Impatient Backoff Algorithm: Fairness in a Distributed Ad-Hoc MAC

Rajarshi Gupta
Corporate R&D, QUALCOMM Inc.
5775 Morehouse Drive, San Diego, CA 92121
Email: rajarshi@qualcomm.com

Jean Walrand
Department of Electrical Engineering and Computer Science
University of California, Berkeley, CA 94720
Email: wlr@eecs.berkeley.edu

Abstract—Many distributed multiple access (MAC) protocols use an exponential backoff mechanism. In that mechanism, a node picks a random backoff time uniformly in an interval that doubles in size after a collision. When used in an Ad-Hoc network spanning multiple interference domains, this backoff mechanism is unfair towards nodes in the middle of the network. Indeed, such nodes tend to experience more collisions than nodes with fewer neighbors; consequently they choose larger backoff delays than those other nodes; and as a result, get lesser throughput.

We propose a different backoff mechanism that achieves a fairer allocation of the available bandwidth, by decreasing the backoff delay upon collision or failure to send a packet. That is, a node becomes more aggressive after each failure. Accordingly, we call this mechanism the Impatient Backoff Algorithm (IBA). The nodes maintain stability of the algorithm by resetting, in a distributed way, the average backoff delays if they become too small. We perform a Markov analysis of the system to prove stability and fairness in simple topologies. We also use simulations to study the performance of IBA in random Ad-Hoc networks, and compare with an exponential backoff scheme. Results show that IBA achieves comparable mean throughput, while delivering significantly better fairness.

Keywords: Fairness, MAC, Ad-Hoc Networks.

I. INTRODUCTION

It has been observed that the widely used exponential backoff mechanism (e.g., IEEE 802.11) is unfair towards nodes in the middle of an Ad-Hoc network spanning multiple interference domains (see [1] and [2]). This unfairness results from the higher degree of contention that these nodes face compared to nodes at the outer edges. We illustrate that unfairness and propose a new backoff scheme to reduce it.

A. Unfairness of Exponential Backoff

To demonstrate the unfairness of the exponential backoff, we consider the network with three links as shown in Fig. 1. The laptops use 802.11b with link rates of 11 Mbps. The table summarizes the achieved rates on each link. Note that MAC overhead limits the maximum possible capacity of a link to $6Mbps$. We observe that links A and B can simultaneously achieve the maximum $6Mbps$ rate. When links A and X are on at once, they share the channel equally, each receiving approximately $3Mbps$. On the other hand, if A, B and X are all on simultaneously, link X’s throughput drops significantly.

It is easy to see the cause of this unfairness. 802.11b follows a backoff mechanism whereby nodes try to capture the channel after waiting for a random backoff selected uniformly in an interval. Upon a collision, nodes are required to double the interval and try again (i.e. try less aggressively). Link X above faces more contention than links A and B. Consequently, it collides more often, and so backs off more. As a result, link X succeeds much fewer times. This phenomenon biases the network against nodes in the middle of an Ad-Hoc network.

The motivation for this approach is stability: by backing off exponentially fast, the rate of transmission attempts decreases quickly even if more nodes become active. Consequently, the likelihood that only one node succeeds quickly approaches 1. Of course, the backoff delay increases somewhat but remains generally small compared to the packet transmission times. When all the nodes share a collision domain, the fairness issue does not arise. Thus, such a backoff scheme is suitable for the shared collision domain of the Aloha network, the original Ethernet, and typical 802.11 configurations. As we just discussed, the situation is quite different in Ad-Hoc networks.

Fig. 1. Simple topology demonstrating the unfairness of 802.11
B. Impatient Backoff Algorithm

We propose the novel Impatient Backoff Algorithm (IBA), that attempts to improve the fairness in a distributed MAC across multiple interference domains. When using IBA, nodes decrease their average backoff delay upon collision – thereby becoming more aggressive in attempting to capture the next slot. Also, nodes increase their average backoff delay upon successful transmission. The danger of the scheme becoming unstable because of frequent collisions is handled by resetting the average backoff delays when they get too small. We use a Markov chain model to show that IBA achieves fairness in simple topologies. We demonstrate the stability of the backoff scheme under reasonable assumptions by proving the positive recurrence of the Markov chain.

