of the Management Information Base (MIB) attributes $CW_{min} \leq CW \leq CW_{max}$ and SlotTime is the duration of one slot which is equal to 20 µs for DSSS PHY in IEEE 802.11 [28].

Fig. 1 displays the values of exponential increase of CW in BEB Algorithm. At the first transmission attempt of a packet, the minimum contention window size is initially selected by BEB which is equal 31. The backoff time is randomly selected from the CW to be any value between 1 and CW-1. Upon unsuccessful transmission caused by collision or lost packet, CW becomes double + 1 at each time.

The Contention Window can take one of the values 31, 63, 127, 255, 511, 1023. When the contention window reaches its maximum value (1023 in BEB), it remains there until it can be reset due to transmit packet successfully or dropped the packet that exceeds the maximum limit of allowed transmission retries which bounded by m (m=7 for basic access mechanism (long frames) and m=4 for RTS/CTS mechanism (short frames)) [8, 29].

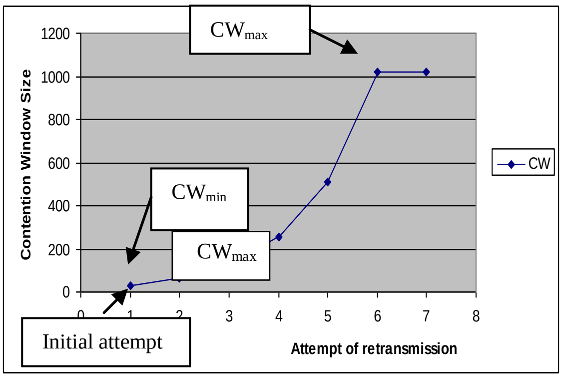

Figure 1: Exponential Increase of CW in BEB Algorithm**.**

Researchers have proposed many backoff algorithms in order to utilize the performance for IEEE 802.11 DCF MAC protocol. Some of these algorithms will be mentioned in the following section.

3 BACKGROUND and RELATED WORK

There are many backoff algorithms proposed in the literature in order to decrease the collision and to achieve an efficient use of the shared channel.

In [19], the authors have proposed Medium Access with Collision Avoidance-Wireless (MACAW) protocol which used a Multiplicative Increase Linear Decrease (MILD) Backoff algorithm. In the MILD algorithm, the nodes increase their contention window multiplicatively upon collision or failure in transmission and decrease their contention window linearly upon success. This algorithm introduced to address unfairness problem in Binary Exponential Backoff (BEB) algorithm.

Manaseer and Masadeh [17] have proposed a Pessimistic Linear Exponential Backoff (PLEB) algorithm. This algorithm based on assumption that the failure in transmission process is caused by the congestion in network. It is considered as a result of combination between linear and exponential increment methods. Using these two increment methods will help to achieve the aim of this algorithm in improving the performance of a MANET in terms of network throughput and average packet delay. By using the linear increment, this algorithm will improve the performance by reducing network delay. On the other hand, using the exponential increment will improve network throughput.

Authors in [10, 16] have proposed Fibonacci Increment Backoff (FIB) algorithm and Logarithmic (LOG) Backoff algorithm respectively. The former algorithm uses a famous math series called Fibonacci Series which aims to reduce the differences between successive contention window sizes, this algorithm achieves a higher throughput when compared with BEB algorithm, the later algorithm uses Logarithmic increments in order to utilize the distribution of random numbers. It achieves a higher throughput and less packet loss. It also achieves stability of network throughput over various speeds of nodes.

In [13], Haas and Deng have proposed the Sensing Backoff Algorithm (SBA) in order to utilize the network throughput and fairness issues. This algorithm based on sensing mechanism (overhearing the channel to get the needed information). So, each node changes its backoff interval based on the results of the sensed channel status.

Authors in [14] have proposed linear Multiplicative Increase Linear Decrease (LMILD) Backoff algorithm, in this algorithm the collided nodes multiplicatively increase their contention windows, while other nodes overhearing the collision increase their contention window in linear way. Upon a success, all nodes decrease their contention windows in a linear way.

In [8], Exponential Increase Exponential Decrease (EIED) backoff algorithm was proposed to improve the performance of the IEEE 802.11 DCF. Upon a collision or Failure, nodes exponentially increase their contention window and upon a success all nodes exponentially decrease their contention windows. This algorithm surpasses BEB in terms of throughput and delay.

Authors in [15] have proposed Predictive DCF (P-DCF) Backoff algorithm to be used in IEEE 802.11 DCF. This algorithm enables nodes to choose their next backoff times by listening to the channel continuously. It reduces the collision probability and outperforms the BEB algorithm in terms of throughput and delay.

4 DYNAMIC BACKOFF ALGORITHM (DBA)

In our work, there are some important issues that should be taken into account when trying to design a backoff algorithm that aims to improve the performance over the network such as determining the methods used to increase and decrease the CW and selecting suitable increase and decrease factors.

4.1 Details of the Algorithm

Dynamic Backoff Algorithm (DBA) aims to take advantages of two increment methods (linear and exponential) for N times. When the backoff mechanism is applied, the contention window size needs to be increased as a response to a failure or collision. As the first response to a failure, *DBA* starts to increase the CW size in an exponential way. Increasing by using exponential will improve network throughput by producing enough length of backoff times, so the new Contention Window (CW) is calculated using the following formula:

$$CW_{new} = CW_{old} * K \qquad (2)$$

Where K is an exponential increase factor.

