

Intelligent Paging Backoff Algorithm for IEEE 802.11 MAC Protocol

Ahmad Momani¹, Muneer Bani Yassein², Omar Darwish³, Saher Manaseer⁴, Wail Mardini⁵

^{1,2,3,5}Dept. of Computer Science, Jordan University of Science and Technology

P.O. Box 3030, 22110, Irbid (Jordan)

⁴Dept. of Computer Science, the University of Jordan

P.O. Box 11942, Amman (Jordan)

E-mail: ¹aamomani6@just.edu.jo, ²masadeh@just.edu.jo, ³oadarwish@just.edu.jo,
⁴saher@ju.edu.jo, ⁵mardini@just.edu.jo

Received: December 14, 2011 Accepted: February 21, 2012 Published: June 19, 2012

DOI: 10.1154/npa.v4i2.1297

URL: <http://dx.doi.org/10.1154/npa.v4i2.1297>

Abstract

Backoff algorithms have been introduced to improve network performance by reducing collisions over networks, especially in large ones. The Binary Exponential Backoff used in the standard IEEE 802.11 Medium Access Control makes exponential increments to contention window sizes. The large contention window gaps produced can significantly reduce the network performance. Therefore, a new increment behavior is introduced in this paper in order to avoid transmissions failure. The Results show that shifting the contention window to the right place based on previous transmissions outperforms the Binary Exponential Backoff Algorithm, the Smart Adaptive Backoff Algorithm and Pessimistic Linear Exponential Backoff Algorithm, up to 7.1%, 16.5% and 19% respectively in term of data delivery ratio.

Keywords: Backoff algorithms, Binary Exponential Backoff, Contention window, Medium Access Control, Failure transmissions.

1. Introduction

Back of algorithms is one of the most important protocols which lie in the Medium Access Control layer (MAC). The main goal behind the Back of algorithms is to avoid packet collisions when the medium is busy. The IEEE 802.11[1] standard is one of the most popular protocols which introduce main services for MAC layer. The IEEE 802.11 standard depends on the Idea of choosing a random back of time according to particular interval values. The maximum value in this interval is defined as Contention Window (CW). Contention window (CW) is set to be the core of this research. The IEEE 802.11 standard increases the CW in the case of medium busy in order to avoid packet collisions using the Binary Exponential Backoff algorithm (BEB) which depends on the binary exponential equation. The Binary Exponential Backoff algorithm and many other Algorithm such as the Logarithmic Backoff Algorithm [2], Smart Adaptive Backoff Algorithm [3] and Pessimistic Linear Exponential Backoff Algorithm [4] focused on the idea of how to increase the value of CW without taking in consideration the value of the CW in the previous transmissions. In this research the Intelligent Paging Backoff Algorithm depends on the value of the CW in the previous transmissions in order to avoid packet collisions. This research focuses on the Mobile ad-hoc network (MANETs) because of the widespread of such networks in the present.

MANETs is a set of unstructured wireless nodes that communicate with each other in a non stable environment. [5].

A MANET has special proprieties which makes it distinguish than other network types. These proprieties are summarized by the following main points [6]:

1. The dual role of the node: each node participates in MANETs has to play the role of the host and the role of router at the same time.
2. Limited power resources: nodes in MANETs depend on limited battery energy, since these nodes are suffering of frequently failure in MANETs which add another challenge for such type of networks.
3. The environment topology can rapidly change over time: this property considered the main challenge for such type of networks. The frequent movements of the nodes in the network leads to more broken routes between the source and the destination pair which leads to more dropped packets in the MANETs.

MANETs are deployed in many practical fields such as military applications, emergency operations, mobile voting [7].

1. Military operations: a MANET can be used in battlefields to establish a reliable communication between military units such as vehicles and solders.
2. Emergency operations: a MANET can be useful in places where the infrastructure based communications are destroyed due to different circumstances such as supernatural situations. This is true since MANETs are very easy to be deployed in a fast and non expensive cost.
3. Mobile voting: a MANET can be used in mobile voting applications in order to

enhance a voting process by merging time and location into decision making process. For example, the proposed application in [8] which was designed to distribute decision making.

MANETs also have many challenges due to mobility of nodes. These challenges can significantly reduce the network performance. Examples of such challenges are [9, 10]:

1. **Constrained resources:** in MANETs nodes suffer from limited resources mainly the channel bandwidth and the power availability. To explain this, Nodes must achieve the maximum data packet size sent successfully. While on the other hand, they must ensure the best performance with the lowest rates of power consumption in order to maintain batteries for the longest life time.
2. **Transmission range:** in MANETs the communication between nodes is bounded by transmission range capability. This means if two nodes need to communicate at a time, they must be within the transmission range of each other. Otherwise, there should be another party to support this process (i.e. another node to connect the source/destination nodes).

This research paper is structured as follows. Section two discusses related work. Section three presents the simulation model and the parameters used in the experiments. Section four concludes the results.

2. Related Work

F. Hong-yu, X. Lei and L.Xiao-hui proposed the Logarithmic Backoff Algorithm of MAC Protocol in Ad Hoc Networks. The main contribution in their work is to increase the CW using the logarithmic equation depends on the number of the nodes in the network [11].

N.Sumathi, A. Thanamani proposed the Implicit Pipelined Backoff Algorithm (IPBA) [R]. Their algorithm depends on applying the pipelined concept for the backoff procedure when the medium is busy, remaining nodes in the network initiate the contention procedure in parallel way for the next packet transmission. [12]

M. Masadeh, S. Manaseer and A. Momani [3] proposed the Smart Adaptive Backoff Algorithm (SABA). This algorithm attempts to achieve the best performance by combining the exponential, logarithmic and linear increments in the attempt to achieve the most adequate increment behavior. It also utilizes the recent history of CW sizes for successful transmissions only. This history based algorithm works the best in small and large network sizes.

