

# Local Estimation of Probabilities of Direct and Staggered Collisions in 802.11 WLANs

Michael N. Krishnan, Sofie Pollin, and Avideh Zakhor

Department of EECS, U.C. Berkeley

Email: {mkrishna, pollins, avz}@eecs.berkeley.edu

**Abstract**—Current 802.11 networks do not typically achieve the maximum potential throughput despite link adaptation and cross-layer optimization techniques designed to alleviate many causes of packet loss. A primary contributing factor is the difficulty in distinguishing between various causes of packet loss, including collisions caused by high network use, co-channel interference from neighboring networks, and errors due to poor channel conditions. In this paper, we propose a novel method for estimating various collision type probabilities locally at a given node of an 802.11 network. Our approach is based on combining locally observable quantities with information observed and broadcast by the access point (AP) in order to obtain partial spatial information about the network traffic. We provide a systematic assessment and definition of the different types of collision, and show how to approximate each of them using only local and AP information. Additionally, we show how to approximate the sensitivity of these probabilities to key related configuration parameters including carrier sense threshold and packet length. We verify our methods through NS-2 simulations, and characterize estimation accuracy of each of the considered collision types.

## I. INTRODUCTION AND RELATED WORK

In 802.11 WLANs, nodes cannot distinguish between packet loss due to fading and collision because the symptoms are the same, namely a missing acknowledgement. The situations which result in each type of loss, however, require different specific actions to maximize throughput. For instance, *channel errors* occur when channel conditions are poor due to large path loss or multipath fading. These errors can be mitigated by using link adaptation (LA) to adapt the modulation and coding levels of each transmission, or by using forward error correction at the application layer. On the other hand, collisions happen when multiple transmissions occur at the same time causing interference and hence low relative signal power. Collision avoidance in the 802.11 DCF is achieved by means of the Binary Exponential Backoff scheme, where colliding nodes choose a larger random backoff counter to minimize repeat collision probability when retransmitting the packet.

In this paper we distinguish between different types of collisions, based on the nature of the way they occur in the network. Our motivation is that by estimating probabilities of each type of collision, it should be possible to arrive at the corrective action to minimize their impact on the future transmissions. Specifically, we focus on two broad classes of collisions, *direct* and *staggered*. A *direct collision (DC)* happens when two nodes start transmitting a packet at the same time. These occur in the 802.11 Distributed Coordination Function (DCF) when two nodes finish their backoff at the same time. Co-channel or hidden node collisions occur when multiple far-away nodes that cannot sense each other transmit at the same time. In these

collisions, transmissions do not necessarily start at exactly the same time. As such, we refer to these collisions as *staggered collisions (SCs)*. Traditionally, staggered collisions are dealt with by transmitting Request-To-Send (RTS) and Clear-To-Send (CTS) messages before each data packet, since the hidden node likely receives the CTS message and avoids collision. This scheme however, is rarely used in practice due to the large overhead.

We further subdivide SCs into two types, namely type 1 and type 2. A *staggered collision of type 1 (SC1)* for a given node is one in which the node under consideration transmits first and is then interrupted by another node. A *staggered collision of type 2 (SC2)* for a given node is one in which the node under consideration interrupts the transmission of a hidden node. To avoid SC1s, it is possible to transmit with a higher transmit power, or to send shorter packets. On the other hand, to avoid SC2s, a node could decrease its carrier sense threshold, in order to sense more nodes, thereby reducing the number of hidden nodes at the cost of deferring channel access more frequently. This distinction is necessary because these two types of staggered collisions each have a different cause, and thus each require a different modification to the link adaptation layer or packet scheduling algorithm.

Current LA techniques based on loss statistics, such as Auto Rate Fallback[1], perform poorly in the presence of collisions, because collisions are misinterpreted as channel errors[2]. In most of these algorithms, the modulation rate is lowered when packets are lost, resulting in longer subsequent packets. If the source of the problem is high network load and a large number of collisions, rather than poor channel conditions, this only exasperates the packet loss problem. Therefore differentiating between collision and channel error is an important problem as far as LA algorithms are concerned. To improve LA performance in the presence of collisions, various schemes have been proposed to distinguish collisions from channel errors on a per-packet basis[3][4]. In this paper, we are primarily interested in the *probability* of each type of loss rather than per-packet granularity in detecting collisions, since this is sufficient to determine expected throughput. It has been shown that an accurate estimate of the probability of collision can significantly improve LA[5]. Once probability of collision has been estimated, channel error probability can be easily inferred by using the overall loss statistics.

In this paper, we propose an approach for estimating various components of collision probabilities for 802.11 networks. Our motivation is to eventually apply such a scheme to

develop LA and cross-layer optimization algorithms in order to maximize throughput by appropriately tuning a wide range of parameters including contention window size, modulation rate, packet length, forward error correction, transmission power, and sensing threshold.

## II. THE BASIC APPROACH

Unlike in wired networks, each node in a wireless network observes a different medium depending on its location. As a result, standard local sensing alone cannot allow a sending node to determine network traffic levels at the intended receiver. For a node to estimate the probability that its next packet collides, it needs spatial information about network traffic.

Our approach to estimating collision probabilities is based on obtaining partial spatial traffic information. Our setup is an 802.11 network in infrastructure mode, with many nodes sending uplink traffic to the AP. Our goal is to estimate the uplink collision probabilities for all nodes. As such, we ignore downlink traffic since there are fewer APs than nodes, and they tend to be more spatially spread out; hence there are fewer collisions between downlink traffic as compared to uplink. Downlink traffic from one AP can potentially also collide with uplink traffic to another AP; however, since these collisions involve uplink transmissions, we assume the problem is dealt with by the nodes sending the uplink traffic. We also assume the traffic to be stationary over the period of time over which collision probabilities are being computed, and that the traffic of all nodes are independent. We do *not* assume the nodes to have any knowledge about packet lengths or traffic shape of other nodes.