Then, after a number of exponential increments, our algorithm linearly increases the CW in order to avoid the rapid growth of CW size that causes a high increase of the backoff value and thus results in reducing network delay, so the new Contention Window (CW) is changed using the following formula:

$$CW_{new} = CW_{old} + T \qquad (3)$$

Where T is a linear increase factor.

The above process will be repeated for one time also. So, in our work N will be equal two.

Upon a success in transmission, it decreases the CW in a linear way instead of resetting the CW to its minimum value as in BEB. We use a linear decrease mechanism to solve the fairness problem (avoid channel domination of only one node) due to that resetting mechanism in BEB takes only one successful transmission to reach CW_{min} which causes a huge variation of the contention window size and degrades the performance in heavy loaded network since each new packet starts with the minimum contention window value which is considered as a small value in heavy loaded network, so in our algorithm the new Contention Window (CW) is changed using the following formula:

$$CW_{new} = CW_{old} - Y \qquad (4)$$

Where Y is a linear decrease factor.

As justified later in the following subsection, we choose a suitable value of Y to be 2 depends on the experiments. So in this case, it takes a maximum of 496 successful transmissions for DBA to reach CW_{min} while in BEB, it takes only one successful transmission to reach CW_{min}.

Fig. 2 shows the increment behaviour of dynamic backoff algorithm while Fig. 3 illustrates the pseudocode of it.

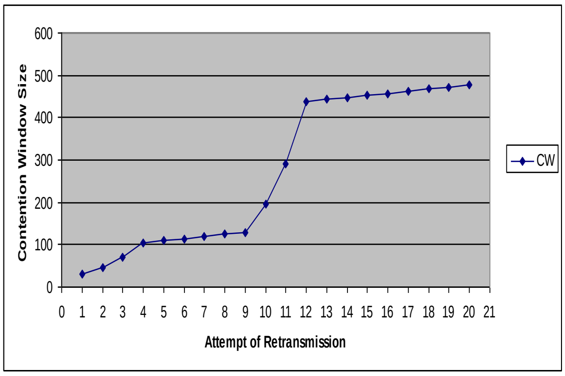

Figure 2: Exponential and linear increase of CW in Dynamic Backoff Algorithm.

```
Step 0: Set BackOffTimer to initial value
Step 1: While BackOffTimer ≠ 0 do
            For each TimeSlot
                If channel is idle then BackOffTimer = BackOffTimer – 1
        If channel is idle for more than DIFS then
                Send
        If (SendFailure) then
          If (NumberOfBackoffs <=W) then
                CW= CW * K
          Else if (NumberOfBackoffs>W && NumberOfBackoffs <=X)
                CW= CW + T
          Else if (NumberOfBackoffs>X && NumberOfBackoffs <=Z)
                CW= CW * U
          Else if (NumberOfBackoffs>Z && NumberOfBackoffs <=O)
                CW= CW + V
           BackOffTimer = Random x;    1 ≤ x ≤ CW -1
      Else
          CW = CW - Y
          BackOffTimer = 0
        Go to Step 1
Step 2: Stop
```

Figure 3: Dynamic Backoff Algorithm

4.2 Experiments to Determine Thresholds for Our Proposed Algorithm

We use Glomosim (version 2.03) simulator [25, 26, 27] to study the impact of different values in order to choose a suitable values and compare them with Pessimistic Linear/Exponential Backoff (PLEB) Algorithm.

Fig. 4 shows the increment behaviour of PLEB while Fig. 5 illustrates the pseudocode of it where k=2, T=5 and W= 124.

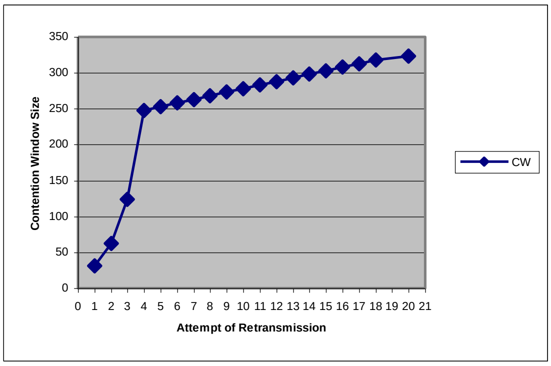

Figure 4: Exponential and linear increase of CW in PLEB.

```
Step 0: Set BackOffTimer to initial value
Step 1: While BackOffTimer ≠ 0 do
            For each TimeSlot
                If channel is idle then BackOffTimer = BackOffTimer – 1
        If channel is idle for more than DIFS then
                Send
        If (SendFailure) then
            If (NumberOfBackoffs <=W) then
                CW= CW * K
            Else
                CW= CW + T
            BackOffTimer = Random x;    1 ≤ x ≤ CW -1


        Else
            CW = Initial value
            BackOffTimer = 0
        Go to Step 1
Step 2: Stop
```

Figure 5: Pessimistic Linear/Exponential Backoff Algorithm [24].

The implementation of dynamic backoff algorithm needs many experiments in order to choose the suitable values of the thresholds especially W which depends on number of exponential increments also to choose values for the increasing and decreasing factors.

Table 1 displays a summary of simulation parameters for this section. The rest of simulation parameters are displayed later in section 5 which have the same values used in this section.