S. Manaseer and M. Masadeh [4] proposed the Pessimistic Linear Exponential Backoff (PLEB). This algorithm combines the exponential and linear increment behaviors. In the first stages of the algorithm and after a transmission failure, the exponential increment is used until the CW reaches the threshold N. after that it continues with linear increments. PLEB works the best when implemented in large network sizes with a high mobility speeds.

S. Manaseer, M. Ould-Khaoua and L. Mackenzie [2] proposed Fibonacci Increment Backoff (FIB) algorithm. This algorithm reduces the differences between CW sizes based on following Fibonacci series formula:

$$f(n) = f(n - 1) + f(n - 2). \quad f(0) = 0, f(1) = 1, n \geq 0$$

[2] Shows that FIB algorithm achieves higher network throughput than the standard BEB used in the IEEE 802.11 MAC protocol.

K. Takahashi and T. Tsuboi proposed an algorithm that starts the backoff timer based on the binary exponential backoff approach, and then they use finishing tags to control the process. Each node in the network should increase its backoff timer by fixed number X after hearing a node with higher finish tag than the others nodes. Their algorithm shows a highest enhancement in terms of the saturation throughput [13].

M. Bani Yassein, S. Manaseer and A. Al-Turani study the difference between the two backoff algorithms, Fibonacci increment backoff (FIB) and the Pessimistic Linear exponential Backoff (PLEB), using different rebroadcast probabilities in the AODV protocol [14].

3. Backoff Algorithms

3.1 Overview

In general, backoff algorithms were introduced in order to decrease collisions over networks as much as possible. They tend to increase the CW size on each transmission failure. This makes the probability of having a small backoff timers decreases. Therefore, the sender nodes shall wait more period of time before transmitting. This mechanism provides the current sending nodes more time space to finish their jobs.

Now, to gain the best performance of a network, a suitable increment behavior should be used. That is, large and small increments ever should be avoided. This is because the large increments lead to a large CW gaps, and so a huge waste of different network resources without doing anything. On the other hand, small increments do not allow adequate backoff time before data retransmission.

The standard IEEE 802.11 MAC layer uses the Binary Exponential Backoff (BEB) algorithm in order to handle collisions of the network. The BEB algorithm works by increasing the CW sizes exponentially based on transmission failure. When a node has a packet to transmit, the node starts sensing the channel and if it is founded to be idle, the node starts transmitting immediately the data packets to the specified destination. Otherwise, the backoff mechanism is triggered. To say more, a backoff timer is chosen randomly among the current CW size. This timer is decremented only at idle slot times. When the count-down timer reaches zero, the node starts sending the data packets despite network status. Now, if the acknowledgement received (successful transmission) from the destination node, the CW size is reset to minimum. On the other hand, if the acknowledgement is lost the CW size is

incremented exponentially [3, 15, 16, 17].

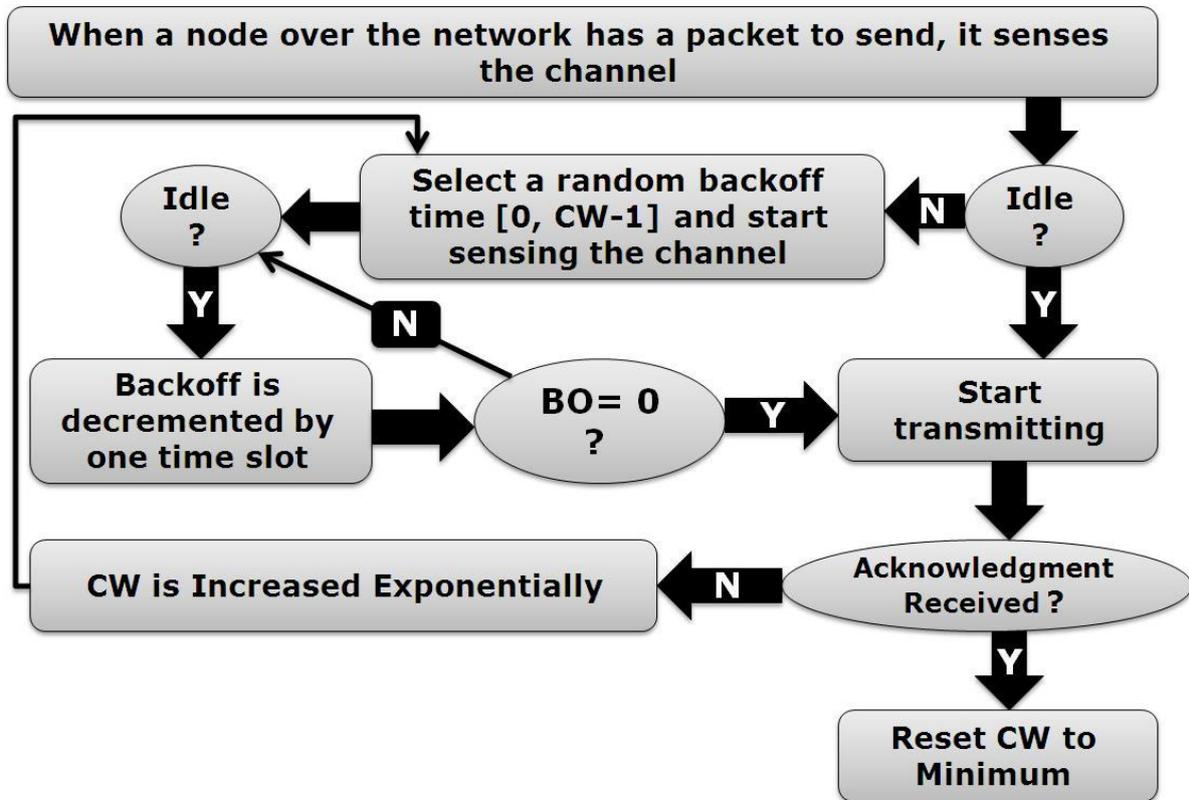


Figure 1: BEB algorithm description [3]