We then use a simulation model to study the performance of IBA in a random topology, and compare it against traditional 802.11-like exponential backoff mechanisms. Results show that IBA is able to maintain a level of mean throughput comparable to exponential backoff – but achieves significantly better fairness. We further study the effect of a realistic reset mechanism which imperfectly propagates the reset control messages hop by hop. We also consider variations caused by nodes that move, and nodes that follow sleep and wake cycles.

To the best of our knowledge, all existing distributed backoff mechanisms adhere to the basic principle of increasing the backoff upon collision. The central contribution of this paper is to propose the exact opposite principle, and demonstrate that this can indeed lead to stable and fair performance.

The paper is organized as follows. We begin by presenting the backoff model for IBA in Section II, and its reset mechanism in Section III. We analyze IBA in Section IV, while Section V presents simulation results. We compare and contrast our ideas with Related Work in the field in Section VI, before concluding the paper.

II. BACKOFF MODEL

A. Assumptions

We make a simplifying assumption that packet transmissions occur in a slotted and synchronized fashion. Each packet time slot is divided into two phases:

1. Backoff Contention Phase. In this phase, each node that has a packet to send generates a random backoff value. It waits for these many backoff mini-slots, where each mini-slot is much smaller than a packet transmission slot. If it has not heard a transmission from a neighboring node while waiting, the node sends out a short ‘Slot Capture Message’. All neighbors which hear this slot capture message, or carrier senses the channel, will keep quiet for this slot.

2. Packet Transmission Phase. At the end of the entire Backoff Contention Phase, all nodes which successfully sent out the slot capture messages will transmit a constant sized packet. Thus, only the nodes that have generated a backoff delay smaller than or equal to those of their neighbors get to transmit a packet. A successful transmission is confirmed by an acknowledgement (ack).

An example is shown in Fig. 2, where five nodes in a line contend for the channel. Each node is assumed to contend with neighbors two hops away. The lower part of the picture shows one slot. During the Backoff Contention Phase, node A chooses a smaller backoff than B and C – consequently it is able to send a Slot Capture Message that is heard by B and C, which keep quiet. Nodes D and E are not affected by A’s slot capture, and E wins that contention. As a result, nodes A and E utilize the Packet Transmission phase in parallel. Although C had a smaller backoff than E, C is quiet in this slot, allowing E to transmit. Note that all nodes wait till the end of the entire Backoff Contention phase, before beginning the Packet Transmission phase.

Collisions can occur if two neighbors choose the same backoff. In this case, neither will hear the other’s slot capture message and both will try to send packets, and collide – resulting in a wasted packet transmission slot. Also, slotted transmission implicitly assumes synchronization between the nodes, the details of which we do not address in this paper.

B. Exponential Random Backoff

Traditional backoff mechanisms choose a random backoff uniformly in \([0, B_L]\), where \(B_L\) is the backoff range. The mean backoff in this case is \(B_L/2\). Instead of using an uniform random variable, IBA chooses the backoff using an exponential random variable with mean \(B_L/2\). The exact number of backoff mini-slots is then determined by rounding the random variable and capping it at a maximum value, since the exponential random variable is unbounded.

We choose the exponential random variable for a key feature. When \(n\) nodes with mean backoffs \(b_1, b_2, \ldots, b_n\) contend, node \(i\) wins the contention when backoff \(B_i < B_j\ \forall j \neq i\). Ignoring collisions, the probability of this occurrence is:

\[
P(\text{Node } i \text{ wins contention}) = \frac{1/b_i}{\sum_{j=1}^{n}(1/b_j)}. \tag{1}
\]
C. Updating Average Backoff Delays

The key principle of IBA is that nodes that face more contention become more aggressive so that they can get their fair share of the channel. This is achieved by updating the backoff based on feedback received in the last slot. Assume that a node has a mean backoff delay $b$ and has a packet to transmit. If the node fails to send in the current slot, either because of a collision or because it loses during the contention phase, it decreases its mean backoff delay by a multiplicative factor $m > 1$. On the other hand, if the node transmits successfully, it increases its mean backoff delay by the same factor. Note that decreasing the mean backoff delay makes a node more likely to win during the next contention phase.

