# Adaptive Backoff Algorithm for Wireless Internet

Muneer O. Bani Yassein
Jordan University of Science and Technology /Department of Computer Science, Irbid, Jordan
Email: masadeh@just.edu.jo

Saher S. Manaseer and Ahmad A. Momani
University of Jordan/Department of Computer Science, Amman, Jordan
Jordan University of Science and Technology /Department of Computer Science, Irbid, Jordan
Email: saher@ju.edu.jo, ahmed_cs2004@yahoo.com

***Abstract*— The standard IEEE 802.11 MAC protocol uses the Binary Exponential Backoff algorithm. The Binary Exponential Backoff makes exponential increments to contention window sizes. This work has studied the effect of choosing a combination between linear, exponential and logarithmic increments to contention windows. Results have shown that choosing the right increment based on network status enhances the data delivery ratio up to 37% compared to the Binary Exponential Backoff, and up to 39 % compared to the Pessimistic Linear Exponential Backoff algorithms for wireless Internet.**

***Index Terms*— Wireless Internet, MAC, CW, Backoff algorithms**

## I. INTRODUCTION

The first appearance of wireless networks was in 1970s. Since that time, these networks are being developed so fast [22, 23]. It has been noticed that in the last decade all trends moved toward wireless Internet technology [23]. Also the mobile wireless network which is also called mobile ad hoc network has become the new age of wireless networks. We can distinguish two types of networks; infrastructure and ad hoc networks [1, 3, 22, 23]. See figures 1 and 2.

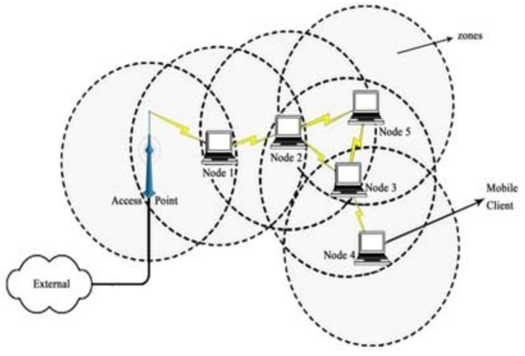

Figure 1: An example of an Infrastructure Wireless Internet [29]

Fig. 1 shows a simple example of the first type of infrastructure wireless networks. Communication between nodes at such networks is managed via a base station or a central access point. Each base station has a limited transmission range; therefore each node in the network connects to the nearest base station within its transmission range [23].

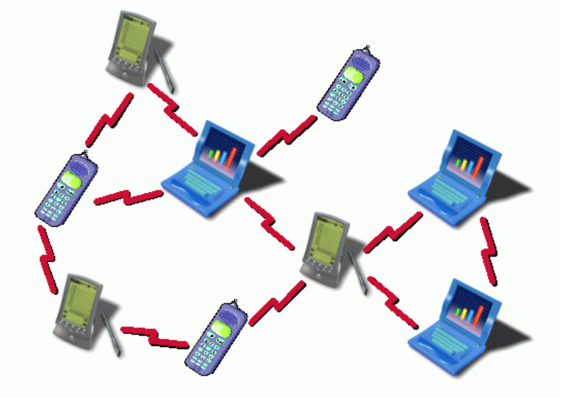

Figure 2: An example of Wireless Internet

On the other hand, Fig. 2 shows an example of the second type of wireless internet, a mobile ad hoc network (MANET). A MANET is a set of mobile nodes that communicate through wireless links. Each node acts as a host and router. Since nodes are mobile, the network topology can change rapidly and unpredictably over time [1, 3]. In other words, a MANET does not have a base station, so communication between nodes is managed by the nodes themselves. Moreover, nodes are not expected to be fully-connected, hence nodes in a MANET must use multihop path for data transfer when needed [24].

Recently, most interests were focused on MANETs due to potential applications provided by this type of networks such as military operation, disaster recovery, and temporary conference meetings.

### *A. Features and Characteristics of MANETs*

Despite a MANET has many features shared with infrastructure networks, it also has its own additional features. Some of these features are:

- Dynamic network topology: nodes in the network are free to move unpredictably over time. Thus, the network topology may change rapidly and unpredictably. This change may lead to some serious issues, such as increasing the number of transmitted messages between nodes of the network to keep

© 2012 ACADEMY PUBLISHER
doi:10.4304/jetwi.4.2.155-163

routing information table updated, and this will increase the network overhead [25].

- Distributed operations: in MANET there is no centralized management to control the network operations like security and routing, therefore, the nodes must collaborate to implement such functions. In other words, the control management is distributed among nodes of the network [22].

  - Limited resources: in MANETs nodes are mobile, so they suffer constrained resources compared to wired networks. For example, nodes in a MANET depend on batteries for communication and computation, so we should take in to account how to optimize energy consumption [26, 27, 28].

### B. *Applications of MANETs*

MANETs are deployed in different environments due to its valuable features of mobility, no base stations, Some of its applications are [22, 23]:

- Military Operations

In battlefield environments, a MANET can be very useful to setup a reliable communication between vehicles and solders where it seems almost impossible to have an infrastructure network in such environments.

