

Local Estimation of Collision Probabilities in 802.11 WLANs with Hidden Terminals

Michael Krishnan, Sofie Pollin, and Avideh Zakhor
Department of EECS, U.C. Berkeley

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 goal is to design a practical approach, 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

802.11 WLANs are increasingly being used for applications with stringent performance requirements due to their ease of deployment. Even though the maximum physical layer rates are increasing with the introduction of new generations of 802.11 WLAN such as 802.11n, the effective throughput delivered to the application layer remains low. The problem stems from an inability to cope with the complexities of the wireless channel due to fading, collisions, and hidden node or co-channel interference. Nodes cannot distinguish one type of loss from another because the symptoms are the same, namely a lost packet. Each type of loss, however, requires different specific actions to maximize throughput.

Fading is a measure of the inherent unreliability of the wireless channel, causing low reception quality of transmitted packets. Packets incurring errors due to poor channel fades are said to experience *channel errors*. Link adaptation (LA) is commonly used to adapt the modulation and coding levels of each transmission in order to improve the error performance in case of low reception quality. Channel errors can also be reduced by forward error correction at the application layer.

Collisions happen when multiple transmissions occur at the same time. They are more likely to happen when there are many nodes in the network with large numbers of packets to send. 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 in the future. 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 as a natural result of the 802.11 Distributed Coordination Function (DCF) when two nodes finish their backoff at the same time. 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.

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 transmitting at exactly the same time. As such, we refer to these collisions as *staggered collisions (SCs)*. The distant nodes causing these collisions are typically associated with different Access Points (APs) on the same channel resulting in co-channel interference. Due to the widespread use of wireless technology, many wireless networks currently coexist in space, and as such, co-channel interference increasingly impacts the network performance. 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 hears the CTS message and avoids collision. This scheme however, is rarely used in practice due to the large overhead.

To protect an ongoing transmission of a given node from staggered collisions, it is possible to transmit with a higher transmit power, or to send shorter packets in order to decrease the probability that a hidden node transmits during the packet. On the other hand, to prevent a given node from transmitting during an ongoing transmission of a distant node, it is possible to decrease the carrier sense threshold; this causes a node contending for the channel to sense more nodes, thereby reducing the number of hidden nodes at the cost of deferring channel access more frequently. It is desirable to distinguish between the above two types of staggered collisions, since they each require a different modification to the link adaptation layer or packet scheduling algorithm. Hence, 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 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.

Current LA techniques based on loss statistics, such as Auto

Rate Fallback (ARF) [1], perform poorly in the presence of collisions [2], because collisions are misinterpreted as channel errors. 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. 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.

There is a growing body of literature on dealing with the spatial nature of wireless networks. In [6] Li *et. al.* model the throughput of an 802.11 network using full spatial information. Their approach is from a network point of view, while ours involves individual nodes locally estimating collision probabilities based on limited spatial information. In [7], a method is proposed to identify causes of “starvation” when a node achieves zero throughput, assuming saturation conditions in the network. However, in order to be useful in practical scenarios, it is desirable to determine the proportion of packets experiencing each type of loss for a broad range of traffic scenarios.

A great deal of work has also recently been reported in the literature on adapting transmission power and carrier sense threshold to optimize network throughput. In [8], Yang *et. al.* show that when adapting the carrier sense threshold, the transmission power should be adapted so that the ratio of the two remains constant in order to avoid starvation. They then propose an algorithm to adapt both jointly in order to maximize the average throughput in the network. However, they make simplifying assumptions on the network interactions, focus primarily on average throughput, and assume that all nodes in the networks use the same transmission parameters. In [9], a method is proposed to locally tune the transmission rate and sensing range using a greedy search algorithm. Similar to the early implementations of LA algorithms, their algorithm involves nodes using a higher throughput configuration for a probation period and reverting to more conservative settings if this decreases throughput. Clearly, their performance could be more robust if they were able to determine the probability of each type of packet loss.

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 SCHEME

In wired networks, it is possible to deploy collision *detection* rather than collision avoidance because the sending node is essentially able to hear the same channel as the receiving node. However, in wireless networks, each node observes a different medium depending on its location. Not only does a sending node hear its own signal much more strongly than any other node, but also it may not be able to hear other transmissions from hidden nodes interfering at the destination of its packet. No amount of sensing the channel can 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 some spatial information about network traffic.

Our approach to estimating collision probabilities is based on obtaining partial spatial traffic information. The setup is an 802.11 network in infrastructure mode, with many nodes sending uplink traffic to the AP. For now we ignore downlink traffic since there are fewer APs than nodes and they tend to be more 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, the problem is at least partially 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 probability is 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 about the “level of 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 then 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 the spatial occupancy of the medium.

To compute the probability that its next packet experiences a collision, each node uses the available statistics to compute its collision probabilities for the 3 different types, 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 occurs when the channel is busy at the AP before the

node under consideration starts to send, a DC occurs when another node starts at the same time, and an SC1 occurs when another node interrupts the node under consideration after the transmission has begun. The remaining lost packets are the result of channel error. In the remainder of the paper, we refer to the node under consideration for the computation of these probabilities as “the station”, and refer to general nodes as “nodes”.