Upon failure, $b := b/m$
Upon success, $b := b \times m$

III. IBA Reset Mechanism

When there are a lot of contending nodes, all but one of them will decrease their mean backoff delay. This increases the chance that several of them will pick the same small backoff delay in the next slot and hence collide again. And again.

In order to avoid this situation, we propose the IBA Reset mechanism. We multiply the mean backoff delay of every node by a constant factor $R_F$ whenever any node’s mean backoff delay falls below a reset limit $R_L$. A reset thus does not alter the result of the contention phase, since Eqn. 1 is unchanged by multiplying every term by $R_F$.

In reality, it is impossible to change the mean backoff delay of all nodes simultaneously; any reset message needs to propagate through the network. We discuss the reset propagation scheme, as also the loss of some reset messages, and see that their effect on overall throughput and fairness is minimal.

A. Reset Propagation

In a practical network, the reset begins at the initiating node, and is broadcast hop-by-hop through the network. This is achieved by setting a reset flag in the next packet broadcast by the resetting node. Every node that hears the reset message also resets its mean backoff, and in turn propagates the message.

Reset messages have a time-to-live field, to ensure that they do not keep circulating indefinitely. Further, there is a minimum time gap between consecutive resets – thus a node does not reset multiple times, upon hearing the same reset message from several broadcast sources.

B. Lost Reset Messages

Given an imperfect broadcast mechanism, reset messages may get lost. But a lost reset message only has limited consequence. Consider a node that was an intended recipient of the lost reset message. This node has an (incorrect) low mean backoff, since it does not reset; so it is likely to win the next few contentions with its reset-effected neighbors. But, as a result of these wins, the node increases its mean backoff, while its neighbors decrease their backoff – moving the system towards a stable situation. The loss of the reset message does cause temporary and local unfairness, but the system always tends to move towards stability, with or without the resets.

C. Congestion Collapse

However, a congestion collapse may still occur if the resets are too slow to stabilize the network. Consider a dense subgraph, e.g. a clique topology with $q$ nodes. At each slot, at most one node succeeds while all the others fail. Consequently, at least $q - 1$ of the nodes divide their mean backoff delay by $m$. For a large enough $q$, some of the nodes will face repeated resets without ever getting a chance to transmit. This will happen when $q > \log_m R_F$.

It is therefore important to determine the expected size of the largest clique in the network, and choose $m$ and $R_F$ accordingly. An alternative is to make the value of $R_F$ dynamic. If a node faces repeated resets without being able to transmit, it may conclude that the reset is not strong enough. It can then send out a super-reset message (can guarantee its transmission by sending this with a backoff of 0) which will increase the reset factor $R_F$ itself for subsequent backoffs. This proposal is currently under further evaluation.

D. Reset Parameters $R_L$ and $R_F$

The reset limit $R_L$ is the smallest value to which the mean backoff may fall until it is reset. A small $R_L$ allows nodes to maintain low mean backoff delays. But the actual number of backoff slots is rounded to the nearest integer. So multiple neighbors with low mean backoff delays are more likely to choose the same backoff, leading to frequent collisions.

Choosing a large value for $R_L$ will alleviate this problem but leads to large backoff values. In our current model this has no ill-effect since all nodes always wait for the completion of the backoff contention phase before attempting to transmit. In a future un-slotted version of IBA, a large $R_L$ will increase the mean times that a node wastes while backing off. For simulation purposes, we chose $R_L = 3.2$, which is $1/5^{th}$ of the initial mean backoff value of 16.

Choosing a large value of $R_F$ makes us lose granularity. Since the actual backoffs are rounded above and below, a large $R_F$ may round off two different backoff values to the same upper limit. For example, with $R_F = 50$ and the backoff values upper limited to 1024, two resetting nodes with mean backoffs of 30 and 60 will both be reset to 1024. Since the nodes have no memory beyond the immediate backoff value, this can lead to unfairness. In our simulations, we chose $R_F = 10$.

E. Multiplicative Factor $m$

A small value of $m$ implies that nodes maintain a lower mean backoff delay on average, since the backoff rate may be closer to $R_L$ without hitting it. This leads to more collisions, following the same argument as Section III-D. On the other hand, a large value of $m$ causes frequent resets. In a dense network, the frequency of resets is approximately $\log_m R_F$.