The Pessimistic Linear Exponential Backoff (PLEB) is one of the latest backoff algorithms presented in December of 2009. This algorithm uses two different increment behaviors: exponential and linear increments. PLEB assumes that congestion in the network will not be resolved in the near future. Thus, PLEB selects exponential increments at the first stages of the algorithm and continues to linear behavior after a threshold N [2].

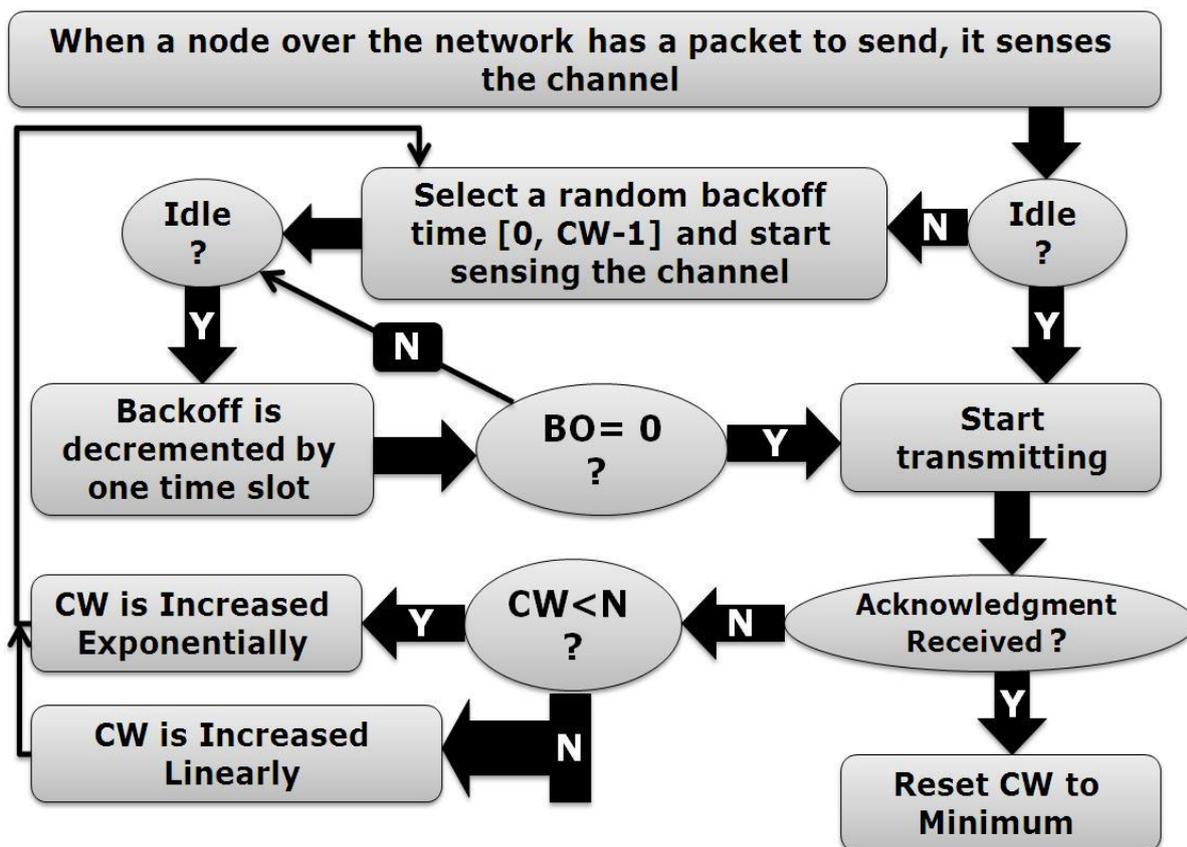


Figure 2: PLEB algorithm description [3]

Another modern backoff algorithm studied in this paper is the Smart Adaptive Backoff Algorithm (SABA). This algorithm works by incrementing the CW exponentially only for a finite number of times (until the history is filled). After that, it calculates the average of the history in order to choose a new increment behavior based on the average value. If the average is less than a threshold N , the increment behavior is logarithmic. Otherwise it will be the linear increment [3].

3.2 The New Proposed Backoff Algorithm

The Intelligent Paging Backoff Algorithm (IPBA) is a new backoff algorithm which aims to slide the contention window value in order to select adequate backoff timer. IPBA bounds the backoff timer value between the last CW size and the new one after increment. Moreover, IPBA uses the quintuplet increments if there is a transmission failure. Although this increment behavior increases the overall end to end packet delay, it increases the data packet delivery ratio significantly. This is normal since this algorithm prevents selecting small backoff values because the CW value is bounded (minimum value is equal to the last contention window value when transmission failure CW_f), unlike the BEB algorithm which allows to select a value between 0 and $CW-1$.