III. THE SIMPLE CASE

We start with 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. In section IV, we examine 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.

A. Probability of direct collision

If there is only a single collision domain – that is, there are no hidden nodes – P_{SC1} and P_{SC2} are both zero. In this case, the total collision probability, P_C , is the same as the direct collision probability, P_{DC} , which can be computed as in [10]. In [10], 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. We refer to the short, unoccupied slots as *idle slots* and the long, occupied slots as *busy slots*. If we denote the total number of busy slots and idle slots over a fixed time window as B and I respectively, then the probability that the network is busy at any given time slot is independent of all other time, slots and is given by

$$\tau = \frac{B}{B + I}. \quad (2)$$

The probability that the station experiences a direct collision is the probability that the slot the station begins to send in is also occupied by another node. Bianchi shows that if the traffic is saturated, nodes can be modeled as being equally likely to send in any slot, and this assumption also roughly holds for unsaturated traffic which is nearly poisson. So under these conditions, P_{DC} is the same as the probability that a slot is occupied at a random time when the station is not sending, i.e.

$$P_{DC} = \frac{B - S}{B + I - S} \quad (3)$$

where S is the number of slots in which the the station is sending.

For a general network with hidden nodes, the values B and I are not the same for every node, since each node will hear a different subset of the nodes in the network. In this case, the values of B and I that are relevant to computing collision

probability are those at the destination of the packet, i.e. the AP. Let B_{AP} and I_{AP} denote the number of busy and idle slots at the AP, I_{STA} denote the number of idle slots at the station, B_{STA} denote the number of busy slots at the station in which it is not sending, and S_{STA} denote the number of busy slots at the station in which it is sending. Then, Eq. (3) can be rewritten as:

$$P_{DC} = \frac{B_{AP} - S_{STA}}{B_{AP} + I_{AP} - S_{STA}}. \quad (4)$$

In order for every node to be able to compute this, the AP periodically broadcasts its counts B_{AP} and I_{AP} to all its associated nodes. The local nodes can then use these values along with their local knowledge of S_{STA} to estimate P_{DC} . This estimation does not require the nodes or AP to be able to decode any packets, rather, they only need to be able to sense them. As a result, this method is relatively insensitive to the fading state of the channel.

B. Probability of staggered collision of type 1

The key here is that *in order for the station to compute the probability of collisions, it needs some spatial information about the network traffic*. In the case of P_{DC} , this amounts to the station having some knowledge of how often the channel is occupied at destination, which is the AP. To estimate P_{SC1} , the station needs to know how often its hidden nodes are sending. This is a significantly harder problem because no single node has this information. Periodic broadcast by all the nodes in the network in order to infer this information would be overhead intensive. However, as we show shortly, it is possible for a node to estimate this information by comparing the statistics collected at the AP, namely B_{AP} and I_{AP} , to those collected locally, namely B_{STA} , I_{STA} , and S_{STA} .

When the station receives the broadcast statistics from the AP, it is potentially able to estimate the rate at which hidden nodes are sending and thus estimate P_{SC1} . Intuitively, if $B_{AP} > B_{STA}$, then the station infers that there must be hidden nodes and further that their probability of sending must be related the the size of the difference between B_{AP} and B_{STA} . In the scenario under consideration in this section, 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. In any given time slot, the probability of sending is defined as τ_l for the local nodes, and τ_h for the hidden nodes. This means the total probability that there is a packet heard by the AP, τ , is given by

$$\tau = 1 - (1 - \tau_l)(1 - \tau_h). \quad (5)$$

From its local statistics, a local node can estimate τ_l as

$$\tau_l = \frac{B_{STA} + S_{STA}}{B_{STA} + S_{STA} + I_{STA}}. \quad (6)$$

Similarly, from the statistics broadcast from the AP it can estimate τ as

$$\tau = \frac{B_{AP}}{B_{AP} + I_{AP}}. \quad (7)$$

Combining (5), (6), and (7), the local node can estimate τ_h as

$$\tau_h = 1 - \left(\frac{I_{AP}}{B_{AP} + I_{AP}} \times \frac{S_{STA} + B_{STA} + I_{STA}}{I_{STA}} \right). \quad (8)$$

The probability that a packet sent by the station avoids a staggered collision of type 1 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 length of the packet in virtual slot times as observed by the hidden nodes. Even though slot times are in general different for local and hidden nodes, all slots during a successful packet must be idle at the hidden node. Thus node A can compute L locally as the length of its transmission – including inter-frame spacing and ACK – divided by length of an idle slot.

Thus the node can estimate the probability of a staggered collision of type 1 as

$$P_{SC1} = 1 - (1 - \tau_h)^L \quad (9)$$

where τ_h is estimated from Eq. (8) using the measured quantities collected locally along with those broadcast from the AP.