Table 1 SIMULATION PARAMETERS

Parameter	**Value**
Number of nodes	100 nodes
Packet Rate	4 packets/s
Nodes speed	1, 4, 10 m/s

Fig. 6 shows the impact of number of increments of linear and exponential on packet delivery ratio with keeping the resetting mechanism as in BEB and PLEB. Experiment 1 to experiment 4 display 3 increments of exponential in the two phases within increase factors k and u equal 1.5 and varying the number of linear increments in phase 1 to 5, 10, 15, 20 respectively within increase factors t and v equal 5. Experiment 5 to experiment 8 display 2 increments of exponential in the two phases within increase factors k and u equal 1.5 and varying the number of linear increments in phase 1 to 5, 10, 15, 20 respectively within increase factors t and v equal 5.

Similar experiments are held in order to study the effect of our proposed algorithm in terms of throughput and end to end delay in order to justify our selected values.

Results show that experiment 1 has a better prformance in terms of throughput and packet delivery ratio as compared to PLEB with no improvement in delay metric.

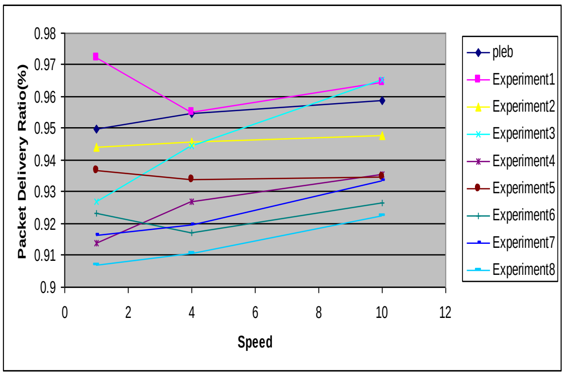

Figure 6: Impact of number of increments of linear and exponential on PDF for K= 1.5, T= 5, U= 1.5, V= 5.

Depends on the results of previous figure and similar experiments which followed the same trend, another type of experiments are held related to experiment 1. Fig. 7 shows the impact of varying exponential increase factor and resetting mechanism of linear on throughput. Experiment 1 to experiment 3 display 3 increments of exponential with k=u= 1.5 but varying the decrease process of CW upon successful transmission as exponentially decrease with y=1.5, resetting to CW_{min} and linearly decrease with y= 2 respectively. Experiment 4 to experiment 6 display 3 increments of exponential at phase 1 and 2 increments at phase 2 with k=u= 1.8 but varying the decrease process of CW upon successful transmission as exponentially decrease with y=1.8 , resetting to CW_{min} and linearly decrease with y= 2 respectively. Experiment 7 to experiment 9 display 3 increments of exponential at phase 1 and 1 increment at phase 2 with k=u= 2 but varying the decrease process of CW upon successful transmission as exponentially decrease with y=2 , resetting to CW_{min} and linearly decrease with y= 2 respectively.

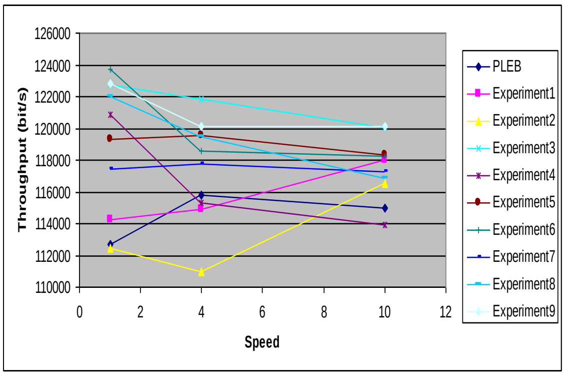

Figure 7: Impact of varying exponential increase factor and resetting mechanism of linear on throughput.

Similar experiments are held in order to study the effect of our proposed algorithm in terms of Packet delivery ratio and end to end delay in order to justify our selected values which followed the same trend.

Results show that experiment 3 has a better performance in terms of throughput and packet delivery ratio as compared to PLEB and the delay is approximately the same.

Based on the results of the above figures, in this paper we focus our discussion to K= 1.5, T= 5, U= 1.5, V= 5, Y= 2 as they offer better performance in comparison to the other possible values, based on these values W = 69.75, , X = 124.6 , Z = 291.7 and O = 1023. In section 5, we will study the impact of change of density and mobility on our proposed algorithm.

5 SIMULATION RESULTS AND ANALYSIS

Simulation has been carried out using GloMoSim simulator version 2.03 to evaluate the performance of our algorithm. The network consists of 100 nodes randomly placed in a 2200m × 600m rectangular field. We use the Random Waypoint Mobility Model (RWMM) as the mobility model [4]. Constant bit rate (CBR) with 512 byte data packets is used. The MAC layer protocol is 802.11. The Ad hoc on-demand distance (AODV) routing protocol is used [20].

Each point on every figure represents the average of 5 trials by using different seed values to change the random number generator. The confidence interval for all results is 95%. The main parameters used in the simulations are summarized in Table 2.

Table 2 SIMULATION PARAMETERS

Parameter	**Value**
Simulator	GloMoSim (version 2.03)
Simulation time	500 second
Simulation area	2200 m x 600 m
Number of nodes	25, 50, 75, 100 nodes
Node placement	Randomly
Routing Protocol	Ad hoc on-demand distance protocol (AODV)
Transmission range	250 m
Bandwidth	2Mbps
Mobility model	Random waypoint
Nodes speed	5, 10, 15, 20, 25, 30 m/s
Pause time	0s
Traffic type	CBR
Data packet size	512 byte
Packet rate	4 packets/second
Number of connection	30
Confidence Interval	95%

5.1 Performance Metrics

Performance of Dynamic Backoff Algorithm is evaluated in terms of the following list of some metrics:

1. **Throughput**: the total amount of data packets successfully received at the destination.
2. **Packet Delivery Ratio (PDR)**: the ratio of total amount of data packets received by the destination nodes to the total amount of data packets sent by the CBR sources [30].
3. **Average End-to-End delay (EED)**: is the average delay taken for a data packet that is originated at the source to be received at the destination. All possible delays caused by buffering during route discovery latency, retransmission delays at the MAC, queuing at the interface queue and propagation delay are included [30].
4. **Normalized routing load**: the average amount of routing control packets that are sent over the amount of data packet received.
5. **Routing overhead**: the ratio of the total amount of routing control packets to the total amount of data packets sent to the MAC layer.