- Emergency Operations

MANETs are very useful to be deployed in places that the conventional infrastructure-based communication facilities were destroyed by earthquakes, volcanoes, and any other supernatural situations. This is true since a MANET is a flexible, mobile, not expensive and can saves time for deployment phase.

- Mobile Conferencing

It is unrealistic to expect that all business is done inside an office environment, so a communication between a group of people or researchers can be achieved using MANETs.

## II. LITERATURE REVIEW

The Binary Exponential Backoff (BEB) [7, 8, 12, 13, 14] is used widely by IEEE 802.11 MAC protocols due to simplicity and good performance in general. The BEB algorithm works as the following:

When a node attempts to send a packet to a specified destination, it first senses the shared medium in order to decide whether to start transmitting or not. If the channel is founded to be idle, the transmission process starts. Otherwise the node should wait a random number of time between the range of [0, CW-1], this time is calculated using the formula:

$$\text{Backoff time} = (\text{Rand}()\ \text{MOD}\ CW)*\text{aSlotTime} \qquad (1)$$

After getting the backoff time, the node should wait until this time reaches zero before start transmitting. The backoff time (BO) is decremented by one at each idle time slot. But if the channel is busy the BO timer will be frozen. Finally if the node received an acknowledgment for the packet sent, the contention window (CW) is reset to minimum for that node. But if the node did not receive an acknowledgment (send failure occur), the CW is incremented exponentially to the new backoff value.

S. Manaseer and M. Masadeh [1] proposed the Pessimistic Linear Exponential Backoff (PLEB). This algorithm is composed of two increment behaviors for the backoff value; the exponential and linear increments. When a transmission failure occurs, the algorithm starts working by increasing the contention window size exponentially. And after incrementing the backoff value for a number of times, it starts increasing the contention window size linearly. PLEB works the best when implemented in large network sizes.

S. Manaseer, M. Ould-Khaoua and L. Mackenzie [2] proposed Fibonacci Increment Backoff (FIB). This algorithm uses the Fibonacci series formula which is defined by:

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

FIB algorithm aims to reduce the difference between contention windows sizes generated, resulting in a higher network throughput than the standard IEEE 802.11.

H. Ki, Choi, S. Choi, M. Chung and T. Lee [15] proposed the binary negative-exponential backoff (BNEB) algorithm. This algorithm uses exponential increments to contention window size during collisions (transmission failures), and reduces the contention window size by half after a successful transmission of a frame. The analytical model and simulation results in [15, 16] showed that the BNEB outperforms the BEB implemented in standard IEEE 802.11 MAC protocol.

S. Kang, J. Cha and J. Kim [17] proposed the Estimation-based Backoff Algorithm (EBA). This new algorithm has two main functions; the first one used to estimate the number of active nodes, and the second used to decide which contention window CW is optimal for the current case. The estimation function uses the average number of idle slots during backoff time to obtain the number of nodes which will be after the optimal CW for the current case.

EBA algorithm outperforms the binary exponential backoff (BEB), the exponential increase exponential decrease (EIED), the exponential increase linear decrease (EILD), the pause count backoff (PCB) and the history based adaptive backoff (HBAB) in network throughput and the mean packet delay.

S. Pudasaini, A. Thapa, M. Kang, and S. Shin [18] proposed an intelligent contention window control scheme for backoff based on Collision Resolution Algorithm (CRA). This algorithm keep a history for a success and failure access attempts in order to use this history to modify the contention window interval (CWmin, CWmax). This modification will cause a dynamic shifting for backoff interval to more suitable region. This new algorithm made some improvements to channel efficiency in terms of packet end-to-end delay.

© 2012 ACADEMY PUBLISHER

A. Balador, A. Movaghar, and S. Jabbehdari [19] proposed a new History Based Contention Window Control (HBCWC) algorithm for IEEE 802.11 MAC protocol. HBCWC made an optimization to the contention window values via saving the last three states of transmission. The main factor in this algorithm is the packet lost rate, if this factor increases due to collisions or channel errors then the CW size will increase. Otherwise it will decrease.

S. Manaseer and M. Ould-Khaoua [7], proposed the logarithmic backoff algorithm (LOG) for MAC protocol in MANETs. This algorithm uses logarithmic increments to provide a new backoff values instead of exponential ones. The new backoff values are extracted using the formula:

$(CW)\ new = (\log (CW)\ old) * (CW)\ old * aSlotTime.$

LOG algorithm generates values that are close to each other in order to achieve a higher throughput when used in MANETs.

V. Bharghavan, A. Demers, S. Shenker, and L. Zhang [10] proposed Multiplicative Increase and Linear Decrease (MILD) backoff algorithm. This algorithm uses multiplication by a factor when failed transmission occurs (due to collision or transmission failure). After a success transmission occur the contention window CW is decremented by a factor in order to reduce the probability of successful users to access the channel all the time. This decrement helps solving the unfairness problem which might occur to other users who have collisions and send failures [8, 9, 10].

J. Deng, P. Varshney, and Z. Haas [9] proposed the linear multiplicative increase and linear decrease (LMILD) backoff algorithm. LMILD uses both linear and multiplicative increments in the case of send failure; that is when a collision occurs, the colliding nodes increase their contention window CW multiplicatively, and other nodes overhearing this collision make a linear increment to their CW. In the case of successful transmission, all nodes decrease their contention windows linearly [8, 9].