The compromise is to choose a large value of $m$ as possible (to minimize collisions), yet choose it small enough
to keep resets under control. For a practical system, this is a design decision based on the efficiency of the reset mechanism. We chose \( m = 1.2 \) for our simulations. In this case, the reset frequency is bounded above by the rate of one reset every \( \log_{1.2} 10 = 12.6 \) slots.

F. Increasing Backoff in a Contention-Free Zone

A curious situation occurs when there are very few nodes in the system, resulting in almost no collisions. Since the transmissions always succeed, the backoffs increase to the highest value, leading to potential wastage. Such a corner case may be detected by the lack of collisions, and the nodes may temporarily adopt a lower ceiling for their mean backoff. Alternatively, if the mean backoff reaches the highest limit (call it \( R'_k \)), the node may divide the mean backoff by \( R'_k \), in effect performing the reverse of the reset operation. This ‘anti-reset’ is then propagated through the network in the same fashion as the normal reset message.

IV. ANALYSIS

For the purposes of analysis, we only evaluate the contention process, assuming that each contention win leads to the successful transmission of a single constant-sized packet.

A. Star Topology

The star topology is characterized by a middle node \( X \) that interferes with every other node, while the outer nodes do not interfere with each other. The star topology is of special interest in terms of fairness, since node \( X \) contends with every other node in the network. Under traditional backoff mechanisms (e.g. Fig. 1), the throughput achieved by node \( X \) continues to decrease as the number of outer neighbors increases. We want to ensure that IBA is fair for this topology.

When node \( X \) wins the backoff contention, it is the only one that transmits. However, when any other node wins the contention, node \( X \) has to keep quiet and so the remaining outer nodes believe they won the round and they all transmit. Thus, assuming that nodes always have a packet to send, the mean backoff delays of all the outer nodes remain in lock step.

Let the mean backoff of the middle node be \( b_1 \) and that of each of the \( n \) outer nodes be \( b_2 \). Since we use exponential random variables to generate the backoffs, the probability that middle node wins contention, as in Eqn. (1), is given by

\[
T_X = \frac{\frac{1}{b_1}}{\frac{1}{b_1} + \sum_{j=1}^{n} \frac{1}{b_2}} = \frac{\frac{1}{b_1}}{\frac{1}{b_1} + \frac{n}{b_2}} .
\]  

(2)

To simplify notation, we define \( r_i = 1/b_i \) to be the rate of the exponential random variable generating the backoff for node \( i \). With this notation, equation (2) becomes

\[
t_X = \frac{r_1}{r_1 + nr_2} .
\]  

(3)

Note the throughput-fairness tradeoff in the star topology. The maximum throughput is achieved when the middle node is quiet, and the outer nodes are always on, leading to a total throughput of \( n \). On the other hand, a fair share allows the middle node to be active 1/2 the time, while the other nodes are active during the remainder. Consequently, the throughput is only \( \frac{1}{2} + \frac{1}{2} = \frac{n+1}{2} \).

Assume that all the outer nodes start with the same initial mean backoff delay. Then, the star topology can be evaluated by an appropriate Markov Chain whose states capture the ratio between the mean backoff delays (i.e. between the backoff rates). Let state \( S_k \) designate the state when \( \frac{b_2}{b_1} = \frac{r_2}{r_1} = n \). Thus, the rates are \( n \times (m^2)^k \), as shown in Figure 3.

Following Eqns. (2) and (3), the probability that the middle node \( X \) in state \( S_k \) succeeds in the next slot is given by

\[
\pi_k = \frac{n \times m^{2k}}{n \times m^{2k} + n} .
\]

(4)

Since \( b_2 \) increases \( \pi_k \) and this moves the state to \( S_{k+1} \), and that of

\[
\pi_0 = 2 \pi_1 , \quad \pi_1 = \frac{m^2}{2} \pi_0
\]

We can then use this value of \( \pi_1 \) to determine \( \pi_2 \).

\[
\pi_1 \left( \frac{1}{m^2 + 1} + \frac{m^2}{m^2 + 1} \right) = \pi_0 \frac{1}{m^{2}x0 + 1} + \pi_2 \frac{m^2x2}{m^{2}x2 + 1}
\]

\[
\Rightarrow \pi_0 = \frac{m^4}{2m^4 - 1} \quad \Rightarrow \pi_2 = \frac{1}{2} \left( 1 + \frac{1}{m^2} \right) \frac{1}{m^2} \pi_0
\]