The outline of our proposed scheme is as follows: All nodes collect local statistics, to be described shortly, about the traffic they sense. The AP then broadcasts the statistics it has collected to all of its associated nodes. Since this is a periodic broadcast from the AP, the overhead does not scale with the number of nodes or the number of packets. By comparing its own statistics with those from the AP, a local station is able to obtain a clearer picture of the spatial occupancy of the medium in order to estimate the probabilities of the different types of collisions. In the remainder of the paper, we refer to the node under consideration for the computation of the collision probability as “the station”, and refer to general nodes as “nodes”.

## III. TRAFFIC STATISTIC COLLECTION AND TRANSMISSION

The transmission of packets by the station is governed by the state of the medium which it can sense. In particular, it can only send when it senses the local medium as idle. However, the reception of the packets depends on the state of the medium around the destination, the AP. Therefore to estimate the probability of collision, the station must be able to compare the occupancy of the medium locally to that at the AP.

We model the traffic state of the medium at each node or AP as a zero-one process over time taking the value 0 when the channel is *idle*, and 1 when it is *busy*. We define busy

to mean that there is some transmission with received energy above the carrier sensing threshold of the node or AP, and idle to mean that there is no such transmission. We call this the busy-idle process at that node or AP. Because the sensing of transmissions requires a much lower SNR than the decoding, the busy-idle signal is much more robust to fading than statistics about successfully received packets. In practice, the collection of this signal may be imperfect because there is a non-zero chance of misclassifying the medium as busy or idle at any given time; however, errors caused by random noise are uncorrelated over samples, and can be filtered out using knowledge of the minimum packet length and interframe spacing. For the purposes of this paper, we assume a perfect signal to be obtainable.

The signal of interest is the vector valued signal  $BI(t) = [BI_{STA}(t), BI_{AP}(t)]$ , where  $BI_{STA}(t)$  and  $BI_{AP}(t)$  denote the busy-idle signals at the station and AP, respectively. The station has access to its own busy-idle signal, but it must acquire  $BI_{AP}(t)$  from the AP. Transmission of this complete waveform from the AP to the nodes could result in a significant amount of overhead. However, perfect resolution is not needed for reasonable estimation accuracy. It is only necessary to have at least one sample per backoff slot time, since nodes check the status of the medium at this resolution to determine when to send. Furthermore, since the information needed by all stations is the same, the AP can simply broadcast this information so that the overhead does not scale with the number of nodes or amount of traffic.

For an estimate of the number of bits that must be sent by the AP, we assume there are at most 1000 non-colliding packets per second heard by the AP. This results in 2000 edges in the busy-idle process per second. For  $10\mu s$  resolution, we can upper-bound the amount of information by a 2-state discrete time Markov chain with transition probability 0.02. The entropy rate of this process is 0.141 bits per sample, or 14.1 kbps. Because there is also regular transmission overhead for this data, the precise overhead will depend on how frequently it is sent, but if the AP were to send this information every few seconds at the lowest possible modulation rate, i.e. 1 Mbps for 802.11b, the overhead would be on the order of 2%.

## IV. ESTIMATING COLLISION PROBABILITIES

To compute the probability that its next packet experiences a collision, each node uses the available statistics to compute its probability of each type of collision, and combines them using the relation:

$$(1 - P_C) = (1 - P_{SC2}) \times (1 - P_{DC}) \times (1 - P_{SC1}) \quad (1)$$

where  $P_{SC2}$  is the probability the next packet experiences a staggered collision of type 2,  $P_{DC}$  is the probability the next packet experiences a direct collision given that it does not experience a staggered collision of type 2, and  $P_{SC1}$  is the probability that it experiences a staggered collision of type 1 given it does not experience either of the other types of collision. The order in which the probabilities occur in Eq. (1) matches with the natural order of events. Specifically, an SC2

TABLE I  
ABSOLUTE ERROR PERCENTAGE FOR  $P_{SC2}$  AS A FUNCTION OF SAMPLING PERIOD, AVERAGED OVER 300 NODES

inter-sample time ( $\mu$ s)	1	2	4	10	15	20	30
mean absolute error (%)	0.4	0.6	0.9	1.6	1.7	3.8	34.07

occurs when the channel is busy at the AP before the station starts to send, a DC occurs when another node starts at the same time, and an SC1 occurs when another node interrupts the station after the transmission has begun. The remaining lost packets are the result of channel errors.

In the subsections that follow, we will describe the estimation of each of the probabilities in Eq. (1). Once the station computes  $P_{SC2}$ ,  $P_{DC}$ , and  $P_{SC1}$ , it combines them via Eq. (1) to estimate its total probability of collision. The estimates are verified via NS simulations in Section V.

#### A. Probability of staggered collision of type 2

An SC2 for the station occurs when the station starts sending during a time in which the channel is already busy at the AP. If the station has access to the full busy-idle process at the AP, it can identify the times when SC2s occur because these are times when it starts sending while the AP is already busy. Even if the statistics do not cover all times or are imperfect, the station can identify all of the times it could possibly have started sending, and count what proportion of these would have experienced SC2s by examining the state of the busy-idle process at the AP at those times.

Table I shows the absolute value of percentage error in estimating  $P_{SC2}$  as a function of sampling period of the busy-idle signal at the AP, averaged over 300 nodes from 30 different NS-2 simulations. As seen, the mean absolute error is below 3% for resolutions higher than  $20\mu$ s, i.e. one sample per idle slot time. As expected, beyond  $20\mu$ s resolution, the error becomes increasingly large. For the remaining NS-2 simulations in this paper, we use a sampling period of  $10\mu$ s.

Further, if the binary-valued busy-idle signal at the station is replaced by a continuous-valued signal indicating the level of power sensed on the channel, the station can determine its potential busy-idle processes for different carrier-sensing thresholds. Note that finer quantization of this signal does not affect the overhead, since it is not transmitted. With this increased information, the station can determine the sensitivity of its probability of SC2 to its current sensing threshold. For example, if the continuous-valued version of the busy-idle signal at the station takes on a value slightly below the current sensing threshold immediately before the station sends a packet which experiences an SC2, then use of a lower threshold would have resulted in the station sensing the channel as busy immediately before its transmission; in this case the SC2 would have been avoided. Knowledge of this sensitivity could potentially lead to better LA algorithms than knowledge of  $P_{SC2}$  alone.