LMILD has shown a better performance than the standard IEEE 802.11 when used in large network sizes.

It also outperforms the pessimistic linear exponential backoff (PLEB) in small networks, but PLEB achieves better performance than LMILD in large network sizes [1].

## III. THE NEW PROPOSED BACKOFF ALGORITHM

### A. Overview

In general, backoff algorithms tend to increase the contention window (CW) size after each transmission failure. Since this is true, a backoff algorithm should use a suitable increment for CW size in order to achieve the best performance. Many increment behaviors were used in this field such as: linear, exponential, logarithmic, and Fibonacci series. If we split the networks into three types: small, medium and large, each increment scheme would suit at the most two network types and drops dramatically at the third one. For example, the exponential increment of BEB algorithm which is used in standard IEEE 802.11 MAC does not achieve the best performance due to large CW gaps produced. Another example is a linear increment of LMILD; it does not allow adequate backoff time before data retransmission.

The Binary Exponential Backoff (BEB) algorithm increases the CW sizes exponentially based on transmission failure. That is, when a node has a packet to transmit, first it starts sensing the channel. If the channel is found to be idle, the node starts transmitting immediately the data packets. Otherwise, the backoff mechanism is triggered. Furthermore, a backoff timer is selected randomly from the current CW size; this timer is decremented only at each time slot found to be idle. When the timer reaches zero, the node transmits the data packets. If the acknowledgement received from the destination node, then CW size is reset to minimum. On the other hand, if the acknowledgement is lost the CW size is incremented exponentially. See Fig. 3.

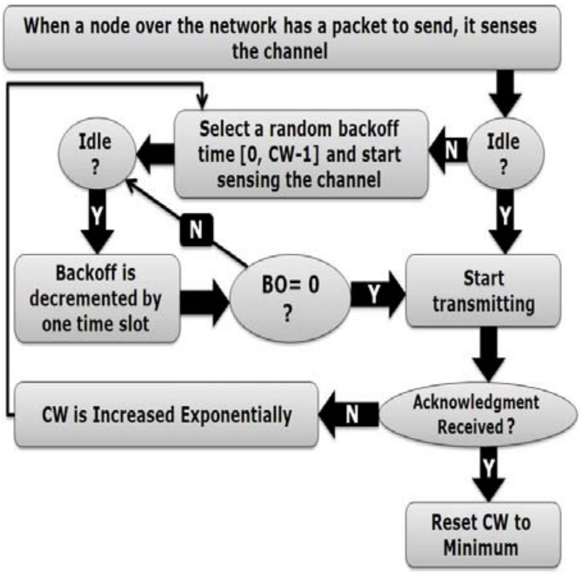

Figure 3: BEB algorithm description

Another backoff algorithm called pessimistic linear exponential backoff (PLEB). This algorithm combines two 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. See Fig. 4.

The next section suggests a new backoff algorithm that aims to improve the network performance overall. Our goal is to achieve a higher data delivery ratio with less overhead to the network. The new suggested algorithm uses a combination of different increments in order to take the advantage of each. The increments used in this algorithm are: linear, logarithmic, and exponential. The exponential increment aims to produce adequate CW lengths at the first stages of the new algorithm. Logarithmic increment provides proper increments to CW if it is still small; generally logarithmic increment is used as a transition stage towards linear increment to avoid continues exponential increments. The last increment used is linear increment, which aims to avoid large increments of exponential and logarithmic ones. The simulation results presented in the next paper show that the new proposed backoff algorithm improves the network delivery ratio and overhead.