By repeating the process, we evaluate the steady state probability of state \( S_k \) as,

\[
\pi_k = \frac{1}{2} \times \left( 1 + \frac{1}{m^{2k}} \right) \times \frac{1}{m^{k(k-1)}} \times \pi_0
\]

(4)

Finally, we can evaluate the actual probabilities, since \( \sum_{k=-\infty}^{\infty} \pi_k = 1 \). We get,

\[
\pi_0 \left[ 1 + \sum_{k=1}^{\infty} \left( 1 + \frac{1}{m^{2k}} \right) \right] = 1
\]

(5)

Tabulating \( \pi_0 \) for various values of \( m \),

<table>
<thead>
<tr>
<th>( m )</th>
<th>1.05</th>
<th>1.2</th>
<th>1.5</th>
<th>2</th>
<th>4</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \pi_0 )</td>
<td>0.123</td>
<td>0.230</td>
<td>0.325</td>
<td>0.395</td>
<td>0.471</td>
<td>0.495</td>
</tr>
</tbody>
</table>

We can further prove the following theorem about the expected transmission probability of nodes in the star topology.
Theorem 1: Each node in the star topology has an expected transmission probability of $\frac{1}{2}$.

Proof: The transmission probability of X at state $S_k$ is given by $t^X_k = \frac{m^k}{m^{2k}+1}$ (Figure 3). Then, the expected transmission probability of node X, $E[t^X]$ is calculated as,

$$E[t^X] = \sum_{k=-\infty}^\infty \pi_k t^X_k = \pi_0 \frac{1}{2} + \sum_{k=1}^{\infty} \pi_k (t^X_k + t^{-X}_k)$$

But $t^X_k + t^{-X}_k = \frac{m^{2k}}{m^{2k}+1} + \frac{m^{-2k}}{m^{-2k}+1} = 1$

Hence $E[t^X] = \frac{\pi_0}{2} + \sum_{k=1}^{\infty} \pi_k = \frac{1}{2}$

The expected probability of successful transmission for the outer nodes, is $1 - \frac{1}{2} = \frac{1}{2}$.

In addition to each node having an average transmission rate of $1/2$, this Markov chain has a strong drift toward $S_0$, which suggests that the short-term behavior is fair.

B. Triangle Topology Model

The other topology that we analyze is the triangle topology (Figure 4), where each of three nodes interferes with the others. Clearly, the situation is symmetric, but we need to ensure that starting at any set of backoff rates, the system drifts back towards a state where all nodes have an equal probability of successful transmission.

The evolution of the system is modeled by a Markov chain that represents the backoff rates of the three nodes. A state is denoted by the triple $(1, b, c)$ that specifies the backoff rates of the three nodes in increasing order. We express each of the rates as a multiple of the smallest rate, so the first term is always 1. Thus the probability of the first node winning the contention is $\frac{1}{1+b+c}$, while the other two nodes win with probability $\frac{b}{1+b+c}$ and $\frac{c}{1+b+c}$, respectively. Every transmission results in one backoff rate being divided by $m$, and the other two rates being multiplied by $m$. As a result, the ratio of rates between any two nodes changes by a factor of $m^2$; consequently values $b$ and $c$ in the Markov chain are powers of $m^2$. In Figure 4, we represent the Markov chain resulting from a triangle topology. In the figure, we use a value of $m = 2$ as an illustration.

The Markov chain is clearly irreducible, but it is periodic with a period of 3. In order to conclude that the system is fair, we need to show the Markov chain is positive recurrent.

Theorem 2: The Markov chain that models the backoff rates in the triangle topology is positive recurrent.

