Thorough study of the Logarithmic Backoff Algorithm for MAC Protocol in

Saher S. Manaseer  
University of Glasgow, UK  
saher@dcs.gla.ac.uk

Mohamed Ould-Khaoua  
University of Glasgow, UK  
mohamed@dcs.gla.ac.uk

Lewis M Mackenzie  
University of Glasgow, UK  
Lewis@dcs.gla.ac.uk

ABSTRACT
In wireless communication environments, backoff is traditionally based on the IEEE binary exponential backoff (BEB). As each node has to obtain medium access before transmitting a message, in dense networks, the collision probability in the MAC layer becomes very high when a poor backoff algorithm is used. The Logarithmic algorithm proposes some improvements to the backoff algorithms that aim to efficiently use the channel and to reduce collisions. The algorithm under study is based on changing the incremental behavior of the backoff value. The Binary Exponential Backoff (BEB) is used by IEEE 802.11 Medium Access Control (MAC). Results from simulation experiments reveal that the algorithm subject to study achieves higher throughput and less packet loss when in a mobile ad hoc environment.

Keywords: IEEE 802.11, Ad Hoc networks, Medium access control, Backoff algorithm, Throughput.

1. Introduction
Since their emergence, wireless networks have become increasingly popular in the computing industry. This is particularly true within the past decade, which has seen wireless networks being adapted to enable mobility. There are currently two variations of mobile wireless networks [10], infrastructure and ad hoc wireless networks. 
Infrastructured networks, i.e., those networks with fixed and wired gateways, have bridges known as base stations. A mobile unit within these networks connects to, and communicates with the nearest base station within its communication radius. Typical applications of this type of network include wireless local area networks [10].

Mobile Ad Hoc Networks (MANETs) are getting more and more attention. Unlike wired networks, MANETs are easily deployed, and need no infrastructure [7]. Such networks can be useful in disaster recovery where there is not enough time or resources to configure a wired network. Ad hoc networks are also used in military operations where the units are moving around the battlefield in a random way and a central unit cannot be used for synchronization [7].

In MANETs, a central station is not needed to control the different types of operations taking place allover the network [12]. A node participating in an ad hoc network must have the ability to act as a client, a server, and a router [7]. Nodes should also have the ability to connect to the network and to automatically configure to start transmitting data over the
network. This is the reason why ad hoc protocols in general function in a distributed manner [12]. The distributed Coordination Function (DCF) is used for synchronous, contention-based, distributed access to the channel [3]. MANETs use a shared medium to transfer data between its nodes.

It is unrealistic to expect a mobile ad hoc network to be fully connected, where a node can communicate directly with every other node in the network. Typically nodes must use a multihop path for transmission, and a packet may traverse multiple nodes before being delivered to its destination.

The wireless medium used by MANETs has a number of problems. Bandwidth sharing, signal fading, noise, interference, etc. with such a shared medium, an efficient and effective medium access control (MAC) is essential to share the scarce bandwidth resource [4] [7]. Based on the features mentioned, the design of the medium access control (MAC) protocol is a significant factor affecting a mobile ad hoc network.

As a part of an efficient medium access control protocol, a backoff algorithm is used to avoid collisions when more than one node try to access the channel [12]. Only one of the nodes is granted access to the channel, while other contending nodes are suspended into a backoff state for some period (BO) [8]. Many backoff algorithms have been developed in the literature [13, 8]. One example is the Multiplicative Increase Linear Decrease (MILD) algorithm [13]. This algorithm improves the total throughput of the network, but the cost of this improvement is the need of a perfect knowledge about collisions happening over the network, which is high cost, hard-to-acquire knowledge.

In a normal LAN the total number of nodes of the network is easily obtained. However, as nodes in MANETs are mobile, knowing the number of nodes may incur a high cost, since this knowledge needs to be updated. One approach to update and keep the knowledge coherent is by exchanging “hello” packets between neighboring nodes [12]. Using “hello” packets is a broadcast over the network. The broadcast generates extra traffic load over the network, consumes a part of the network resources, causes a longer delay, requires more control processing, and even gives more work to the backoff algorithm itself. Other backoff algorithms have tried to find a fixed optimum backoff value to use. But, even though, the distributed functioning was not complete [1].