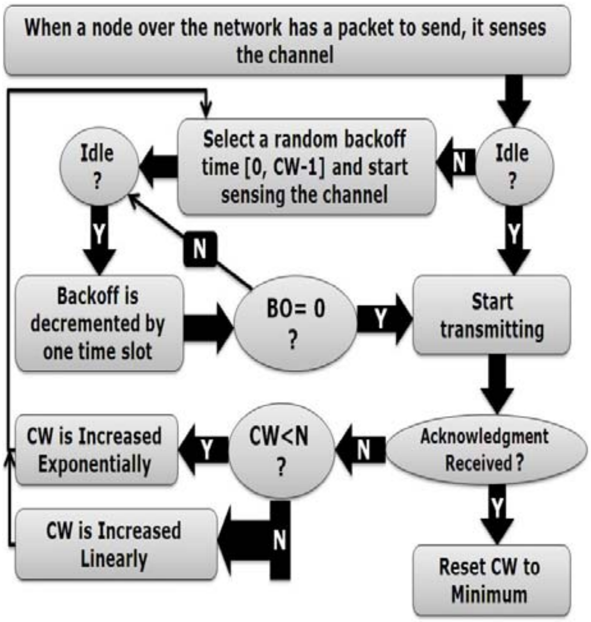

Figure 4: PLEB algorithm description

### B. *The Smart Adaptive Backoff Algorithm*

The Smart Adaptive Backoff Algorithm (SABA) is the new suggested backoff algorithm. This algorithm assumes that a network performance can be enhanced if the network is very sparse or congested too much. The explanation for this is the following; SABA assumes that solving the network collision in very sparse and congested networks will not be in the near future. Therefore, exponential increments are used. In very sparse networks, paths can be broken easily due to mobility, and mostly there exists only one path in the route table. So, the exponential increments will provide adequate values to establish other paths and start using them. On the other hand, in congested networks many nodes in the network use the same paths, so we should provide more time before start transmitting on this path. However, figure 5 shows in detail the basic functionality of SABA 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 value is saved in the history array as shown in line 21 of the algorithm. Otherwise, the backoff mechanism will be triggered. The lines 7-19 show a brief description of the adaptive process in SABA algorithm. It starts with line 7 by incrementing the CW exponentially only for a number of times based on transmission success. In other words, SABA provides an exponential increment for a node, and saves the CW size in case of transmission success. This process is repeated until the array of five elements is full. After that, the lines 8 and 9 shows that SABA calculates the average of CW sizes in the history array in order to start a new increment behavior based on the average value. This average is computed only once. Now, the lines 11-16 express that one increment behavior will be chosen: linear or logarithmic. If the average is not high (less than threshold N) the next increment behavior will be the logarithmic increment. Otherwise it will be the linear increment.

```
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         If (CW array of last five successes is full)
8             If (Array used for the first time)
9                 Calculate the average of the history array and use
                  it as a new CW value
10            Else
11                If (CW > N) then
12                    CW = CW + T
13                    Backoff-Timer = Random x;    1 ≤ x ≤ CW - 1
14                Else
15                    CW = Log (CW) * CW
16                    Backoff-Timer = Random x;    1 ≤ x ≤ CW - 1
17        Else
18            CW = CW * 2
19            1   Set BO to initial value
20  Else
21      Save the CW value used in the history
22   Go to line number 1
23  Stop Backoff-Timer = Random x          1 ≤ x ≤ CW - 1
20  Else
21      Save the CW value used in the history
22   Go to line number 1
  23  Stop
```

Figure 5: The Smart Adaptive Backoff Algorithm

The purpose of using the average is to reduce the CW size if it is a large number. Both of logarithmic and linear increments aim to avoid excessive of CW sizes in order to enhance the network performance.

© 2012 ACADEMY PUBLISHER

## IV. SIMULATION AND RESULTS' EVALUATION

In this research paper the network performance is measured by two criteria: packet delivery ratio and network overhead. In this paper, we present and evaluate the simulation results that were obtained for different scenarios. In the simulation experiments, we varied the number of sources and the maximum node speed.

We implemented the proposed backoff algorithm SABA using the GloMoSim [21] simulator to evaluate the performance of the new algorithm compared to the well-known BEB and PLEB algorithms.

### *A. Simulation Environment*

Our simulations were run using a network of 10, 20, 50 and 100 nodes placed randomly within a 1000 meter × 1000 meter area. Each node has a radio propagation range of 250 meters, and the channel capacity is 2 Mb/s. Each run executed for 900 seconds. We used the IEEE 802.11 as the MAC layer protocol. The Constant Bit Rate (CBR) node traffic is used in the simulations. We used the random waypoint model for node mobility. We used various node maximum speeds: 1, 2, 3 and 4 meter per second. In addition, we used traffic loads of 5, 10 and 20 packets per second, repeated for 5 and 10 sources.

### *B. Simulation Results*

Different performance metrics were used to evaluate backoff algorithms [1, 2, 4, 6, 7, 9, 10, 11, 12]. This study uses data delivery ratio and network overhead parameters to measure performance levels. The ideal case is to achieve maximum delivery ratio and minimum network overhead.

The figures (6, 7, 8, 9, 10, 11, 12, 13) show that SABA achieves the best performance in light and heavy (small and large) networks. This is not surprising since a network with a small number of nodes does not frequently trigger backoff mechanism; therefore, SABA which uses the average of last five exponential increments (successful increments only) outperforms the exponential increments in BEB and PLEB. Moreover, when the number of nodes increases and the network become of a large size (e.g. a network of 100 nodes), the exponential increments has essential role in network performance; that is, in BEB the exponential increments continues to enlarge the gaps between contention window sizes which in turn significantly reduces the network performance. Moreover, in PLEB algorithm the linear increments begin early in a way that does not allow the increments to the best values for large networks (i.e. the increments still small compared to network size and nodes mobility speed). On the other hand, SABA provides the best contention window increments. SABA aims to utilize exponential, logarithmic and linear increments to achieve the best performance in large networks. It starts with exponential increments until the node successfully sends five times. After that, it is expected to have a large contention window size. Therefore, it is reduced by calculating the average contention window sizes of these successive transmissions. After that, if the result is a large contention window size, the increment is linear. Otherwise, increment to contention window is logarithmic and then continues in linear way. The logarithmic increment is significantly smaller than exponential increment but it still larger than linear (this is needed in large networks to perform in a better way).

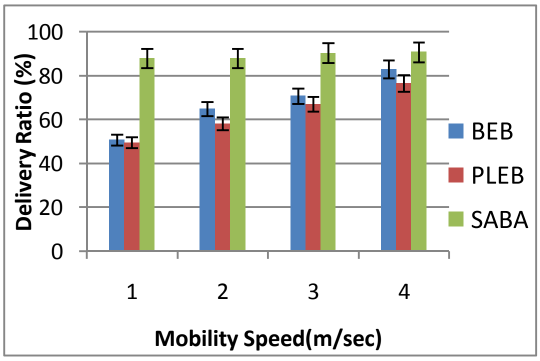

Figure 6: Data delivery ratio of BEB, PLEB and SABA for 20 nodes, 5 sources each source sending 20 packets per second.