Proof: Let $\mathcal{A}$ be a finite set of the states (to be defined below), including $(1,1,1)$. Define the function $f$ as

$$f(1,b,c) = \log_m b + \log_m c.$$  

(9)

Given state $S$, we further define the Lyapunov drift function

$$\gamma(S) = -f(S) + \left( \sum_{T \in N(S)} P_{ST} \cdot f(T) \right).$$  

(10)

Here $N(S)$ are the states neighboring to state $S$, and $P_{ST}$ is the transition probability from state $S$ to state $T$. By Pakes’ Lemma [3], if there is some $\epsilon$ such that

$$\gamma(S) \leq -\epsilon < 0, \forall S \in \mathcal{A},$$

then the chain is positive recurrent. Let $0 < \epsilon \ll 1$. By observing the structure of the Markov chain, we notice that there are three varieties of states as illustrated in Figure 5.
In this case, there are three possible backoff states: where we satisfy all the three cases above. By Pakes’ Lemma, the chain is positive recurrent.

This analysis allows us to conclude that no matter what the initial backoff delays are, the three nodes will drift towards states where their backoff delays are equal.

V. SIMULATION RESULTS

We have built a simulation model using Matlab [4] that allows us to compare the performance of the Impatient Backoff Algorithm with a slotted exponential backoff algorithm (EBA). EBA chooses a backoff uniformly in a given range, starting with $[0, 32]$. The node with the smallest backoff in its neighborhood is able to transmit in that slot. A collision doubles the range, while a successful transmission brings it back to the initial range. No changes are made to the backoff when the node is quiet. This models an idealized and slotted version of the 802.11 protocol [5].

We consider a random field of size $4km \times 4km$ and place 100 stationary Ad-Hoc nodes at random on it. The nodes have a transmission range of $500m$ and an interference range of $1km$. For starters, every node is assumed to be full buffer and have packets to send at all times (we relax this assumption in Sec V-B). Note that at any slot, multiple transmission can take place at different parts of the field.

Figure 6 expresses the simulation results for EBA and IBA, on the same topology. We denote a node by a circle with its center at the node’s location. The area of the circle is proportional to the throughput achieved by the node (i.e. number of successful transmissions).

By comparing the two graphs visually, we can qualitatively see the fairer nature of IBA. EBA in Figure 6 (left) includes many nodes with very small throughput circles – all of whom are able to increase their throughput in IBA, as denoted by the larger circles in Figure 6 (right). In fact, the lowest throughput achieved by a node in EBA is merely 0.009, while the lowest throughput in IBA is 0.049. Thus, IBA allows the highly congested nodes to significantly increase their share of the throughput, and hence achieve a fairer balance. We can also analyze the fairness quantitatively. Jain’s fairness index [6] in the EBA simulation is only 0.66 and it increases to 0.75 for IBA.

It is also important to compare the aggregate throughput of the two schemes, and it turns out that the mean throughput is about the same, at 0.1066 and 0.1046. This result is replicated over many simulations: IBA has significantly higher fairness than EBA, but very similar throughput.

A. Reset Propagation and Lost Resets

As explained in Section III, IBA requires all nodes to reset their mean backoff delays when any mean backoff goes below a reset limit. In a practical situation, this requires the propagation of the reset message through the network. We model this in our simulations by propagating the reset message
hop by hop from the originating node. A node resets its backoff only when it receives a reset message.

The reset messages have a time-to-live field to ensure expiration after a single reset. Also, a node does not reset more than once in a fixed number of slots, thus multiple reset messages starting from different parts of the network at around the same time cause the backoff delays to be reset only once. Furthermore, reset messages may get lost in the way. Our simulations account for situations where up to 10% of reset messages get lost.

It turns out that reset propagation and loss has marginal effect on the throughput and fairness of IBA. A delayed or lost reset message implies that the intended recipient node does not increase its backoff in a timely manner and hence continues to have a low backoff. However, the result of this is that it wins a few contentions unfairly – which in turn causes it to increase its backoff anyway. The unfairness only allows a few extra packet transmissions and does not persist. Thus the scheme is fair in the long run.

### B. Variations in Simulation Scenarios

**Movement:** We investigate the effect of random walk movement on IBA nodes, where every node moves randomly in discrete steps every 10 time slots. As in previous experiments, IBA achieves comparable throughput to EBA, but significantly better fairness.

**Sleeping and Waking Nodes:** We also consider the case when nodes do not always have packets to send. This is simulated by considering blocks of slots when a node is either awake or asleep. For simple topologies, the results show IBA to maintain its fairness even with a changing number of neighbors. An example is shown in Table I for a clique of five nodes. Different nodes have a different fair share since they are active for varying amounts of time, but the IBA scheme is able to provide them with that share.