Figure 3 shows the basic functionality of IPBA algorithm. As shown, the first 5 lines of

the algorithm set the initial value for the backoff timer. And then, starts the decrement of that time based on idle time slots. This means, the timer will be decremented only at idle time slots. Otherwise, it freezes. After the timer reaches zero, the data packet is transmitted. Now, in case of successful transmission the CW, CW_f values are reset to minimum. Otherwise, the backoff mechanism will start. The lines 7-10 show a brief description of the sliding process in IPBA algorithm. It starts with line 7 by saving the current CW value in CW_f before creating a new one (line 8). Now, the lines 9-10 shows the sliding process for the CW value, and how the backoff timer will a number between the values CW_f and CW.

```
1  Set BO to initial value
2  While Bo ≠ 0 do
3  For each time slot
4      If channel is idle then BO=BO -1
5          Wait for a period of DIFS then Send
6      If (Send-Failure) then
7          Save the  $CW_f = CW_{current} / 2$ 
8           $CW_{new} = CW_{current} * 5$ 
9           $Y = CW - CW_f$ 
10          $Backoff-Timer = Random\ x\ \% Y + CW_f; \quad 1 \leq x \leq CW - 1$ 
11     Else
12         Reset CW,  $CW_f$  to minimum, minimum / 2, respectively
13     Go to line number 1
14     Stop
```

Figure 3: The Intelligent Paging Backoff Algorithm

Figure 4 represents a brief description of IPBA algorithm. It shows the essential modifications made to the BEB in the standard IEEE 802.11 MAC protocol. X represents the last contention window value before the increment process.

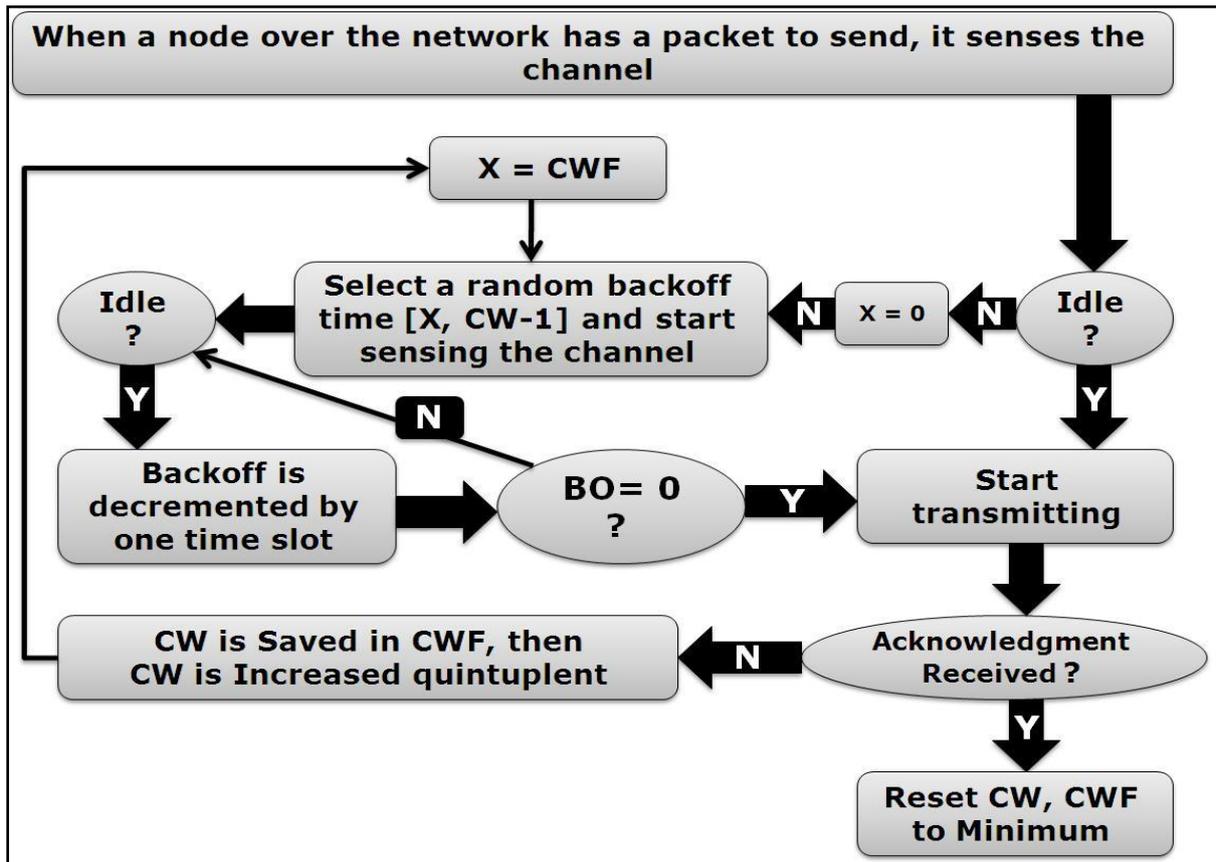


Figure 4: IPBA algorithm description

4. Simulation Results

In this research The Global Mobile Information System Simulator (GloMoSim) are used. GloMoSim is a scalable simulation environment for wireless networks. GloMoSim is a sequential and parallel simulator which designed as a set of modules. These modules were developed using PARASEC, a C-based simulation language. GloMoSim was designed using a layered approach [18].

The GloMoSim simulator consists of the following layers:

- The radio layer.
- The medium access layer.
- The network layer.
- The transport layer.
- The application layer.

GloMoSim has many parameters which their values can be provided using the configuration file CONFIG.IN. Moreover, the simulation results are stored in an output file GLOMO.STAT.

In addition, the simulator uses an input file APP.CONF that specifies the number of sources, destinations, and the amount of data packets to send to destinations.

The simulator uses an appropriate model for each layer. We used the CBR model for the application layer, the UDP model for the transport layer, the IEEE 802.11 model for the MAC layer and we used the AODV (Ad hoc On-demand Distance Vector protocol) for the network layer.