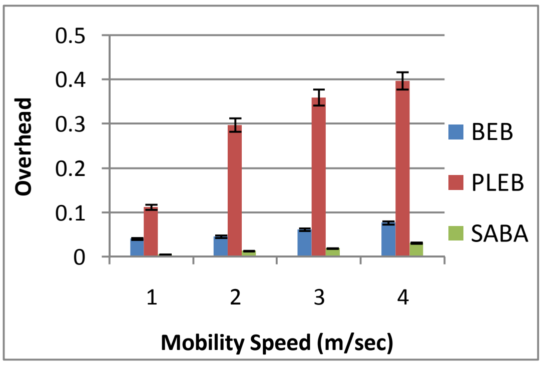

Figure 7: Overhead of BEB, PLEB and SABA for 20 nodes, 5 sources each source sending 20 packets per second.

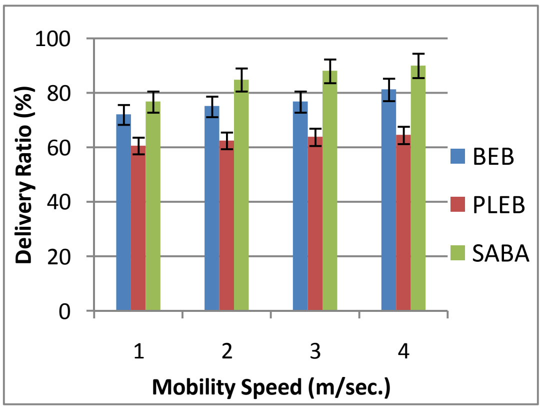

Figure 8: Data delivery ratio of BEB, PLEB and SABA for 20 nodes, 10 sources each source sending 10 packets per second.

At medium sized networks, the simulation results have shown in figures (14, 15, 16, 17) that SABA is still closely comparable the BEB and PLEB algorithms, and even gives a better performance at high traffic rates. BEB, PLEB and SABA backoff algorithms start to increase the contention window size exponentially.

© 2012 ACADEMY PUBLISHER

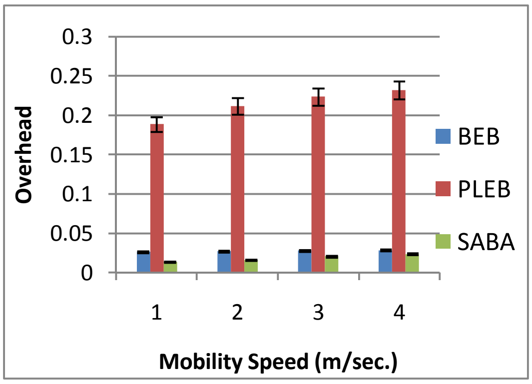

Figure 9: Overhead of BEB, PLEB and SABA for 20 nodes, 10 sources each source sending 10 packets per second.

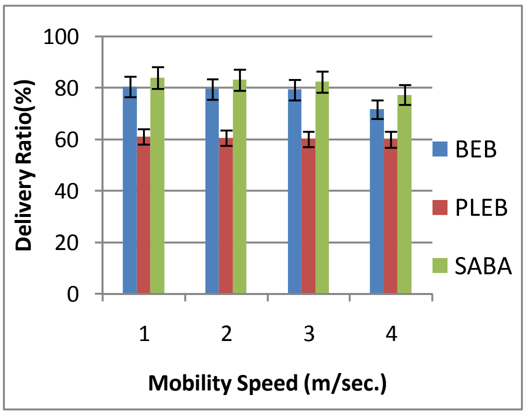

Figure 10: Data delivery ratio of BEB, PLEB and SABA for 100 nodes, 5 sources each source sending 20 packets per second.

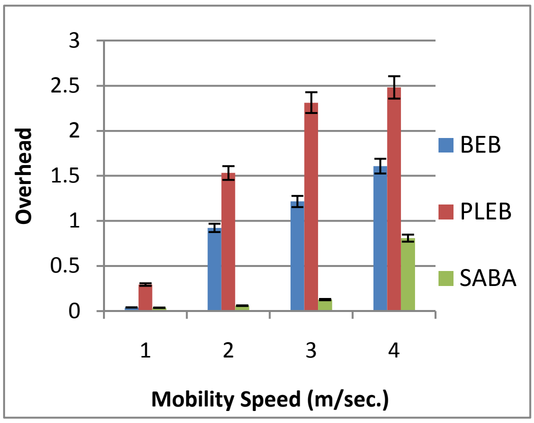

Figure 11: Overhead of BEB, PLEB and SABA for 100 nodes, 5 sources each source sending 20 packets per second

The number of backoff processes expected to be moderate. Therefore, the continuous exponential increments in BEB algorithm would not be a problem in this case. Moreover, SABA algorithm continues with logarithmic increments which are more suitable for high traffic rates in this type of networks. While the PLEB algorithm starts linear increments early making this mechanism unsuitable to gain the best performance for medium networks.