Many researchers have proposed the mechanism of channel sensing, or packet sensing to avoid collision. The sensing mechanisms typically rely on the transmitter and receiver performing a handshake prior to the transmission of the data packet [2]. More specifically, The Medium Access Collision Avoidance (MACA) method proposed by Karn [11] implements the handshake via a pair of Request-To-Send (RTS) and Clear-To-Send (CTS) messages. When a node has to send data to another, it first sends a short RTS packet to the destination. The receiver responds with a CTS packet [2]. On receipt of the CTS, the sender sends its queued data packet(s). All other nodes overhearing the CTS message will defer from sending out any packet until the predicted transmission period indicated in the CTS packet, is passed. Any node that overhears the RTS signal but not CTS is allowed to send out packets in a certain time period as either the RTS/CTS handshake is not completed or it is out of range of the receiver.

In the IEEE 802.11 standard MAC protocol, the Binary Exponential Backoff (BEB) is used. This algorithm functions in the following way [19]:

When a node over the network has a packet to send, it first senses the channel using a carrier sensing technique. If the channel is found to be idle and not being used by any other node, the
node is granted access to start transmitting. Otherwise, the node waits for an inter-frame space and the backoff mechanism is invoked. A random backoff time will be chosen in the range [0, CW-1]. A uniform random distribution is used here, where CW is the current contention window size. The following equation is used to calculate the backoff time (BO):

\[ BO = (\text{Rand}() \mod CW) \times \text{aSlotTime} \] (1)

The backoff procedure is preformed then, by putting the node on a waiting period of length BO. Using carrier sense mechanism, the activity of the medium is sensed at every time slot. If the medium is found to be idle then the backoff period is decremented by one time slot.

Backoff time (BO) new = (BO) old - aSlotTime (2)

If the medium is determined to be busy during backoff, then the backoff timer is suspended. Meaning that backoff period is counted in term of idle time slots. Whenever the medium is determined to be idle for longer than an inter-frame space, backoff is resumed. When backoff is finished with a BO value of zero, a transfer should take place. If the node succeeded to send a packet and receive an acknowledgment for that packet, then the CW for this node is reset to the minimum, which is equal to 31 in the case of BEB. If the transfer fails, the node goes into another backoff period. When going for another backoff period again, the contention window size is exponentially increased with a maximum of 1023 [21].

BEB has a number of disadvantages. One major disadvantage is the problem of fairness. BEB tends to prefer last contention winner and new contending nodes over other nodes when allocating channel access. This is done by choosing a random backoff value from a contention window (CW) which has a smaller size for new contending nodes and contention winners. This behavior causes what is known as “Channel capture effect” in the network. Another problem of BEB is stability. BEB has been designed to be stable for large number of nodes. Studies showed that it is not.

In this paper, we perform a deeper study and analysis of the logarithmic backoff algorithm proposed in a previous published work [12]. Some additional factors, such as mobility speed of nodes, are taken into consideration. Moreover, more topologies are studied to investigate the performance of the algorithm subject to study.

The rest of this paper is organized as follows. Section 2 describes the modified logarithmic backoff algorithm studied by this paper. Section 3 describes simulation environment and discusses the results compared to IEEE 802.11 BEB algorithm. Section 4 concludes the paper.

2. The modified backoff algorithm

In IEEE 802.11 MAC protocol, The BEB algorithm exponentially increases the size of contention window. BEB uses the following equation to increase the contention window size

\[ BO = (\text{Rand}() \mod CW) \times \text{aSlotTime} \]

The logarithmic backoff algorithm has used the logarithm of the current backoff time as the increment factor to calculate the next backoff. The following formula is used to give this result [12]:

\[ (BO)_{new} = (\log (BO)_{old}) \times (BO)_{old} \times \text{aSlotTime} \] (3)

The used formula provides different outcome for backoff times, the behavior of the new
formula can be seen in Figure 1. The further we go with backoff; the closer are the new values to the old values generated by the modified algorithm [12].

Figure 1: CW increase in logarithmic algorithm

Figure 2 explains the pseudo code of the modified algorithm.

<table>
<thead>
<tr>
<th>The modified logarithmic backoff algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 0:</strong> set BO to the initial value</td>
</tr>
<tr>
<td><strong>Step 1:</strong></td>
</tr>
<tr>
<td>while BO ≠ 0 do</td>
</tr>
<tr>
<td>For each time slot</td>
</tr>
<tr>
<td>If channel is idle then BO = BO - 1</td>
</tr>
<tr>
<td>If channel is idle for more than IDFS then</td>
</tr>
<tr>
<td>Send</td>
</tr>
<tr>
<td>Else</td>
</tr>
<tr>
<td>BO = log(BO) * BO</td>
</tr>
<tr>
<td>Go to step 1</td>
</tr>
<tr>
<td>Stop</td>
</tr>
</tbody>
</table>

Figure 2: The modified backoff algorithm.