#### B. Probability of direct collision

A direct collision for a station occurs when another node begins its transmission at the same time as the station. This event cannot be directly observed by comparing the busy-idle

traces of the station and the AP since the transmission of the station itself will cause the AP to become busy, and there is no way for it to know whether there is also another node sending.

However, if some simplifying assumptions are made about the behavior of nodes, it is possible to arrive at a good approximation. For our approximation, we assume that all nodes are equally likely to send at any time in which they sense the channel as idle. In [6], Bianchi shows that nodes in a single collision domain with saturated traffic behave in this way when each time step is the length of a backoff slot. Increasing the time resolution does not significantly change this behavior. However, changing the rate of application layer traffic so that the transmit queue is not always backlogged can affect this behavior. We will show via simulations that our assumption results in reasonable estimates given that the application layer traffic is poisson.

For a direct collision to occur at time  $t$ , the AP must sense the channel as idle at time  $t - 1$ , and then hear the start of transmissions by both the station and another node. If the station is assumed to be equally likely to send at any time when the channel is idle around it, the probability of direct collision can be computed as the probability that some other node begins to send at time  $t$  when the channel is idle at both the station and the AP at time  $t - 1$ . To compute this probability, the station must exclude the times  $t$  which it itself sends, since it cannot determine if another node is sending. Let  $S(t)$  denote the indicator that the station is sending at time  $t$ . Then, using the busy-idle signals, the station can estimate  $P_{DC}$  as

$$P_{DC} = \frac{\sum_t \mathbb{I}\{BI(t-1) = [0, 0], BI_{AP}(t) = 1, S(t) = 0\}}{\frac{1}{T} \sum_t \mathbb{I}\{BI(t-1) = [0, 0], S(t) = 0\}} \quad (2)$$

where  $T$  is the length of a backoff slot and  $\mathbb{I}\{\cdot\}$  is the indicator function. The numerator gives the number of times another node starts to transmit when the station could have started its transmission, and the denominator gives the total number of opportunities the station had to transmit. The  $1/T$  is needed in the denominator because the station will only attempt to send 1 of every  $T$  time steps – those at the start of a backoff slot.

#### C. Probability of staggered collision of type 1

$P_{SC1}$  is the most difficult probability to estimate because it depends on the action of hidden nodes with respect to the station while the station is sending. But since the transmission of the station causes the medium to be busy at both the station and the AP, there is no way to observe this behavior.

To understand how to estimate  $P_{SC1}$ , we start by considering a simple scenario consisting of a single AP which hears all traffic, and two sets of grouped nodes which we call “local” and “hidden”. The nodes in each set can all hear members of their own set, but no nodes in one set can hear any of the nodes in the other. We then examine the additional phenomena we have to account for in the more general case where there are multiple APs, and a more complex connectivity graph between nodes, so that the set of hidden nodes to any given node may be unique to that node.

In the simple scenario, we assume there are two sets of nodes which cannot sense packets from each other. Thus, the processes determining the times the nodes from each set send packets are independent. Let  $\tau_l$  denote the rate at which local nodes begin to send given that they observe the channel to be idle, and let  $\tau_h$  denote the rate at which hidden nodes begin to send given that they observe the channel to be idle.

Then,  $\tau$ , the total probability that there is a packet heard by the AP given that the channel was idle in the previous time is given by

$$\tau = 1 - (1 - \tau_l)(1 - \tau_h). \quad (3)$$

From its local statistics, a local station can estimate  $\tau_l$  as

$$\tau_l = \frac{\sum_t \mathbb{I}\{BI_{STA}(t-1) = 0, BI_{STA}(t) = 1\}}{\frac{1}{T} \sum_t \mathbb{I}\{BI_{STA}(t) = 0\}} \quad (4)$$

Similarly, from the statistics broadcast from the AP, the local station can estimate  $\tau$  as

$$\tau = \frac{\sum_t \mathbb{I}\{BI_{AP}(t-1) = 0, BI_{AP}(t) = 1\}}{\frac{1}{T} \sum_t \mathbb{I}\{BI_{AP}(t) = 0\}} \quad (5)$$

Combining (3), (4), and (5), the station can estimate  $\tau_h$  from the pair of busy-idle signals.

The probability that a packet sent by the station avoids an SC1 is the probability that no hidden nodes send during the station's packet. This probability is given by  $(1 - \tau_h)^L$ , where  $L$  is the *effective length* of the packet for the hidden nodes, that is the number of opportunities the hidden nodes have to send during the successful transmission of the packet.

This is strongly related to the notion of *virtual slots* in [6]. In [6], Bianchi shows that a single collision domain 802.11 network with saturated traffic can be thought of as operating in discrete time where the ‘virtual’ time slots are of variable length, i.e. either a short slot which is the length of a backoff slot when no one is sending, or a long slot which is the length of a full transmission, ACK, and inter-frame spacing when a node sends a packet. In our scenario, there are two separate collision domains, each with independent notions of virtual time. The station's transmission will have a length of one virtual slot in its own collision domain, but will last for more virtual slots in the other collision domain.

In this simple scenario, the station can compute  $L$  locally as the length of its transmission – including inter-frame spacing and ACK – divided by length of a backoff slot, since in order for the transmission to be successful, all hidden nodes must remain silent, making all the virtual slots short backoff slots. Thus the station can easily compute  $P_{SC1}$  as

$$P_{SC1} = 1 - (1 - \tau_h)^L. \quad (6)$$

In a more general network topology with multiple AP's not hearing everything, and unconstrained node locations, there are multiple hidden nodes which hear different subsets of the other nodes, making the problem more difficult. There is no longer a single number  $\tau_h$  which characterizes the behavior of all the nodes of interest, nor is there a single  $L$  for all of these nodes. Further, the presence of nodes which are heard by both

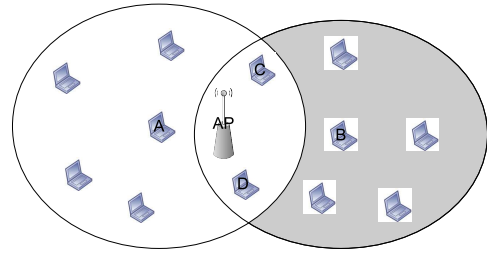


Fig. 1. The silencing of competing nodes: While node A is sending, only nodes in the shaded region can send, because the other nodes hear node A and are silenced.

the station and its hidden nodes causes a phenomenon we call the ‘coupling of transmission times’. We discuss these three issues in the following three subsections.