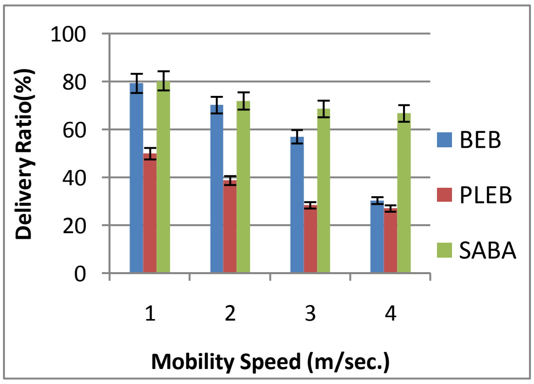

Figure 12: delivery ratio of BEB, PLEB and SABA for 100 nodes, 10 sources each source sending 20 packets per second.

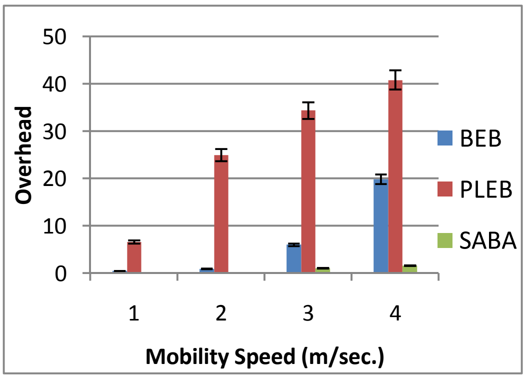

Figure 13: Overhead of BEB, PLEB and SABA for 100 nodes, 10 sources each source sending 20 packets per second.

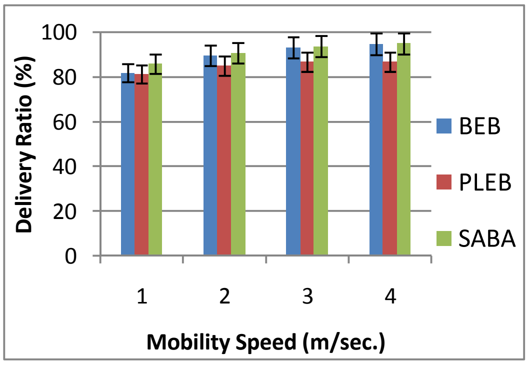

Figure 14: Data delivery ratio of BEB, PLEB and SABA for 50 nodes, 5 sources each source sending 20 packets per second.

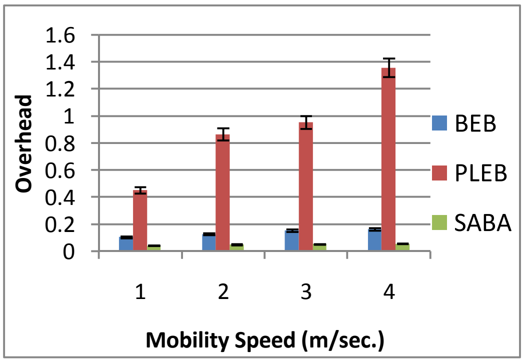

Figure 15: Overhead of BEB, PLEB and SABA for 50 nodes, 5 sources each source sending 20 packets per second.

© 2012 ACADEMY PUBLISHER

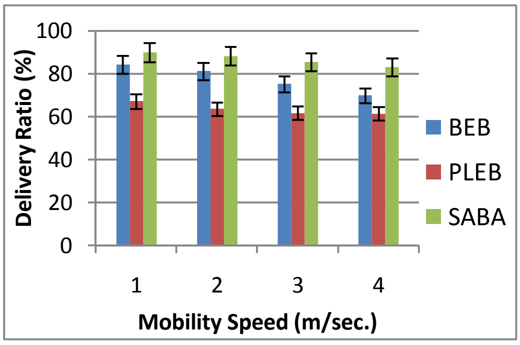

Figure 16: Data delivery ratio of BEB, PLEB and SABA for 50 nodes, 10 sources each source sending 10 packets per second.

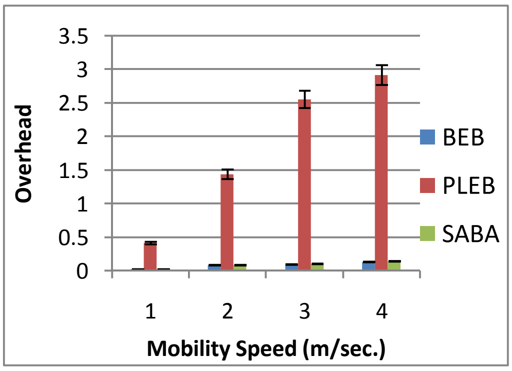

Figure 17: Overhead of BEB, PLEB and SABA for 50 nodes, 10 sources each source sending 10 packets per second