The simulator reads the parameters from the configuration file at the start of the simulation and begins the Simulation process. At every instance of time, it saves the different information at some data structures and at the end of the simulation process, it gathers the full information from the data structures and saves this information in an output file called GLOMO.STAT. From this file, we can derive the different conclusions such as number of control packets, end-to-end delay, and the delivery ratio.

The following table shows the values for the environment parameters:

Table 1. Simulation Parameters

Parameter	Value
Number of nodes	30, 60, 100
Medium Bandwidth	2 Mb/s
Simulation time	900 seconds
Application Model	(CBR)
Node speed	50, 100, 150 m/s
Traffic Load	10 packets/sec.
Network Dimensions	1000 * 1000

The following figures show the delivery ratio, end-to-end delay and network overhead of The Intelligent Paging Backoff Algorithm, BEB, PLEB and SABA when the number of nodes is varied by 30, 60 and 100 nodes respectively. Also, the traffic load used in the simulation is 10 packets per second. Moreover, the nodes speed is also varied by 50 m/s, 100 m/s, and 150 m/s respectively. The extensive experiments show that IPBA outperform each of BEB, PLEB and SABA algorithm significantly in terms of delivery ratio. And also it is still competing algorithm in terms of end to end delay and network overhead.

4.1 Data Delivery Ratio

The figures (5, 6, 7) in this subsection show that the ratio of delivering data packets to the destination nodes decreases for all applied algorithms when the mobility speed increases. This result can be justified since the increasing of node mobility leads to more broken routes. Moreover, the figures show that IPBA has the best data delivery ratio overall, especially in medium network sizes.

Figure 5 shows that IPBA's delivery ratio outperform each of BEB, SABA and PLEB algorithms by approximately 2%, 8%,19% respectively for more stable areas (50 m/s) and 3%,3%,8.5% respectively in dynamic areas (150 m/s) within a sparse environment (30 nodes) .

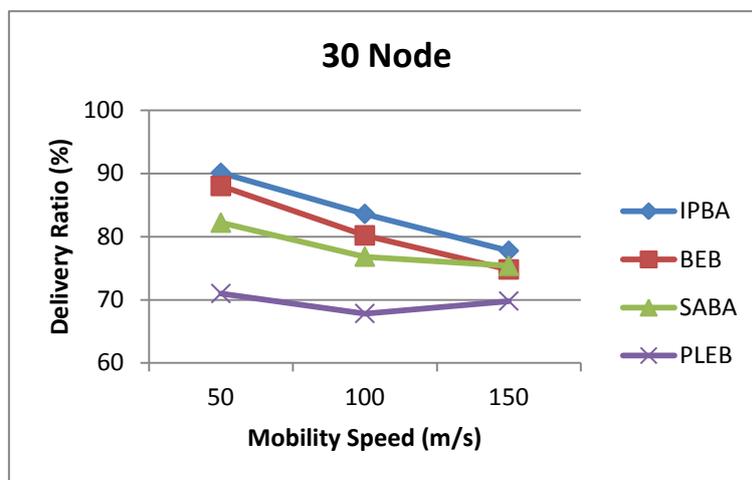


Figure 5: Delivery ratio of IPBA, BEB, PLEB and SABA when the number of nodes is 30 and traffic load is 10 Pack/Sec.

Figure 6 shows that IPBA's delivery ratio outperform each of BEB, SABA and PLEP algorithm's delivery ratios by approximately 6 %, 7%,17.5% respectively for more stable areas (50 m/s) and 7%,12%,17% respectively for dynamic areas (150 m/s) within a mid-between sparse and dense- environment (60 nodes) .

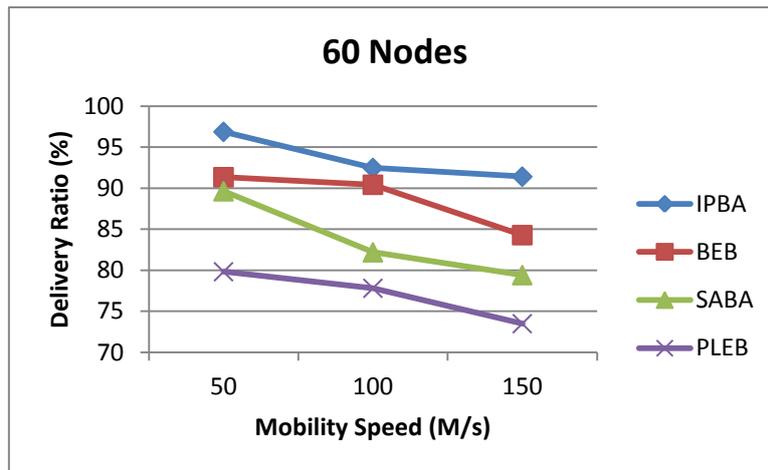


Figure 6: Delivery ratio of IPBA, BEB, PLEB and SABA when the number of nodes is 60 and traffic load is 10 Pack/Sec.

Figure 7 shows that IPBA's delivery ratio outperform each of BEB, SABA and PLEB algorithms delivery ratio by approximately 2% ,10% ,17% respectively for more stable areas (50 m/s) and 2%,9%,17% respectively for dynamic areas (150 m/s) within a dense environment (100 nodes) .