5.2 Effect of Network Density

In this set of experiments, the number of nodes varying from 25 to 100, the maximum speed is set to 10 m/s, the packet rate is 4 and the CBR connections are 30. Having a high throughput is one of the main features that should be provided by MAC protocol which implies of having low overhead and low collision probability.

Fig. 8 shows the performance of PLEB and DBA. The figure represents the relation between throughput and network density. When the number of nodes increased within a certain area, the throughput is increased. Both PLEB and DBA increased in the same behavior. However, we notice a slight improvement in throughput by using DBA compared to the PLEB especially in dense networks. In sparse networks with 25 nodes, there is no much difference, however in medium size networks that contains 50 nodes and dense networks that contains 100 nodes, the throughput by DBA is increased by

5%, 4% respectively compared to PLEB.

This can be explained as follows. In sparse networks, when a collision occurs the CW is increased exponentially in both algorithms in the same behavior and there will be a chance to have a transmission sooner after few increases since not many nodes exists. In dense networks, since more calculations are needed to have a successful transmission and the calculations of the two algorithms differ after a certain number of calculations; we see this difference in throughput improvement.

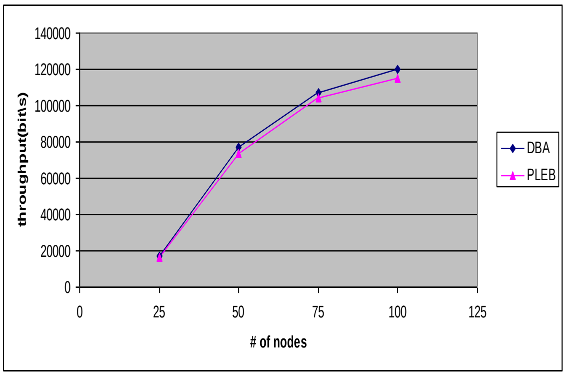

Figure 8: Throughput vs. node density with node speed 10 m/s.

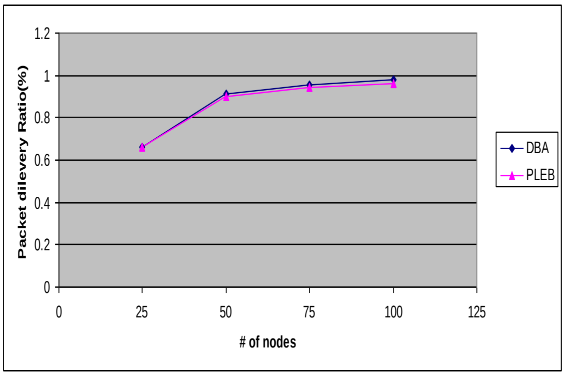

Figure 9: packet delivery ratio vs. node density with node speed 10 m/s.

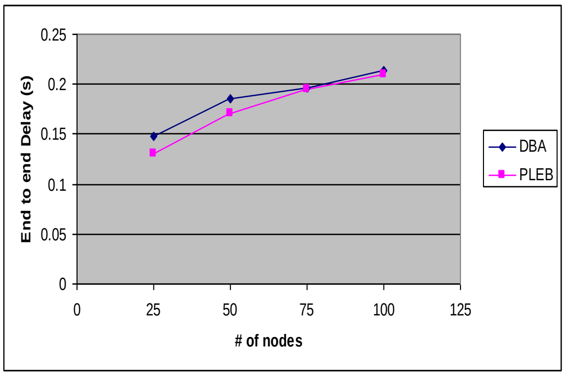

Figure 10: End to end delay vs. node density with node speed 10 m/s.

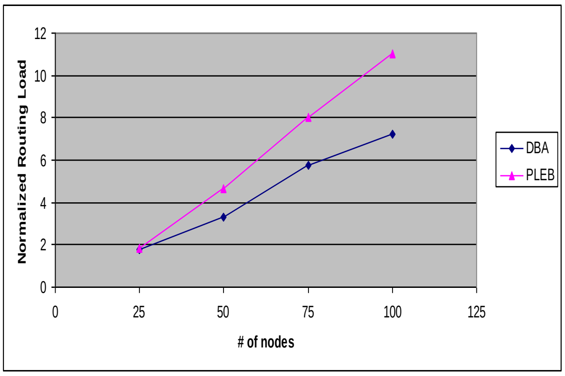

Figure 11: Normalized routing load vs. node density with node speed 10 m/s.

In Fig. 9, we can see that the delivery ratio increases when the number of nodes increases. Again for sparse networks the difference is very small, whereas the difference increases for dense networks.

For example, for the case of 100 nodes, DBA improves packet delivery ratio by around 2%. Same reasons justify this difference since more and more retransmission trials needed with more nodes in the network.