### C1. Time-variation of $\tau_h$

For a general network topology, we define  $\tau_h(t)$  as the probability that at least one of the hidden nodes with respect to the station starts to transmit in the next time instance. This quantity is time-varying because the hidden nodes are affected by other nodes which are not in the set of hidden nodes.

The greatest difficulty with estimating  $\tau_h(t)$  is that from the busy-idle signals, the station can only directly determine the network behavior while it is not sending. This is because when the node itself is sending,  $BI(t) \equiv [1, 1]$  regardless of what the hidden nodes are doing, since the AP and station both hear the station's transmission. In order to estimate  $P_{SC1}$ , the station needs to estimate  $\tau_h(t)$  while it is sending. We approximate this value as a constant which we denote by  $\tau_h^*$ . Similarly, we denote by  $\tau_h^{idle}$  the average value of  $\tau_h(t)$  while the medium is idle at both the AP and the station. This value is straightforward to observe from the busy-idle signals as

$$\tau_h^{idle} = \frac{\sum_t \mathbb{I}\{BI(t-1) = [0, 0], BI(t) = [0, 1]\}}{\frac{1}{T} \sum_t \mathbb{I}\{BI(t) = [0, 0]\}}. \quad (7)$$

As seen shortly, our approach for estimating  $\tau_h^*$ , and hence  $P_{SC1}$ , is to first estimate  $\tau_h^{idle}$ , and then scale it by an appropriate scaling factor,  $\bar{R} > 1$  to obtain  $\tau_h^*$ .

The primary factor which causes  $\tau_h^*$  to be greater than  $\tau_h^{idle}$  is what we call the silencing of competing nodes. An example of this is shown in Fig. 1 where the circles around nodes A and B represent their respective sensing/sensed ranges. We define the sensing range of node A as range of nodes which A can sense, and the sensed range as the range of nodes which can sense node A. For simplicity, in this discussion we assume the sensing and sensed ranges to be the same. If node A is sending, nodes C and D are silenced, since they can sense A. Therefore, the only nodes that can transmit are those in the shaded region. As a result, node B experiences less competition for the channel when A is sending than when A is not sending, since in the latter case B has to compete with all nodes within its sensing range.

We now elaborate on this process for a given hidden nodes B as shown in Fig. 1. To begin with, assuming a sufficiently large number of nodes spaced uniformly at random, the number of nodes competing with node B for the channel roughly scales

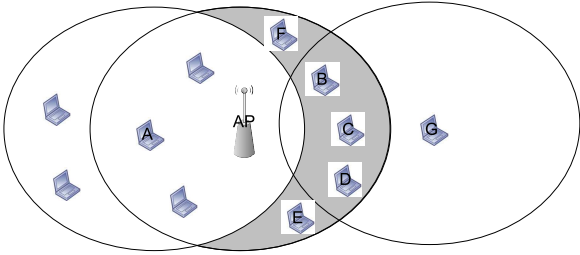


Fig. 2. The silencing of hidden nodes: If node G starts sending, nodes B through D are silenced and cannot cause a SC1 with A. Effectively, they observe a shorter packet, in virtual slots, than nodes E and F.

with the area within B's sensing range for which the channel is currently clear. Furthermore, the sending rate of node B is inversely proportional to the number of nodes it competes with. Putting these together, we conclude that the sending rate of a node B scales inversely with the area around it for which the channel is clear. The crescent-shaped shaded region in Fig. 1 corresponds to the portion of node B's sensing range for which the channel is potentially clear while the station A is transmitting. When A is not transmitting, the entire circle around B is potentially clear. Let  $R(d_B)$  denote the ratio of the area of the circle around B to the area of the shaded region, which is a function of  $d_B$ , the distance between A and B. Then the rate at which node B sends while node A is sending is given by:

$$\tau_B^* = \tau_B^{idle} R(d_B) \quad (8)$$

where  $\tau_B^{idle}$  denotes the rate at which node B sends when the station A is not sending.

A similar effect occurs for each of the hidden nodes with respect to the station. Therefore,  $\tau_h^*$ , the average probability that a hidden node sends in a given virtual slot while the station is sending, is given by:

$$\tau_h^* = \sum_{i \in \mathcal{H}} \tau_i^{idle} R(d_i) \approx \tau_h^{idle} \bar{R} \quad (9)$$

where  $\mathcal{H}$  is the set of hidden nodes to the station A, and  $\bar{R}$  is the average of  $R(d)$  over all these nodes.

The approximation in Eq. (9) assumes that there are no hidden nodes with significantly different distances and sending rates from the others, as this would potentially necessitate scaling one portion of the total transmission probability by a drastically different value. In practice, this is a reasonable assumption since due to the geometry constraints, hidden nodes have a limited range of possible distances from the station. Specifically, they must be far enough to not sense the station, but close enough to interfere at the AP. Since the station must be within about half of an interference range from the AP, the admissible region for hidden nodes with respect to the station is typically a slim, crescent-shaped region, similar to the shaded region in Fig. 2.

Since  $\bar{R}$  is unknown, the station must estimate it as the expectation of  $R(d)$  taken over the distribution of  $d$ . For the results in this paper, we assume that nodes are distributed according to a spatial poisson process. Depending on the application, it may be possible to have some other distribution, but regardless,  $\bar{R}$  can be pre-computed and stored.