Furthermore, the results show some other important information; first, the linear increment behavior directly affects the network overhead. For example, PLEB algorithm provides the lowest network performance compared to BEB and SABA for all scenarios. This is true since the linear increments in the early stages of PLEB do not allow adequate increments for the CW values. To explain this in detail, the three types of network should be studied extensively. That is, in a sparse network linear increments make the source node sends the route request more frequent in order to establish a path between source and destination nodes (this is hard because the number of nodes is still small). Furthermore, in a medium and large network it is expected to have a more frequent backoff triggers. Therefore, it is normal to have a congested network and more broken routes. For all reasons mentioned above, it is normal to expect that linear increments in backoff algorithms cause a higher network overhead than exponential ones for the applied scenarios in this thesis. Secondly, in case of lightly loaded networks, the data delivery ratio is increasing. This means that a network of 20 nodes (5 and 10 sources sending data packets) and a network of 50 nodes (only 5 sources sending data packets) the data delivery ratio has increased due to the increasing mobility of nodes can be more useful. In other words, the data delivery ratio increases because when routes break due to mobility some other routes are built quickly. Finally, at highly loaded networks (ex. network of 100 nodes) the data delivery ratio decreases as mobility speed increases. This

## V. CONCLUSIONS

In this paper we presented a new backoff algorithm for MANETs called the Smart Adaptive Backoff Algorithm (SABA). The main objective of this work is to evaluate the performance of the new backoff algorithm in terms of network size, mobility speeds and traffic rates. The results obtained approve that changes made to contention window size increment and decrement directly affects network performance metrics such as data delivery ratio and overhead.

The results have shown that SABA outperforms BEB and PLEB algorithms in different network types. The data packet delivery ratio of SABA against BEB and PLEB algorithms was fluctuating. For example, in a small network type when a transmission rate is twenty packets per second, the number of sources is five and mobility is low SABA outperform BEB and PLEB by 37.11% and 38.45%, respectively. At high mobility, SABA outperforms BEB and PLEB by 7.74% and 14.13%, respectively. At medium network type, when a transmission rate is ten packets per second, the number of sources is ten and mobility is low SABA outperforms BEB and PLEB by 5.59% and 22.79%, respectively. At high mobility, SABA outperforms BEB and PLEB by 13.19% and 21.56%, respectively. At large networks, when a transmission rate is ten packets per second, the number of sources is ten and mobility is low SABA outperforms BEB and PLEB by 0.93% and 30.34%, respectively. At high mobility, SABA outperforms BEB and PLEB by 36.42% and 39.67%, respectively.

The network overhead of SABA against BEB and PLEB algorithms was fluctuating. For example, in a small network type when a transmission rate is twenty packets per second, the number of sources is five and mobility is low SABA outperform BEB and PLEB by 0.249% and 0.392%, respectively. At high mobility, SABA outperforms BEB and PLEB by 0.299% and 0.746%, respectively. At medium network type, when a transmission rate is twenty packets per second, the number of sources is ten and mobility is low SABA outperforms BEB and PLEB by 0.427% and 8.875%, respectively. At high mobility, SABA outperforms BEB and PLEB by 1.793% and 28.486%, respectively. At large networks, when a transmission rate is twenty packets per second, the number of sources is ten and mobility is low SABA outperforms BEB and PLEB by 0.454% and 6.645%, respectively. At high mobility, SABA outperforms BEB and PLEB by 18.248% and 39.211%, respectively.

In general, the results of this research paper indicate that each type of networks needs a different way to handle contention window increment. That is, for small networks low increments are preferred (lower than exponential). On the other hand, for medium and large networks it is preferred to have low increments after large ones.

Finally, this work has studied the effect of choosing the behavior changing point between linear, logarithmic and exponential increments in the proposed algorithm SABA. Results have shown that using the suitable increment type according to the network status increases overall network performance.

## ACKNOWLEDGMENT

We would like to thank the staff of the Faculty of Computer and Information Technology, Jordan University of Science and Technology (JUST) and the University of Jordan (JU), Jordan, for their great and continuous support.