Fig. 10 represents the relation between end to end delay and network density. According to the results, PLEB outperforms DBA at node density 25 and 50 by 2% and 1% respectively, but at dense network end to end delay is nearly the same in both algorithms. This is due to that in DBA, the number of increments produced by exponential factor which leads to generate a longer backoff values is greater than PLEB. These longer backoff timers lead to longer network delay. Also, slow decrement behavior in DBA causes the network to have longer redundant waiting times and leaves CW sizes larger than needed which produce unnecessary delay.

Fig. 11 depicts the effect of network density on normalized routing load. The normalized routing load increases as the number of nodes increases. Both PLEB and DBA increased in the same behavior. DBA outperforms PLEB by 23% at all node density. We notice that at sparse network of 25 nodes, the normalized routing load at two algorithms is nearly the same because the contention between nodes is low so the invocation of backoff algorithm is low in this case. But at medium and dense networks the ratio of distance between the two lines is obviously large due to high contention between nodes since the invocation of backoff algorithms is increased.

The control overhead increases as a number of nodes increases due to collisions and broken links. Thus, more RTS packets for sensing activities are needed and the invocation of backoff algorithm is increased.

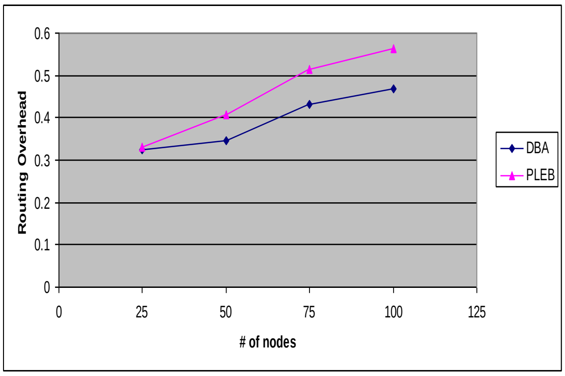

Figure 12: Routing Overhead vs. node density with node speed 10 m/s.

Fig. 12 demonstrates the effect of network density on the routing overhead. We evaluated the two algorithms by using the same routing protocol which is AODV that uses blind flooding technique where each node rebroadcasts the RREQ packets it receives to all its neighbors and so on, which degrades the performance of the network because it causes a huge routing load overhead. The routing overhead in DBA is reduced by 12% at all node density due to that the total amount of data packets sent to the MAC layer is reduced comparing to it in PLEB.

Nodes in MANET have a distributed nature, due to this feature; the packet collisions are not completely eliminated. So senders of the collided packets should defer for a period of time to avoid repeated collisions.

As can be seen in Fig. 13, when the number of nodes increases, the contention among nodes increases to gain access to the shared channel so the collision is highly exists. DBA reduces the possibility of collisions regards to PLEB by 41%. This is because of reducing the repeated attempts to access the shared channel due to its exponential increments in two phases that generate a longer backoff times.

The number of broken links for different number of nodes is shown in Fig. 14. In DBA, the number of broken links is reduced to 41% at dense network comparing to PLEB this is results from the decrease of the number of control packets (control overhead).

5.3 Effect of Network Mobility

In order to simulate the backoff algorithms more effectively and to reflect some scenarios in realistic world that varies from running human speed to fast car speed, the maximum speed of 100 nodes varies from 5 to 30 m/s and the pause time is set to 0 in order to evaluate the performance at high mobility scenario. The packet rate is 4 and the CBR connections are 30.

Fig. 15 and Fig. 16 present throughput and packet delivery ratio vs. speed respectively. As we can see there is a closer relation between the two metrics, within increasing the speed, the throughput and packet delivery ratio are decreased in both schemes. DBA increases the throughput and packet delivery ratio by up to nearly 3%, 2% comparing to PLEB. Additionally, as the speed of mobile nodes increases, PDR in both algorithms decreases because of increasing packet loss which results from increasing the link breakages due to the mobility of nodes.

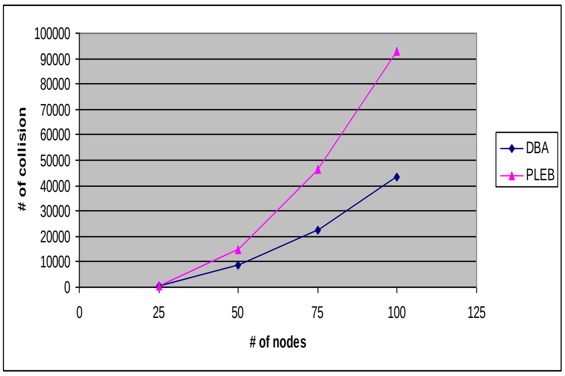

Figure 13: number of collision vs. node density with node speed 10 m/s.

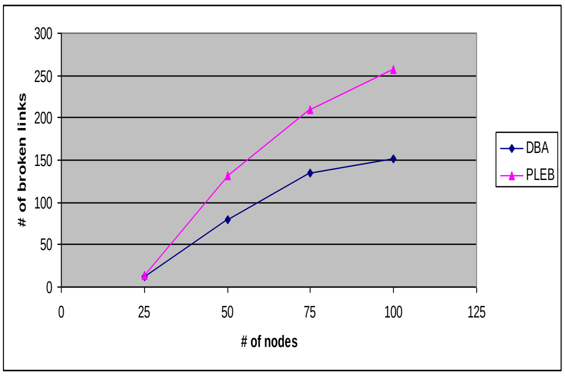

Figure 14: Broken links vs. node density with node speed 10 m/s.