## C2. Non-uniformity of $L$

The other key parameter used for the estimation of  $P_{SC1}$  is the effective length of the packet for the hidden nodes,  $L$ . Recall that for a hidden node with respect to the station, the effective length  $L$  of a packet sent by the station is the number of opportunities for the hidden node to interrupt it, or equivalently, the number of virtual slots that elapse for the hidden node during the transmission of the station's packet. If there are no nodes which are sensed by the hidden nodes but not by the AP,  $L$  would straightforwardly compute as mentioned above. However, in an actual network, there may be nodes exposed to the hidden nodes which are not sensed by the AP, but which silence some of the hidden nodes for a period of time; this gives these hidden nodes less opportunities to interrupt the packet sent by the station, making  $L$  effectively shorter for those hidden nodes. An example of this is shown in Fig. 2 where the circles around nodes A and G represent the range for which they can be sensed, and the circle around the AP represents its interference range. The shaded region denotes the region of hidden nodes with respect to the station A, namely nodes B through F, which are within the interference range of the AP yet outside the sensed range of A. If node G starts sending, hidden nodes B, C, and D are silenced, and thus experience fewer transmission opportunities than experienced by hidden nodes E and F, making A's packet effectively shorter for those nodes. It is useful to think of each node as having a different perception of virtual time. In this example, the transmission of node G causes nodes B, C, and D to freeze the virtual time in one long slot while virtual time continues to elapse in many short slots for nodes E and F. Thus, depending on the position and activity of exposed nodes such as G, there is a time- and space-varying scaling between effect packet length and real-time packet length at the hidden nodes.

For the station, A, the probability that a given hidden node, B, causes an SC1 can be expressed as

$$P_{SC1}^{(B)} = 1 - (1 - \tau_B^*)^{L_B(l)} \quad (10)$$

where  $l$  is the length of the packet in real time, and  $L_B(l)$  is the effective length of the packet sent by the station in virtual slots as observed by node B. The difficulty in directly applying Eq. (10) is that virtual time,  $L_B(l)$ , does not necessarily progress linearly with real time,  $l$ . If this were the case, plotting  $P_{SC1}$  versus  $l$  would yield an exponential according to Eq. (10). Through NS-2 simulation, we have empirically shown this not to be the case.

Specifically, Fig. 3 shows a plot of an example cumulative histogram of the "real" time,  $l$ , between the start of the station's packet and the start of the next packet sent by any of its hidden nodes. To obtain this plot we run NS-2 simulations for a network with 7 APs at fixed locations covering hexagonal cells, and 50 nodes placed at random according to a spatial poisson process. We fix the modulation rate of all nodes to 11 Mbps and send packets of maximum length for 802.11b, i.e. 2 kB or 1610 $\mu$ s. For each instant the station starts a transmission, we record the time until its next hidden node starts to transmit.



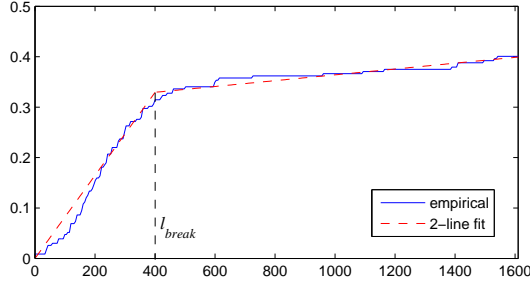


Fig. 3. Plot of the cumulative histogram of the real time,  $l$ , in  $\mu\text{s}$  between the start of the station's packet and the start of the next packet sent by any of its hidden nodes.

Fig. 3 is an example cumulative histogram of these values for a particular station.

The significance of Fig. 3 is that the duration of the packet sent by the station in real time,  $l$ , can be used as the value on the horizontal axis in order to look up  $P_{SC1}$  for the station on the vertical axis from the shown empirical curve. In particular,  $P_{SC1}$  is the probability that the next packet sent by any of the station's hidden nodes arrives before time  $l$ . Our overall approach to computing  $P_{SC1}$  is for the station to use the busy-idle signals at the station and AP to estimate this curve, and then to look up  $P_{SC1}$  based on its packet length. Specifically, we model this curve as a two-piece piecewise linear curve and estimate the slopes using the busy-idle signals.

For a single hidden node, B, the slope of the curve in Fig. 3 can be estimated as the derivative of  $P_{SC1}$  with respect to  $l$  in Eq. (10),

$$\frac{\partial}{\partial l} P_{SC1}^{(B)} = -[\ln(1 - \tau_B^*)](1 - \tau_B^*)^{L_B(l)} \frac{\partial L_B(l)}{\partial l}. \quad (11)$$

The key issue is that the rate at which virtual time elapses with respect to real time  $l$ , i.e.  $\frac{\partial L_B(l)}{\partial l}$ , is a not constant. This is because of the nature of the virtual slot lengths. In essence, virtual time can be thought of as consisting of many consecutive short, idle, slots together with isolated long, busy, slots.

It is useful to think of  $\frac{\partial L_B(l)}{\partial l}$  as a random function of  $l$ . To compute the expectation of  $\frac{\partial L_B(l)}{\partial l}$ , we split the realizations into two cases, depending on the state of the channel as observed by node B when the station begins its transmission:

- 1) If the station begins its transmission when node B observes the channel as idle, then for small  $l$ ,  $\frac{\partial L_B(l)}{\partial l}$  is relatively large, namely  $(1 \text{ virtual slot})/(\text{length of backoff slot})$ . This is because node B is observing idle backoff slots and has  $L_B(l) = 1$  opportunity to transmit per  $l = \text{length of backoff slot}$  seconds. For larger  $l$ , eventually a silencing node, such as G in Fig. 2, starts to transmit, decreasing  $\frac{\partial L_B(l)}{\partial l}$  to  $(1 \text{ virtual slot})/(\text{length of average busy slot})$ . This is because for node B only  $L_B(l) = 1$  virtual slot elapses during  $l = \text{length of busy slot}$  seconds. Let  $l_{break}$ , show in Fig. 3 denote the expected time at which this decrease occurs, i.e. the expected amount of time before a node

similar to G in Fig. 2 starts sending when the station, A, is sending.