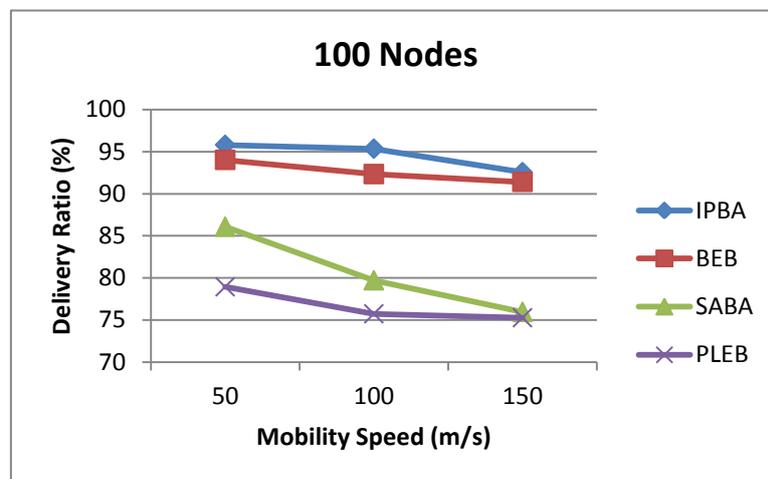


Figure 7: Delivery ratio of IPBA, BEB, PLEB and SABA when the number of nodes is 100 and traffic load is 10 Pack/Sec.

4.2 End-to-End Delay

The figures (8, 9, 10) show the performance metric end-to-end delay. This metric show how long it takes for a packet to be transferred from the source node to the application layer of the destination node.

Figure 8 shows that IPBA's end to end delay outperform SABA's end to end delay by

approximately 1 second for more stable areas (50 m/s) and 2 seconds for dynamic areas (150 m/s) within a sparse environment (30 nodes) .

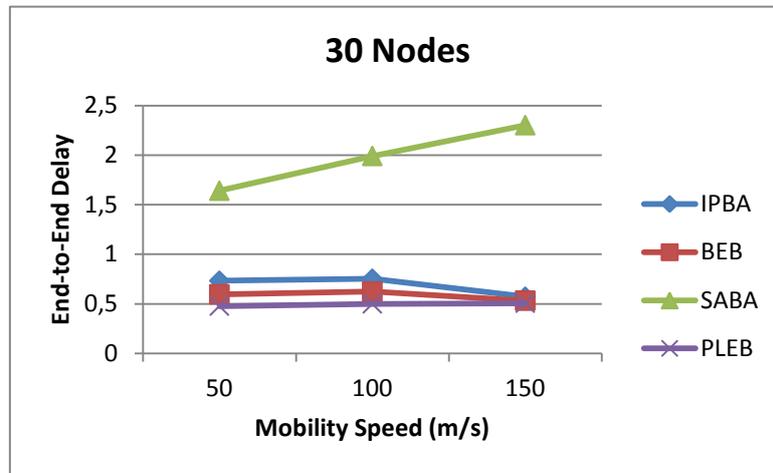


Figure 8: End To End Delay of IPBA, BEB, PLEB and SABA when the number of nodes is 30 and traffic load is 10 Pact/Sec.

Figure 9 shows that IPBA's end to end delay outperform SABA's and PLEP's end to end delay by approximately 1.2, 0.3 second respectively for more stable areas (50 m/s) and 2.7, 0.5 seconds respectively for dynamic areas (150 m/s) within a mid -between sparse and dense- environment (60 nodes), While each of IPBA and BEB algorithms behave approximately the same with a simple preference for IPBA exactly during dynamic areas.

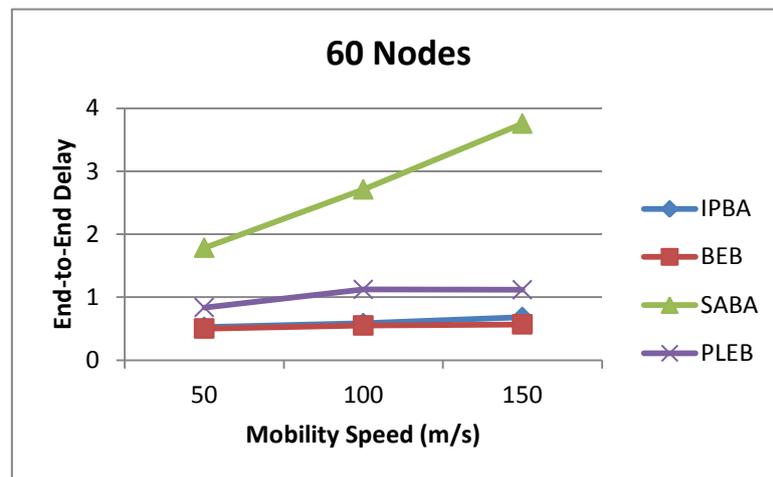


Figure 9: End To End Delay of IPBA, BEB, PLEB and SABA when the number of nodes is 60 and traffic load is 10 Pact/Sec.

Figure 10 shows that IPBA's end to end delay outperform SABA's and PLEP's end to end delay by approximately 1, 1.4 second respectively for more stable areas (50 m/s) and 3, 1.25, for dynamic areas (150 m/s) within a dense environment (100 nodes), While each of IPBA and BEB algorithms behave approximately the same with a simple preference for IPBA in

dynamic areas.