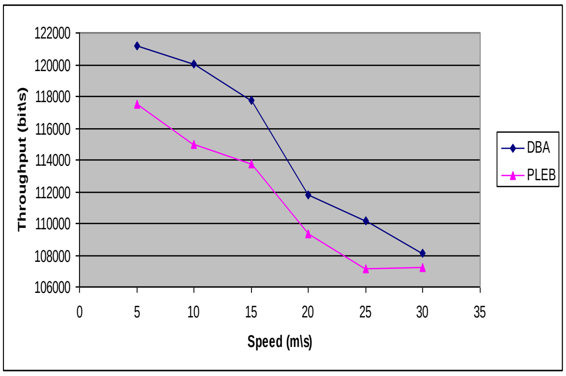

Figure 15: Throughput vs. speed at 4 packets/s with 30 CBR connections.

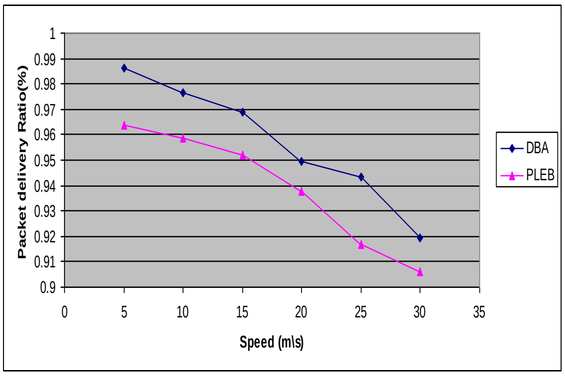

Figure 16: Packet delivery ratio vs. speed at 4 packets/s with 30 CBR connections.

As can be seen in Fig. 17, PLEB gives a slight end to end delay improvement comparing to DBA when speed of nodes (<=15 m\s) but at high speed nodes (>15 m\s), the improvement becomes more obvious. This is return to the following two reasons. The first one related to that the number of increments produced by exponential factor in DBA which leads to generate a longer backoff values is greater than in PLEB. These longer backoff timers lead to longer network delay. The second reason related to the action taken upon successful transmission. In DBA, slow decrement behavior performed by linear method causes the network to have longer redundant waiting times and leaves CW sizes larger than needed which produce unnecessary delay where as the rapid resetting process in PLEB doesn't generate a noticeable delay in case of successful transmission.

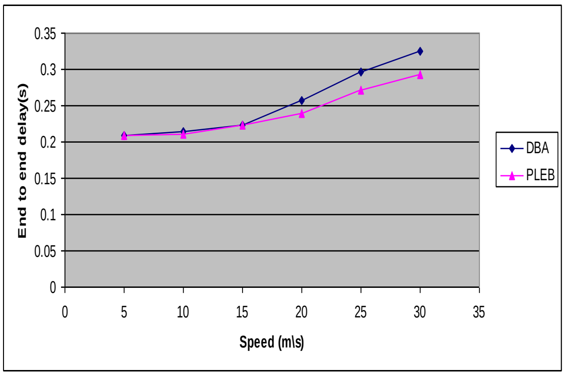

Figure 17: End to end delay vs. speed at 4 packets/s with 30 CBR connections.

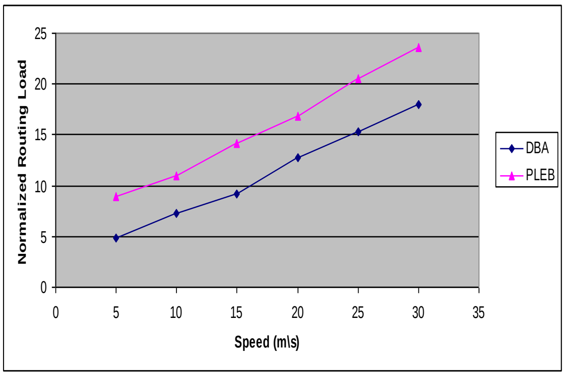

Figure 18: Normalized Routing Load vs. speed at 4 packets/s with 30 CBR connections.

Fig. 18 shows the normalized routing load versus speed. We notice that the normalized routing load increases with increasing node speed in both algorithms because the sender nodes will generate more routing load to sense the status of the channel. But the number of sensing in DBA is less than in PLEB due to its two phases of increment methods.

Fig. 19 shows the performance of the two algorithms in terms of routing overhead. For all speeds, DBA reduces the overhead to 14% because it reduces the number of attempts to access the channel and thus reduces the need to retransmit RTS packets for sensing activities.

Fig. 20 and Fig. 21 depict the effect of node speed on the number of collisions and broken links respectively. According to the results, the number of collisions and broken links by using DBA are reduced to 56%, 34%. The number of collisions is reduced in DBA because of the decreasing in percentage of repeated attempts to access the shared channel due to exponential increments in two phases. In addition, Number of broken links increased with increasing node speed because when we have a long backoff value, this leads to that nodes may leave outside the transmission range before attempt to retransmission again which increased the broken links possibility.

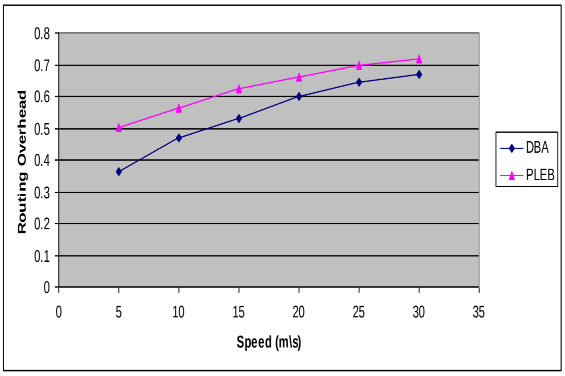

Figure 19: Routing Overhead vs. speed at 4 packets/s with 30 CBR connections.