- 2) If the station begins its transmission when node B observes the channel as busy, then it is equally likely that the station has started its transmission at any time during the long, busy slot. In this case,  $L_B$  is initially constant, and eventually increases rapidly when the current busy slot expires and idle slots resume. However, since this increase is equally likely to happen at any time, averaging over all of these cases yields the same average value  $\frac{\partial L_B(l)}{\partial l}$  for all  $l$ .

Combining the above two cases, we conclude that the random function  $\frac{\partial L_B(l)}{\partial l}$  has a higher expected value for small  $l$  than for large  $l$ . The actual value of this expectation depends on how often the channel is busy on average at the hidden node B, because this gives the weighting of the two cases as well as the actual value of each. A reasonable proxy for how often the channel is busy at the hidden node B is the proportion of time the busy-idle signal at the AP is busy, denoted by  $P(B_{AP})$ . This quantity can be readily computed at the station, as the mean of  $BI_{AP}$ .

The empirical curve of Fig. 3 shows the aggregate effect of this  $\frac{\partial L(l)}{\partial l}$  for all hidden nodes for the station. Specifically, taking all hidden nodes into consideration, Eq. (11) can be approximated as

$$\frac{\partial}{\partial l} P_{SC1} \approx -[\ln(1 - \tau_h^*)](1 - \tau_h^*)^{L(l)} \frac{\partial L(l)}{\partial l} \quad (12)$$

where  $L(l)$  denotes the average length of station's packet in virtual slots as observed by all the hidden nodes. Similar to  $\frac{\partial L_B(l)}{\partial l}$ , we expect  $\frac{\partial L(l)}{\partial l}$  and hence  $\frac{\partial}{\partial l} P_{SC1}$  to have a higher expected value for small  $l$  than for large  $l$ . This is in agreement with the shape of the empirical curve in Fig. 3.

Based on the above analysis, which predicts  $\frac{\partial L(l)}{\partial l}$  to have a higher value for small  $l$  than for large  $l$ , it is reasonable to approximate the empirical curve in Fig. 3 with a 2-piece piecewise linear function, with a steeper initial slope for  $l < l_{break}$ , and a less steep slope for  $l > l_{break}$ . We have empirically found  $l_{break} = 400\mu\text{s}$  to result in fairly accurate estimates of the curve in Fig. 3 in NS-2 simulations of 802.11b networks. Future work involves investigating ways for the station to estimate the break point based on local and AP statistics.

Having fixed  $l_{break}$ , the piecewise linear function in Fig. 3 is completely determined by 2 parameters: the slope for  $0 < l < l_{break}$ , denoted by  $m_1$ , and the slope for  $l_{break} < l < l_{max}$ , denoted by  $m_2$ . To estimate  $P_{SC1}$ , the station must somehow estimate these slopes from the busy-idle signals available to the station. Obtaining a reasonable approximation for the entire curve, rather than for one value of  $P_{SC1}$ , has the added advantage of allowing the station to determine the sensitivity of  $P_{SC1}$  with respect to packet length from the slope of the curve in Fig. 3. In general, the smaller the slope is at the current packet length, the less sensitive  $P_{SC1}$  is to packet length.

Rather than estimating  $m_2$ , we opt to estimate  $m_{avg}$ , the

average slope over the entire range of  $l$ , as

$$m_{avg} = \frac{(m_1 l_{break} + m_2 l_{max} - l_{break})}{l_{max}} \quad (13)$$

where  $l_{max}$  is the length of the longest possible packet the station can send, i.e. 2KB in our example.  $m_2$  can easily be computed from  $m_1$  and  $m_{avg}$ .

We have empirically found a reasonable model for the initial slope to be

$$m_1 = \alpha_0(P(B_{AP})) - \alpha_1(P(B_{AP})) \ln(1 - \tau_h^*) \quad (14)$$

and for the average slope to be

$$m_{avg} = \beta_0(P(B_{AP})) - \beta_1(P(B_{AP})) \ln(1 - \tau_h^*). \quad (15)$$

$\alpha_0$ ,  $\alpha_1$ ,  $\beta_0$ , and  $\beta_1$  are functions of  $P(B_{AP})$ , and are looked up from a table by the station. The first terms in Eqs. (14) and (15), i.e.  $\alpha_0$  and  $\beta_0$ , are independent of  $\tau_h^*$ , are absent in Eq. (12), and are discussed in detail in Section IV-C3.

From Eq. (12) and Fig. 3,  $\alpha_1$  and  $\beta_1$  are the average values of  $(1 - \tau_h^*)^{L(l)} \frac{\partial L(l)}{\partial l}$  for  $l < l_{break}$  and  $l < l_{max}$ , respectively. As noted earlier,  $\frac{\partial L(l)}{\partial l}$  depends on  $P(B_{AP})$ . Similarly, it can be argued that  $(1 - \tau_h^*)^{L(l)}$  depends on  $P(B_{AP})$ . This is because  $\tau_h^*$  is the average sending rate of the hidden nodes, and  $L(l)$  depends on the sending rate of the hidden node neighbors. Putting these together, we conclude that the quantity  $(1 - \tau_h^*)^{L(l)} \frac{\partial L(l)}{\partial l}$ , and thus  $\alpha_1$  and  $\beta_1$  are also dependent on  $P(B_{AP})$ .

When the average network traffic,  $P(B_{AP})$ , is low, nearly all hidden nodes experience idle channel conditions when the station begins to send. Thus,  $\frac{\partial L_B(l)}{\partial l} = (1 \text{ virtual slot}) / (\text{length of idle slot})$  for small  $l$ . In this case, it is also likely that  $\tau_h^*$  is small, so  $(1 - \tau_h^*)^{L(l)} = 1$  for small  $l$ . Therefore, for small  $P(B_{AP})$ ,  $\alpha_1$ , which corresponds to small  $l$ , i.e.  $l < l_{break}$ , is given by  $(1 \text{ virtual slot}) / (\text{length of backoff slot})$ . As  $P(B_{AP})$  increases, there are more long, busy slots, causing both  $(1 - \tau_h^*)^{L(l)}$  and  $\frac{\partial L(l)}{\partial l}$ , and thus  $\alpha_1$  to decrease. Similarly,  $\beta_1$  decreases with increasing  $P(B_{AP})$  due to a similar phenomenon. These observations are verified shortly via simulations shown in Table II of Section IV-C4.