### VI. RELATED WORK

#### A. Backoff Schemes in Ad-Hoc Networks

Our focus is on the backoff mechanism to handle congestion in ad-hoc networks, and their resulting throughput and fairness. Bianchi [7] presents a two dimensional Markov model of the exponential backoff mechanism in 802.11. By assuming that the probability of collision of a node does not depend on its own state history, the author is able to derive expressions for the packet transmission probability and saturation throughput. Ergen and Varaiya extend that work in [8].

Slotted media access protocols have been studied for several decades (see [9] for an overview). In particular, detailed analysis of Slotted Aloha MAC [10] has evaluated its throughput and fairness. Recently, Yuan and Marbach [11] have proposed a rate control for random access networks. By controlling the rate at which the nodes attempt to transmit, the system is shown to be stable, i.e. positive recurrent.

Our model for IBA differs from [7], [8] and also [10], [11] in a crucial aspect – we consider Ad-Hoc networks spanning multiple interference domains. Consequently, the nodes do not all share the same medium, and so have different degrees of contention, and therefore different collision probabilities. It is this fact that biases such protocols against middle nodes in the network. Addressing this fairness issue is the primary goal of
this paper, and leads us to propose the strategy of becoming more aggressive upon collision.

B. Fairness in Ad-Hoc MAC

Many authors have studied the issue of achieving fairness in an Ad-Hoc MAC (e.g. [12], [13]). As noted in [14] this is critical since fairness mechanisms in upper layers such as TCP is ineffective without a fair MAC layer underneath.

In [15], the authors define a fairness index, and each station keeps an estimate of its fairness over all the other stations. If the ratio of fairness is larger than a high threshold, the window size doubles; if the ratio is below a low threshold, the window size is halved. The window size remains unchanged when the ratio stays within the thresholds.

In the MACAW protocol [16], the authors introduce an additional field in the data packet to propagate the value of backoff timer so that the stations in the same region share the same backoff timer. The backoff algorithm here is Multiplicative Increase and Linear Decrease. In turn, the authors of [17] propose an Additive Increase Multiplicative Decrease mechanism. Each station keeps increasing its sending rate by a constant rate, unless there is a collision – when the sending rate is decreased by a multiplicative factor. A Multiplicative Increase/Decrease rule is proposed in [18]. Finally, two other generalized backoff mechanisms in [19] and [20] use more sophisticated equations to control the backoff.

All of the above backoff algorithms follow the same basic philosophy: Upon collision, nodes increase their backoff and become less aggressive. IBA is unique because it reacts to collisions by becoming more aggressive upon collision. Our primary focus for the update rule is the widely prevalent exponential backoff, but clearly other backoff schemes might be applied too, within an impatient backoff framework.

VII. Conclusions

Traditional distributed Ad-Hoc MAC protocols (e.g. 802.11b) use random backoff delays to avoid collisions and share bandwidth amongst contending nodes. The fundamental backoff rule is to become less aggressive upon collision. We observe that in a network spanning multiple interference domains, this approach leads to unfairness towards nodes in the middle of the network.

We propose a novel backoff scheme that counters the conventional backoff wisdom. Nodes in our Impatient Backoff Algorithm decrease their backoff when they collide or are unable to send – thereby becoming more aggressive. This approach tends to help nodes with more neighbors and leads to a fairer allocation of bandwidth. The danger of the system becoming unstable due to frequent collisions is handled by resetting the mean backoff delays when they get too low.

We use Markov chains to analyze IBA in simple topologies. We look at two extreme topologies – the unfair star topology, and the symmetric clique topology. By proving positive recurrence of the system, we show that IBA is indeed stable.

We compare the performance of IBA with an idealized slotted exponential backoff scheme. In a random topology, IBA maintains the same mean throughput as EBA but has a significantly higher fairness index. Further extensions involving movements, and nodes that switch between active and sleep phases also give similar results.

IBA is by no means a complete MAC protocol. It is however a radically novel backoff mechanism for distributed Ad-Hoc networks – suggesting an approach that is counter to the basic principle of all existing schemes. This paper shows the feasibility and benefits of this scheme, and also proposes mechanisms to make IBA practicable. We hope that this will encourage further research in the direction of such impatient MAC protocols.

REFERENCES