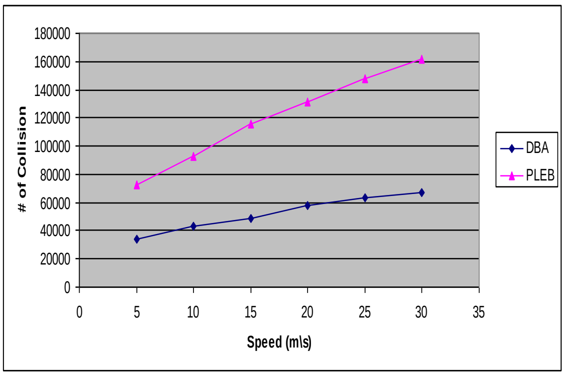

Figure 20: Number of collision vs. speed at 4 packets/s with 30 CBR connections.

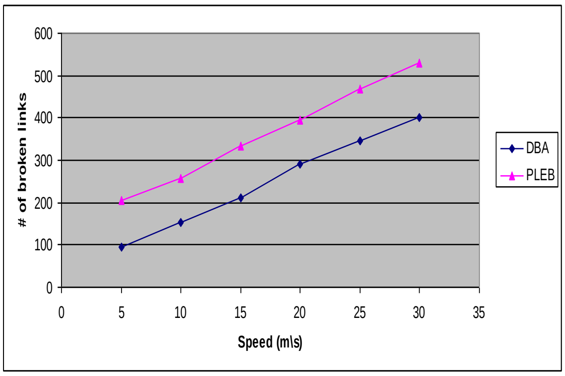

Figure 21: Number of broken links vs. speed at 4 packets/s with 30 CBR connections.

6. CONCLUSION and FUTURE WORK

This research proposed a new backoff algorithm and studied the effect of network mobility by using different values and apply it in different criteria (e.g number of increments) to select suitable thresholds to be used in our proposed algorithm and compared them with PLEB. The results show that when thresholds are set to K= 1.5, T= 5, U= 1.5, V= 5, Y= 2, best results were obrained as compared to other possible values in terms of throughput, packet delivery ratio and delay which is approximately the same compared to PLEB.

After that, Performance evaluation is done by studying the effect of network density and network mobility by using our predetermined thresholds.

The simulation results show that our proposed algorithm outperforms PLEB, one of the recent and most popular backoff algorithms, in terms of several performance metrics with keeping end to end delay approximately the same. On average, DBA improves the throughput, packet delivery ratio, normalized routing load, routing overhead, collision and broken links by 3%, 1%, 23%, 12%, 31%, 31% respectively at all nodes density and by 2%, 3%, 31%, 14%, 56%, 35% at all nodes speed compared to PLEB.

The dynamic backoff algorithm has been studied using the Random Waypoint Mobility Model (RWMM). In this model, nodes move at a speed uniformly distributed in the interval [MIN SPEED, MAX SPEED]. So each node has a fixed speed at all simulation time which not reflect the nature of some nodes that moves with different speeds in real world, so it is useful to evaluate our proposed algorithm using other mobility models such as random direction model.

In our study we set the pause time to 0 in order to study a high mobility scenario. So more research can be conducted to study the effect of changing the pause time and study its impact on the performance.

The Constant Bit Rate (CBR) traffic generator type is used. Future work can study the effect of using other traffic types like Variable Bit Rate (VBR).

In order to verify the results in simulation, analytical model can be developed such as markovian model.

As a first response to a failure or collision, our proposed algorithm uses the exponential method to increase the CW then linear method and apply this manner again. So it is applied in two phases. Future work can study the effect of reverse order on the selected methods, so as the first response of failure; the new algorithm can use a linear method to increase the CW then exponential method and apply this manner again till reach the maximum value of CW. Additionally, future work can study the effect of more than 2 phases. Moreover, it is useful to study the effect of other increment methods and apply it more than one times in order to merge advantages of them.

REFERENCES

[1] Wu, H. and Pan, Y., "*Medium Access Control in Wireless Networks (Wireless Networks and Mobile Computing)*", Nova Science Publishers, 2008.

[2] Schiller, J., "*Mobile Communication*", 2nd Edition, Addison Wesley, 2003.

[3] Nieminen, K., "*Introduction to Ad Hoc Networking*", Networking Laboratory Helsinki University of Technology, pp 1-10, 2003.

[4] Camp, T., Boleng, J., and Davies, V., "*A survey of mobility models for ad hoc network research*", Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, Vol. 2 , no. 5, pp 483–502, 2002.

[5] Liang, H. M., Zeadally, S., Chilamkurti, N. and Shieh, C. K., "*Optimizing Channel Access in Wireless Local Area Network Environments with a New Backoff Approach*", International Journal of Multimedia and Ubiquitous Engineering Vol. 4, No. 2, pp 35-48, 2009.

[6] Kumar, S., Raghavan, V., Deng, J., "*Medium Access Control protocols for ad hoc wireless networks: a Survey*", Science direct, Ad Hoc Networks, Vol. 4, No. 3, pp 326-358, 2006.

[7] Tariq, S., "*MAC Algorithms in Wireless Networks – Applications, Issues and Comapisons*", Master's Thesis, Umea University, Sweden, pp 1-61, 2005.