### C3. Coupling of transmission times

We now justify the existence of  $\alpha_0$  and  $\beta_0$  in Eqs. (14) and (15). They are related to a phenomenon which we call the coupling of transmission times. When an intermediate node such as node C or D in Fig. 1 is sending, it silences both nodes A and B. As a result, while nodes A and B would have transmitted independently in the absence of intermediate nodes, in the presence of such a node, both of their available transmission times are reduced to times when nodes C and D are silent. This increases the number of staggered collisions between nodes A and B because they are more likely to send around the same time. The amount by which this increases the  $P_{SC1}$  is dependent on how often nodes such as C or D are sending. We have empirically found  $P(B_{AP})$  to be a reasonable proxy for this. For lower-traffic networks, this

TABLE II  
SAMPLE VALUES OF  $\alpha$ 'S AND  $\beta$ 'S

range of $P(B_{AP})$	$\alpha_0$	$\alpha_1$	$\beta_0$	$\beta_1$
.75-.8	0	5e-2	0	1.91e-2
.8-.85	0	5e-2	0	1.78e-2
.85-.88	1e-4	5e-2	1e-4	1.29e-2
.88-.9	1e-4	5e-2	1e-4	1.30e-2
.9-.92	2e-4	3.96e-2	2e-4	9.48e-3
.92-.94	3e-4	3.23e-2	2e-4	8.10e-3
.94-.96	6e-4	2.41e-2	3e-4	4.75e-3
.96-.98	9e-4	2.00e-2	4e-4	2.17e-3

effect is negligible, but for higher traffic networks with larger values of  $P(B_{AP})$  this becomes more pronounced. Thus  $\alpha_0$  and  $\beta_0$  should increase with  $P(B_{AP})$  as verified shortly via simulations shown in Table II of Section IV-C4.

### C4. Estimating $\alpha_0$ , $\alpha_1$ , $\beta_0$ , and $\beta_1$

In general, computing closed-form analytical expressions for  $\alpha_0$ ,  $\alpha_1$ ,  $\beta_0$ , and  $\beta_1$  as functions of  $P(B_{AP})$  is a non-trivial task. Instead, we pre-compute them for various ranges of  $P(B_{AP})$  via regression from simulation data using the the NS-2 simulation package, which we have modified to collect the busy-idle signal at the AP and nodes as well as record the fate of each packet – DC, SC1, SC2, or no collision. We use the topology described in Section IV-A, with all nodes sending poisson application layer traffic at a fixed rate, which varies over different simulations. We use 13 different rates, ranging from 2 kB/s to 120 kB/s, and repeat each rate 5 times with different random topologies. From each of the 65 trials, we collect the measured data, namely the busy-idle waveform, at the center AP and all the nodes in its cell. We also record the empirical cumulative histograms for each station, i.e. the solid curve in Fig. 3 to use as ground-truth. We then choose a random subset of the over 400 stations from the 65 trials to be used for training data. We optimize  $m_1$  and  $m_2$  for each station to achieve the minimum squared distance between the 2-line approximation and the empirical curve. For each of the training stations, we use the optimal  $m_1$  and  $m_2$ , as well as the local estimates of  $\ln(1 - \tau_h^*)$  to perform a linear regression using Eqs. (14) and (15) in order to determine the values of  $\alpha_0$ ,  $\alpha_1$ ,  $\beta_0$ , and  $\beta_1$ . In practice, nodes store a table of values for  $\alpha_0$ ,  $\alpha_1$ ,  $\beta_0$ , and  $\beta_1$  and look them up the based on the observed  $P(B_{AP})$ .

An example portion of this table is shown in Table II. As expected,  $\alpha_1$  and  $\beta_1$  decrease with  $P(B_{AP})$  while  $\alpha_0$  and  $\beta_0$  increase with it. Also, for  $P(B_{AP}) < 0.9$ ,  $\alpha_1 = 1/(20\mu s) = (1 \text{ virtual slot}) / (\text{length of backoff slot})$ . Since  $\ln(1 - \tau_h^*)$  is on the order of  $10^{-2}$ , all the terms in Eqs. (14) and (15) are of comparable size.

Fig. 4 shows the resulting average normalized error in the estimate of the curve in Fig. 3, defined as  $|y^{estimate} - y^{empirical}| / y^{empirical}$ , for two bins of values of  $P(B_{AP})$ , namely 0.6 to 0.7 and 0.92 to 0.94. The averaging is done over multiple stations in multiple topologies with similar mean values of  $B_{IAP}$ . As seen, the estimation accuracy improves with packet length. The performance for lengths shorter than  $200\mu s$  is unimportant, since this is below the minimum packet length including inter-frame spacing for 802.11b. For longer

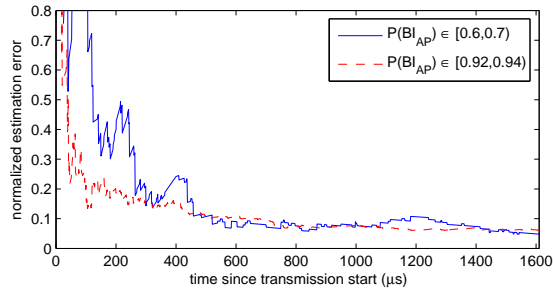


Fig. 4. Average absolute value of normalized estimation error,  $|(P_{SC1}^{estimate} - P_{SC1}^{actual})/P_{SC1}^{actual}|$ , for two traffic levels.

packets, errors become less than 10%.

Having estimated the probabilities of each type of collision, we can now combine them to find the total  $P_C$  using Eq. (1).

### V. SIMULATION RESULTS

To verify the above analysis, we use the NS-2 simulation package, which we have modified as described above. We post-process the busy-idle signals generated by NS in MATLAB to arrive at estimates of the various collision probabilities. We repeat the 65 trials of Section IV-C4 with different random seeds. Each run lasts 30 seconds. The estimates are computed using the busy-idle signals from the first 10 seconds, while ground truth is obtained by empirically counting the number of packets incurring each fate over the full 30 seconds.