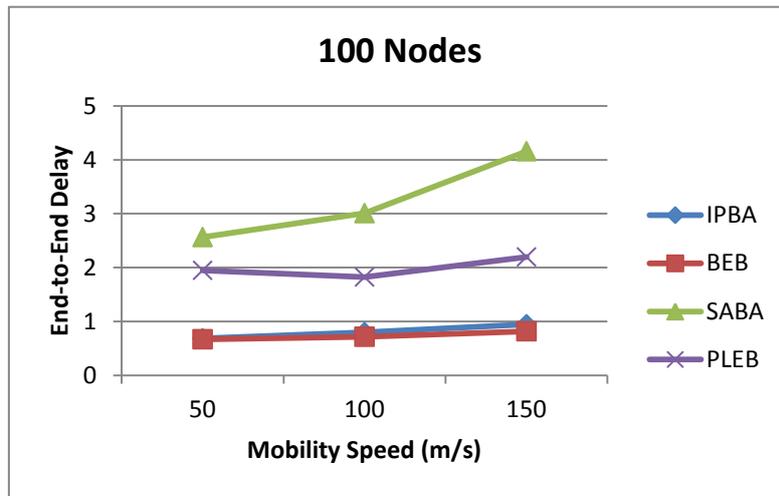


Figure 10: End To End Delay of IPBA, BEB, PLEB and SABA when the number of nodes is 100 and traffic load is 10 Pact/Sec.

4.3 Network Overhead

The rest of figures (11, 12, 13) show another performance metric called the network overhead, this metric measures the number of packets carrying control messages (route request, route reply, and route error) for route discovery and routing to the number of packets carrying data [15].

Figure 11 shows that IPBA's overhead outperform SABA's and PLEP's overhead by approximately 0.1%, 0.9 for more stable areas (50 m/s) and 0.15%, 1.2% for dynamic areas (150 m/s) within a sparse environment (60 nodes), While each of IPBA and BEB algorithms behave approximately the same in the dynamic (150) and staple environments but in med environment speed BEB defeat IPBA.

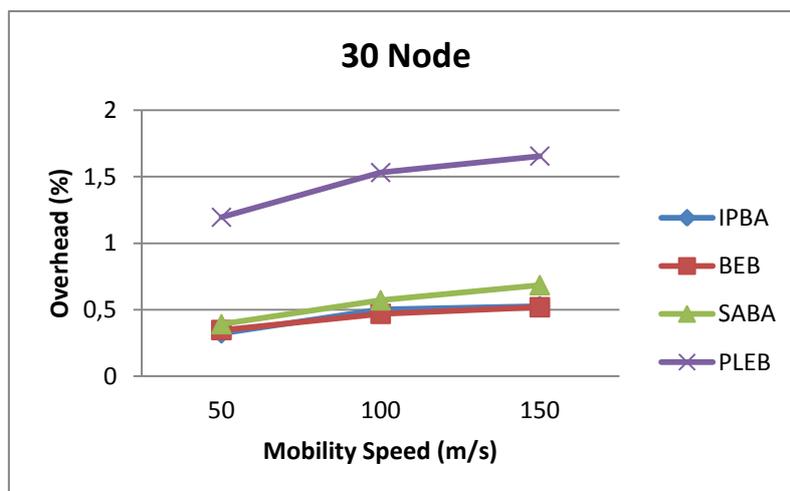


Figure 11: overhead of IPBA, BEB, PLEB and SABA when the number of nodes is 30 and traffic load is 10 Pact/Sec.

Figure 12 shows that IPBA's, SABA's and BEB overhead is the same for more stable areas (50 m/s) but IPBA's outperform PLEP's by 2.5% for the same case. IPBA outperform SABA and PLEP's by approximately 1%, 3.6% respectively for dynamic areas (150 m/s) within a mid -between sparse and dense- environment (60 nodes), while each of IPBA and BEB algorithms behave approximately the same with a simple preference for BEB in dynamic areas.

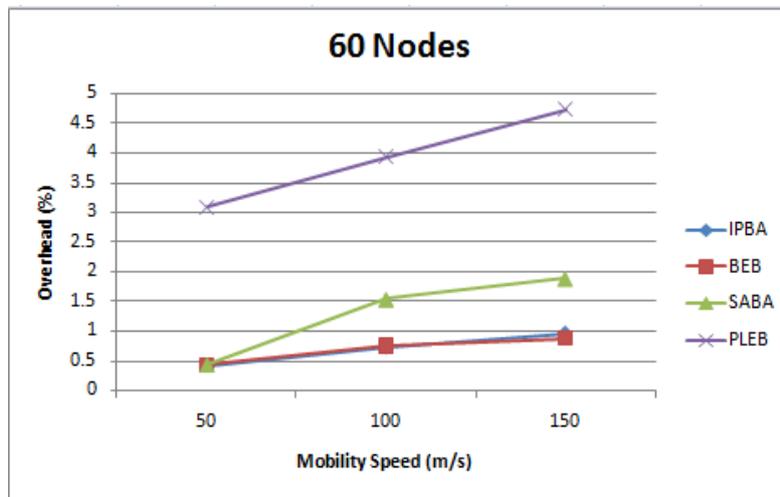


Figure 12: overhead of IPBA, BEB, PLEB and SABA when the number of nodes is 60 and traffic load is 10 Pack/Sec.

Figure 13 shows that IPBA's overhead outperform SABA's and PLEP's overhead by approximately 1%, 4% for more stable areas (50 m/s) and 1.5%, 4% for dynamic areas (150 m/s) within dense environment (100 nodes), while each of IPBA and BEB algorithms behave approximately the same with a simple preference for BEB in dynamic areas.