## REFERENCES

[1] S. Manaseer and M. Masadeh. “Pessimistic Backoff for Mobile Ad hoc Networks”. Al-Zaytoonah University, the International Conference on Information Technology (ICIT'09), Jordan, 2009.
[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] Y. Yuan, A. Agrawala. “A Secure Service Discovery Protocol for MANET”. Computer Science Technical Report CS-TR-4498, Computer Science Department, University of Maryland, 2003.
[4] S. Manaseer, M. Ould-Khaoua, and L. Mackenzie. “On a Modified Backoff Algorithm for MAC Protocol in MANETs”. Int. J. of Information Technology and Web Engineering, Vol. 2(1), pp. 34-46, 2007.
[5] P. Karn. “MACA - A new channel access method for packet radio”. ARRL/CRRL Amateur Radio 9th computer Networking Conference, London, pp. 134–140, 1990.
[6] 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, the 25th UK Performance Engineering Workshop (UKPEW), UK, 2009.
[7] S. Manaseer and M. Ould-Khaoua. “Logarithmic Based Backoff Algorithm for MAC Protocol in MANETs”. Technical Report, University of Glasgow, 2006.
[8] H. Wu and Y. Pan. “Medium Access Control in Wireless Networks”. Nova Science Publishers Inc, pp. 26- 29, 2008.
[9] J. Deng, P. Varshney, and Z. Haas. “A New Backoff Algorithm for the IEEE 802.11 Distributed Coordination Function”. Communication Networks and Distributed Systems Modeling and Simulation (CNDS’04), San Diego, California, 2004.
[10] V. Bharghavan et al, “MACAW: A Media Access Protocol for Wireless LAN’s”, in Proc. ACM SIGCOMM ’94, pp. 212–225, 1994.
[11] S. Zhalehpoor and H. Shahhoseini. “SBA Backoff Algorithm to Enhance the Quality of Service in MANETs”. International Conference on Signal Acquisition and Processing ICSAP, Kuala Lumpur, Malaysia, pp. 43-47, 2009.
[12] 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.
[13] J. Goodman et al. “Stability of binary exponential backoff”. In the Proc. of the 17-th Annual in Journal of the ACM, Vol. 35(3), pp. 579–602, 1988.
[14] Y. Wang, “Medium Access Control in Ad Hoc Networks with Omni-directional and Directional Antennas”, University of California, doctoral Dissertation, 2003.
[15] H. Ki, Choi, S. Choi, M. Chung, and T. Lee. “Performance evaluation of Binary Negative-Exponential Backoff Algorithm in IEEE 802.11 WLAN”. Lecture Notes in Computer Science, Springer-Verlag Berlin Heidelberg, vol. 4325, pp. 294–303, 2006.
[16] B. Choi, S. Bae, T. Lee, and M. Chung. “Performance Evaluation of Binary Negative-Exponential Backoff Algorithm in IEEE 802.11a WLAN under Erroneous Channel Condition”. Lecture Notes in Computer Science, Springer-Verlag Berlin Heidelberg, vol. 5593, pp. 237-249, 2009.
[17] S. Kang, J. Cha, and J. Kim. “A Novel Estimation-Based Backoff Algorithm in the IEEE 802.11 Based Wireless Network”. 7th Annual IEEE Consumer Communications & Networking Conference (CCNC), Las Vegas, Nevada, USA, 2010.
[18] S. Pudasaini, A. Thapa, M. Kang, and S. Shin. “An Intelligent Contention Window Control Scheme for Distributed Medium Access”. 7th Annual IEEE Consumer Communications & Networking Conference (CCNC), Las Vegas, Nevada, USA, 2010.
[19] A. Balador, A. Movaghar, and S. Jabbehdari. “History Based Contention Window Control in IEEE 802.11 MAC Protocol in Error Prone Channel”. Journal of Computer Science, 6 (2), pp. 205-209, 2010.
[20] 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), Vol. 7(6), pp. 290-295, 2007.
[21] 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, pp. 154-161, 1998.
[22] I. Stojmenovic. “Handbook of wireless networks and mobile computing”, Wiley, New York, 2002.
[23] C-K Toh. “Ad hoc mobile wireless networks, protocols and systems”, Prentice-Hall, New York, 2002.
[24] IEEE, ANSI/IEEE standard 802.11, 1999 Edition (R2003), Part 11: “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”.
[25] R. Rajaraman. “Topology control and routing in ad hoc networks: a survey”. ACM Special Interest Group on Algorithms and Computation Theory (SIGACT), Vol. 33(2), 2002.
[26] V. Rodoplu and T. Meng. “Minimum energy mobile wireless networks”. IEEE Journal of Selected Areas in Communications, Vol. 17(8), pp. 1333-1344, 1999.
[27] S. Nesargi and R. Prakash. “MANETconf: configuration of hosts in a mobile ad hoc network”. Proceedings of IEEE Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), Vol(2), pp. 1059-1068, 2002.
[28] 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, pp. 608-613, 2002.
29. Manaseer. “On Backoff Mechanisms for Wireless Mobile Ad Hoc Networks”. Doctoral Dissertation, United Kingdom (UK), Glasgow University, December, 2009

© 2012 ACADEMY PUBLISHER

**Muneer O. Bani Yassein** received his B.Sc. degree in Computing Science and Mathematics from Yarmouk University, Jordan in 1985 and M. Sc. in Computer Science, from Al Al-bayt University, Jordan in 2001. He received his Ph.D. degree in Computing Science from the University of Glasgow, U.K. in 2007. His current research interests include networking protocol development and analysis the performance probabilistic flooding in mobile ad hoc networks.

**Saher S. Manaseer** received PhD degree in Computer Science from the Department of Computing Science at the University of Glasgow in 2009. His main area of research is Computer Networks and Embedded Systems. Currently, he is an active researcher in the field of Mobile Ad Hoc Networks. More specifically, his research on MANETs is focused on developing MAC layer protocols. Before obtaining his PhD, He got his Masters in Computing Science from the University of Jordan in 2004 where his research in that era was focused on Software Engineering. Since he earned his undergraduate degree in Computer science, he has worked in two parallel directions, being Training and software development.

**Ahmad A. Momani** received his B.Sc. degree in Computer Science from Jordan University of Science and Technology, Jordan in 2007 and M. Sc. in Computer Science, from Jordan University of Science and Technology, Jordan in 2011. His main area of research is Computer Networks. Currently, he is an active researcher in the field of Mobile Ad Hoc Networks. More specifically, his research on MANETs is focused on developing MAC layer protocols. Before obtaining his Masters, he has worked as a programmer and then a teacher in the Ministry of Education.

.

© 2012 ACADEMY PUBLISHER