Fig. 5(a) shows a cumulative histogram of the absolute error in estimating  $P_C$  for three different values of  $P(B_{AP})$ . Data for other traffic levels is not shown, but performance is similar. Figs. 5(b)-(d) show the error in each component of  $P_C$  for the same three traffic levels. As seen,  $P_{SC2}$  has a consistently small error. Due the way collisions are counted, SC2s are the dominant type of collision for higher traffic scenarios, making precise estimation of  $P_{DC}$  and  $P_{SC1}$  more difficult and less important as  $\text{mean}(BI_{AP})$  increases. In particular, for  $P(BI_{AP}) > 0.9$ ,  $P_{SC2}$  is almost 1, and there is not a sufficient number of packets which are not classified as SC2s for even the empirical count of  $P_{DC}$  or  $P_{SC1}$  used as ground truth to be accurate. It can be seen that while the error in estimating  $P_{DC}$  and  $P_{SC1}$  grows as traffic increases, the total error in  $P_C$  does not. In general, 90% of the time, the  $P_C$  estimate is within 20%, and 50% of the time, it is within 10%.

### VI. CONCLUSION AND FUTURE WORK

In this paper, we have presented a classification of the types of collisions that occur in WLANs, and proposed a method for accurately estimating them locally at any node via sharing spatial information about network traffic. Future work includes implementation and verification using actual hardware as well as development of algorithms that use this additional information to improve throughput through intelligent adaptation of parameters such as modulation rate, packet length, FEC, carrier sense threshold, contention window, and transmit power.

### REFERENCES

[1] Ad Kamerman and Leo Monteban, "WaveLAN-II: A High-Performance Wireless LAN for the Unlicensed Band", *Bell Labs Technical Journal*, Vol.2, No.3, Summer 1997, pp.118-133.

[2] Sunwoong Choi, Kihong Park, and Chong-kwon Kim, "On the Performance Characteristics of WLANs: Revisited", in *Proc. of ACM SIGMETRICS 2005*, Banff, Alberta, Canada, June 2005.

[3] Jongseok Kim, Seongkwan Kim, Sunghyun Choi, and Daji Qiao, "CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs", in *Proc. of IEEE INFOCOM 2006*, Barcelona, Spain, April 2006.

[4] Qixiang Pang, Soung C. Liew, and Victor C. M. Leung, "Design of an Effective Loss-Distinguishable MAC Protocol for 802.11 WLAN", *IEEE Communication Letters*, Vol. 9, No. 9, pp. 781-783, September 2005.

[5] Hyogon Kim, Sangki Yun, Heejo Lee, Inhye Kang, and Kyu-Young Choi, "A simple congestion-resilient link adaptation algorithm for IEEE 802.11 WLANs", in *Proc. of IEEE GLOBECOM 2006*, San Francisco, California, November 2006.

[6] Giuseppe Bianchi, "Performance Analysis of the 802.11 Distributed Coordination Function", *IEEE JSAC*, Vol. 18, No. 3, March 2000, pp.535-547.

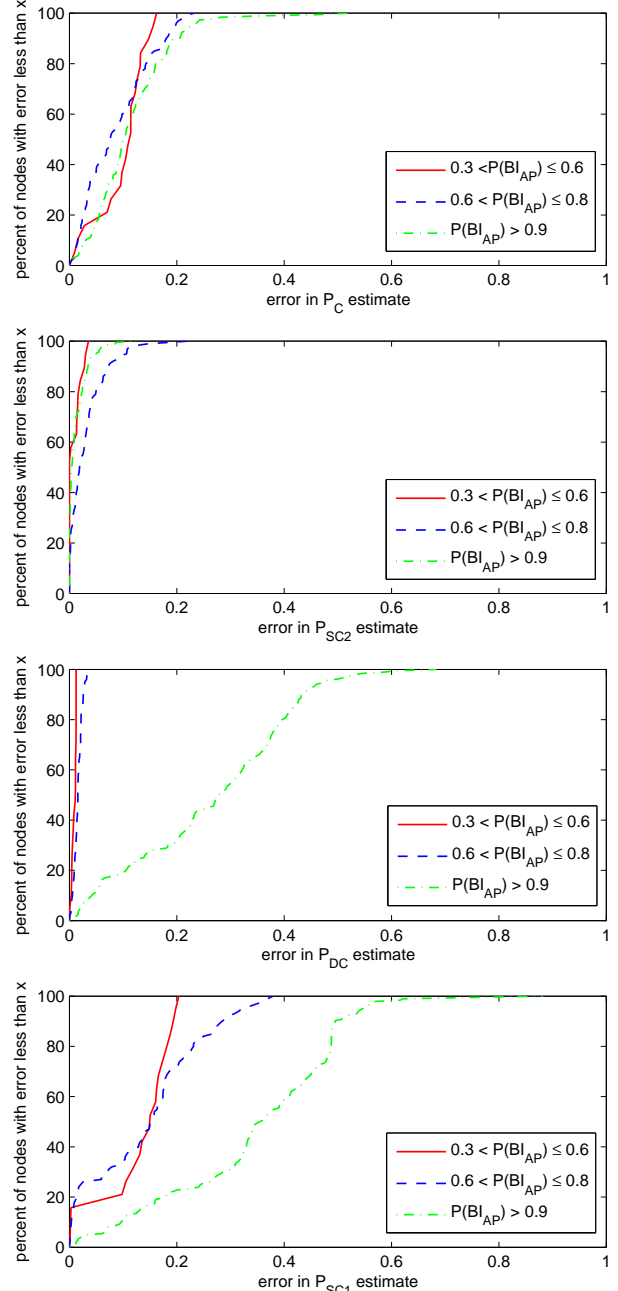


Fig. 5. Cumulative histogram of the estimation errors in (a) $P_C$ , (b) $P_{SC2}$ , (c) $P_{DC}$ , and (d) $P_{SC1}$  for three traffic levels.