The main idea behind choosing such an equation for calculating backoff time is that instead of going on a backoff period for X time slots, the node goes into two consecutive backoff periods say $i_1$ and $i_2$ where $i_1 + i_2 \approx X$ when the node is on a backoff period for a consecutive number of times. This allows the node a chance to access the channel and transmit in a way like if backoff is stopped in the middle of the backoff period X [12].

Reducing the channel capture effect [5] is another perspective of the new algorithm. In BEB, when a node loses in contention for channel access, there is a relatively high possibility that the next backoff timer will be double as the current value; this assigns the node larger probability to lose in the next contention against new arrivals and contention winners [9]. When using the logarithmic algorithm, the difference between the two backoff periods is smaller, and so the chance of losing the contention is not dramatically increased by the logarithmic algorithm. The modified algorithm also has stopped using the uniform random distribution. This, as we shown in the results in Section 3, hides the effect of the increment behavior of the CW.
3. Simulation Results and analysis
Simulation has been performed using 9 topologies with different number of nodes in each topology with a maximum speed of 10.0 m/s. And 20 more scenarios with different mobility speeds for a network of 50 nodes. The traffic over the network has been generated as Constant Bit Rate (CBR) traffic, the pause time is 10 s. Queue length is 50 packets.

Figure 3 displays the result of running the modified algorithm against the standard IEEE 802.11 binary exponential backoff algorithm. The figure shows that the throughput is higher for the modified algorithm. A network with larger number of nodes has a better throughput than the case of small number of nodes. The reason of this is, for larger number of nodes, contention is much higher and so, it is more probable for a node to backoff for more consecutive periods, this leads a more significant effect of the behavior of logarithmic algorithm and backoff values start to be closer.

The BEB uses a uniform random number distribution generator. The random distribution used covers the effect of contention window increment behavior. The reason of excluding the random number distribution in our algorithm is that a formal distribution only benefits from the window size increment by 50% of the generated random values, which is even useless in the case of the modified algorithm. The following graph in Figure 4, show a comparison of the same modified algorithm with another version of itself, where a random number distribution is used in the same way used by the standard binary exponential backoff algorithm. While many researches try to analyze and understand the incremental behavior of a backoff algorithm, the effects of all the changes made are reduced by the random number distribution used.

![Fig. 3: Throughput of BEB vs. modified algorithm.](image1)

![Fig. 4: Throughput of two versions of the modified algorithm.](image2)
Figure 5 shows that over different speeds, the exponential algorithm gives higher throughput at some speeds. But, the value of the overall throughput in the network is unstable, i.e. changing the speed 4 m/s to 6 m/s, for example, increases the throughput by 30%. Moreover, at speeds where BEB is accomplishing higher throughput, the difference in throughput is not of a significant value. By using the Logarithmic algorithm, the maximum throughput change is 15% compared to 45% increment for networks using BEB.

Figure 6 is used to demonstrate that, for speeds from 7 m/s to 12 m/s using BEB leads to less packet drops over the network. Although, again, the gap is not wide between the results of the two algorithms, the backoff algorithm used is not the only factor that affects the number of dropped packets; queue size is another factor for example. Generally, the two algorithms drop approximately the same amount of data for this network topology with 50 nodes only.

Figure 7: Throughput of the standard algorithms against the modified algorithm for different speeds.
Figure 7 presents evidence on the improvement Logarithmic algorithm is achieving. The results plotted in Figure 6 have been obtained using 50 nodes only. As mentioned before, logarithmic algorithm tends to accomplish better results with larger networks. As shown in Figure 7, increasing the number of nodes over the network leads to less packet loss for the logarithmic algorithm.

![Figure 7: Improvement in packet loss for different networks.](image)

Figure 8 exhibit packet forwarding count as an indicator on the activity of nodes and channel usage; Once again, logarithmic algorithm achieves higher number of forwarded packets almost over all values of speed used for this experiment.

4. Conclusions and future work

The Binary Exponential Backoff (BEB) is used by the IEEE 802.11 Medium Access Control (MAC) protocol. BEB uses uniform random distribution to choose the backoff value. In this paper, we have studied a modified logarithmic backoff algorithm, which uses logarithmic increments instead of exponential extension of window size to eliminate the effect of random number distribution. Results from simulations have demonstrated that the modified algorithm increased the total throughput of the mobile ad hoc networks especially when the system size is large. Moreover, the tested algorithm also has shown stable network throughput over different speeds of nodes, and also increased packet forwarding over the network. One drawback of the studied algorithm is a slight increment in the number of dropped packets during simulation time. But, this drawback is caused by other factors as well. Studying these factors is left for future work.
References