[8] Song, N.O., Kwak, B.J., Song, J. and Miller, L.,"*Enhancement of IEEE 802.11 distributed coordination function with exponential increase exponential decrease backoff algorithm*", In Proceedings of VTC 2003-spring, vol. 4, Jeju, Korea, pp 2775-2778, 2003.

[9] Yun, L., Zhao, W.L., Long K.P. and Chen Q.B., "*A New Backoff Algorithm to Improve the Performance of IEEE 802.11 DCF*", Springer-Verlag Berlin Heidelberg, pp 488-497, 2005.

[10] Manaseer, S., Ould-Khaoua, M. and Mackenzie, L.,"*Fibonacci Backoff Algorithm for Mobile Ad Hoc*

Networks", DCS Technical Report Series Dept of Computing Science, University of Glasgow, pp 1-6, 2006.

[11] Berqia, A. and Angoma, B., "*Fairness and QOS in Ad-Hoc Networks*", Vehicular Technology Conference (VTC Spring 2008), IEEE, pp 16-20, 2008.

[12] Ibrahim, M. and Alouf, S., "*Design and Analysis of an Adaptive Backoff Algorithm for IEEE 802.11 DCF Mechanism*", IFIP International Federation for Information Processing, pp 184-196, 2006.

[13] Deng, J. and Hass, Z.J, "*On Optimizing the Backoff Interval for Random Access Schemes*", IEEE Transactions on Communications, volume 51, issue 12, pp 2081- 2090, 2003.

[14] Deng, J., Varshney, P., and Haas, Z., "*A new backoff algorithm for the IEEE 802.11 distributed coordination function*" , In Communication Networks and Distributed Systems Modeling and Simulation (CNDS '04), pp 1-6,2004.

[15] Choi, N., Seok, Y., Choi, Y., Kim, S. and Jung, H., "*P-DCF: Enhanced Backoff Scheme for the IEEE 802.11 DCF*",IEEE VTC 2005-Spring, Stockholm, Sweden, pp 1-4, 2005.

[16] Manaseer, S., Ould-Khaoua, M. and Mackenzie, L.," *On a Modified Backoff Algorithm for MAC Protocol in MANETs*", International Journal of Business Data Communications and Networking, Vol. 5(1), pp. 60-73, 2009.

[17] Manaseer, S. and Masadeh, M., "*Pessimistic Backoff for Mobile Ad hoc Networks* ", IEEE, In Proceeding of the 4th International Conference on Information Technology ICIT, Jordan, pp 1-10, 2009.

[18] Krishna, C., Chakrabarti, S. and Datta, D., "*Impact of Contention Resolution Algorithms on the Performance of IEEE 802.11 DCF-based MAC Protocol in a Mobile Ad hoc Network*", In Proceedings of 11th National conference on communication (NCC- 2005), IIT, Kharagpur, pp 78-81, 2005.

[19] Bharghavan, V., Demers, A., Shenker, S. and Zhang, L., "*MACAW: A media access protocol for wireless LAN's*", In Proceedings of the ACM SIGCOMM, London, pp: 212-225, 1994.

[20] Perkins, C. E., Royer, E. M. and Das, S. R. "*Ad Hoc On Demand Distance Vector (AODV) Routing,*", IETF Internet Draft, draftietf-manet-aodv-08.txt, pp 1-11, 2001.

[21] Ki, H., Choi, S., Chung, M., and Lee, T.," *Performance Evaluation of Binary Negative-Exponential Backoff Algorithm in IEEE 802.11 WLAN",* Sungkyunkwan University, Springer-Verlag Berlin Heidelberg, pp 294–303, 2006.

[22] Krishna, C., Chakrabarti, S., Datta, D., *"A modified backoff algorithm for IEEE 802.11 DCF-Based MAC protocol in a mobile ad hoc network",* IEEE TENCON 2004, vol. B, vol. 2, pp 664- 667, 2004.

[23] Chlamtac, I., Conti, M. and Liu, J., "*Mobile ad hoc networking: imperatives and challenges*", Science Direct, pp 13-64, 2003.

[24] Manaseer, S., "*On Backoff Mechanisms for Wireless Mobile Ad Hoc Networks*", PhD Thesis, University of Glasgow, pp 1 -156, 2009.

[25] Zeng, X., Bagrodia, R. and Gerla, M., "*Glomosim: A library for parallel simulation of large-scale wireless networks*". In Proceedings of the 12th Workshop on Parallel and Distributed Simulation, 26.29, pp 154-161.

[26] GloMoSim: Global Mobile Information Systems Simulation Library. http://pcl.cs.ucla.edu/projects/glomosim/.

[27] Gerla, M., Bajaj, L., Takai, M., Ahuja, R., and Bagrodia, R., "*GloMoSim: A Scalable Network Simulation Environment*". Technical Report 990027, University of California, 13, 1999.

[28] WU, C., FENG, J., and FAN, P., "*On a New Queue Backoff Fair Algorithm for Ad Hoc Networks*", Parallel and Distributed Computing, Applications and Technologies. PDCAT'2003. In Proceedings of the Fourth International Conference, pp 335-339, 2003.

[29] Nasir, Q., Albalt, M., "*History Based Adaptive Backoff (HBAB) IEEE 802.11 MAC Protocol*", Communication Networks and Services Research Conference (CNSR 2008), pp 533-538, 2008.

[30] Pucha, H., Das, S. and Hu, Y., " *The performance impact of traffic patterns on routing protocols in mobile ad hoc networks*", Science Direct, Elsevier, pp 3595–3616, 2007.