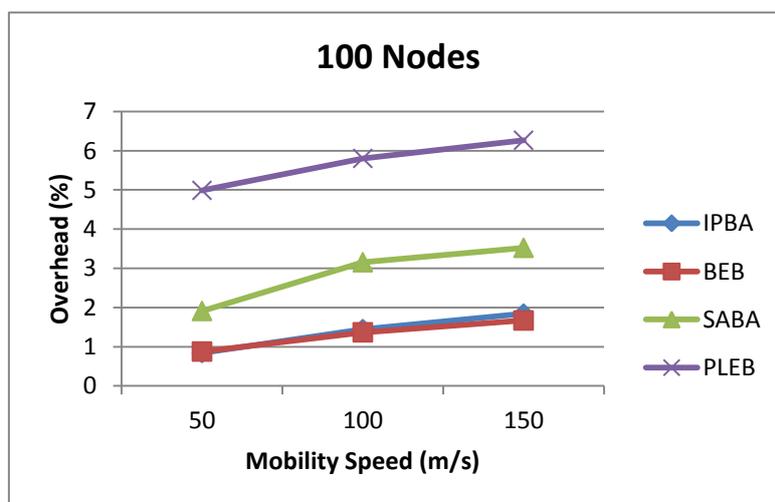


Figure 13: overhead of IPBA, BEB, PLEB and SABA when the number of nodes is 100 and traffic load is 10 Pack/Sec.

5. Conclusions and Future Work

In this paper, a new backoff algorithm IPBA is introduced and evaluated extensively compared to SABA and BEB algorithms. The results revealed that the IPBA outperforms the other two algorithms. IPBA backoff algorithm improves the packet delivery ratio up to 7.1% compared to BEB, up to 19% compared to PLEB, and up to 16.5% compared to SABA backoff algorithms in mobile ad hoc networks. Therefore, the increments in IPBA which depends on sliding the CW value to the right place increase the overall performance of the network.

As a future work number of neighbors for the sending node can be taken in consideration, adding to that: fuzzy logic techniques can also be applied to determine the proper shifting amount for contention window. Finally data mining algorithms can be also applied to predict the best contention window size using historical data that gained by simulation or fact results.

References

- [1] IEEE std 802.11-1999 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.
- [2] S. Manaseer, M. Ould-Khaoua, and L. Mackenzie. Fibonacci Backoff Algorithm for Mobile Ad Hoc Networks. Liverpool John Moores University, the 7th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (PGNET 06), Liverpool, 2006.
- [3] M. Bani Yassein, S. Manaseer, and A. Momani, "Adaptive Backoff Algorithm for Wireless Internet". Accepted and to be published in May of 2012, the journal of Emerging Technologies in Web Intelligence (JETWI).
- [4] S. Manaseer and M. Masadeh, "Pessimistic Backoff for Mobile Ad hoc Networks". Al-Zaytoonah University, ICIT'09, Jordan, 2009.
- [5] D. Sudipto, R. Rajesh, D. Pradip, "Optimizations to Multipath Routing Protocols in Mobile Ad hoc Networks," Proc. Intl. Conference on Emerging Applications of IT (EAIT 2006), 2006.
- [6] O. Uzoamaka, O. Ajirioghene, "The Performance of Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," [dissertation], Blekinge Institute of Technology, 2009.
- [7] I. Stojmenovic. Handbook of wireless networks and mobile computing, Wiley, New York, 2002.
- [8] J. DiMicco. "Mobile Ad Hoc Voting". Proceedings of CHI Workshop on Mobile Ad-Hoc Collaboration, 2002.
- [9] Y. Tseng and T. Hsieh, "Fully power-aware and location-aware protocols for wireless multi-hop ad hoc networks," Proceedings of the Eleventh International Conference on Computer Communications and Networks, 608-613, 2002.

<http://dx.doi.org/10.1109/ICCCN.2002.1043135>

- [10] J. Wu and F. Dai, "A distributed formation of a virtual backbone in MANETs using adjustable transmission ranges," Proceedings of the 24th International Conference on Distributed Computing Systems, 372-379, 2004.
- [11] F. Hong-yu, X. Lei, and L. Xiao-hui, "Logarithmic Backoff Algorithm of MAC Protocol in Ad Hoc Networks", in Proc. The Smart Internet, 2010, pp.1695-1698. <http://dx.doi.org/10.1109/ICEE.2010.429>.
- [12] N. Sumathi and A.S. Thanamani, "Pipelined backoff scheme for bandwidth measurement in QoS enabled routing towards scalability for MANets", in Proc. A2CWIC, 2010, pp.41-41. <http://dx.doi.org/10.1145/1858378.1858419>.
- [13] K. Takahashi, T. Tsuboi. A backoff algorithm for improving saturation throughput in IEEE 802.11 DCF. 14th IST Mobile & Wireless Communications Summit, June 2005.
- [14] M. Bani Yassein, S. Manaseer, and A. Al-Turani, "A Performance Comparison of Different Backoff Algorithms under Different Rebroadcast Probabilities for MANET's", University of Leeds, UKPEW, UK, 2009.
- [15] B. Ramesh, D. Manjula. An Adaptive Congestion Control Mechanism for Streaming Multimedia in Mobile Ad-hoc Networks. International Journal of Computer Science and Network Security (IJCSNS), 7(6), 2007, 290-295.
- [16] C. Huy, H. Kimz, and J. Hou. An Evaluation of the Binary Exponential Backoff Algorithm in Distributed MAC Protocols. Technical Report, University of Illinois – Urbana Champaign, 2005.
- [17] J. Goodman et al. Stability of binary exponential backoff. In the Proc. of the 17-th Annual in Journal of the ACM, 35(3), 1988, 579–602. <http://dx.doi.org/10.1145/44483.44488>.
- [18] X. Zeng, R. Bagrodia, M. Gerla. GloMoSim: A Library for Parallel Simulation of Large-scale Wireless Networks. Proceedings of the 12th workshop on Parallel and distributed simulation, Banff, Alberta, Canada, 1998, 154-161. <http://dx.doi.org/10.1145/278009.278027>

Copyright Disclaimer

Copyright reserved by the author(s).

This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).