C. Probability of staggered collision of type 2

An SC2 for the station occurs when the station starts sending during a slot in which the channel is already busy at the AP. Since the station is equally likely to send in any slot for which it senses the previous slot as idle, and experiences an SC2 if the channel is busy at the AP during this time, this is the proportion of idle slots at the station which do not correspond to idle slots at the AP. Since in this section we assume the AP hears all traffic, the set of slots which are idle at the AP must be a subset of the slots which are idle at the station. P_{SC2} can thus be estimated as

$$P_{SC2} = \frac{I_{STA} - I_{AP}}{I_{STA}} \quad (10)$$

Once the station computes P_{SC2} , P_{DC} , and P_{SC1} , it combines them via Eq. (1) to estimate its total probability of collision.

D. Simulation results

To verify the above analysis, we modify the NS-2 simulation package to facilitate the collection of the required statistics at the AP and nodes. We then post-process the statistics in MATLAB to arrive at estimates of the various collision probabilities. Fig. 1 shows estimates of P_{DC} , P_{SC1} , P_{SC2} , and P_C as a function of time for a station in a 3 node scenario using the results in this section. Of the remaining two nodes, one is assumed to be local to the station, and the other one hidden. Starting at $t = 10$, packets are assumed to arrive at the send queue of each node via a poisson process with constant rate. After this they undergo the standard 802.11 backoff scheme, so the actual send times are not precisely poisson. The rate of the arrival process is relatively low or else the probability of staggered collisions would have approached one. As seen, the estimates and the actual values of the various probabilities match closely.

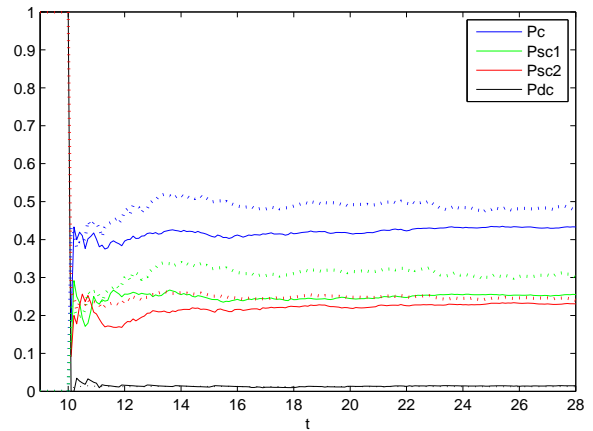


Fig. 1. NS-2 simulation results comparing actual and estimated values of various collision probabilities. The dotted lines denote estimates while the solid lines correspond to empirical counts. The dotted black line corresponding to the P_{DC} estimate is difficult to see as it more or less coincides with the actual count in the solid black line.

IV. EXTENSION TO MORE COMPLEX NETWORKS

In the simple scenario of Section III, the behavior of the hidden nodes is inferred from locally measurable statistics and some information from the AP, namely the number of busy and idle slots, B_{AP} and I_{AP} . However, in a more complex scenario with unconstrained node locations and multiple APs not hearing everything, it is considerably more difficult to infer the level of traffic of nodes which are hidden to the station. There are two reasons behind this:

- 1) First, the station may be affected by *exposed nodes*, or nodes which it can sense but the AP cannot. This drastically changes the computation of P_{SC2} as compared to Section III since the times when the station and the AP sense the channel as idle are less correlated. Thus knowledge of the *number* of busy and idle slots at the AP and station is not sufficient.
- 2) Second, unconstrained node location results in a more complex connectivity graph of the network. For example, for two nodes that are hidden to each other, the existence of common nodes that can be heard by both results in statistical dependency between their transmission times. As a result, the behavior of a hidden node with respect to the station should be modeled differently depending on whether or not the station is sending. As seen later, this has a significant effect on P_{SC1} estimation.

The above factors do not affect the P_{DC} estimation, but they do make computing probabilities of staggered collisions much more complex than in the simple scenario studied in Section III.

A. P_{SC2} estimation in a more complex network

In computing P_{SC2} for the simple scenario of Section III, the station takes advantage of the fact that the idle slots at the AP are a subset of the idle slots at the station. However, in the more complex scenario of this section, the presence of exposed nodes, which are heard by the node and not by the AP, invalidates this assumption. Our approach to remedy this loss of perfect correlation is to collect additional information. In particular, we

can model the traffic state 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 call this the busy-idle process at the particular node or AP. To estimate P_{SC2} , the station must estimate the joint statistics of this process at the AP and station, not merely the marginals at each, which is all that is captured in B and I . Therefore, temporal information is needed to relate these processes to one another.

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 looking at the state of the busy-idle process at the AP at those times. Further, if the binary-valued busy-idle signal at the node 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. With this information, the node 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 node takes on a value slightly below the current sensing threshold immediately before the station sends a packet, then 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.

The above procedure assumes that the station has access to the entire binary valued busy-idle process at the AP over time. While this is desirable from an estimation accuracy point of view, the sending of this complete waveform from the AP to the node could result in a significant amount of overhead. As seen shortly, perfect resolution is not needed for reasonable estimation accuracy; furthermore, this overhead does not scale with the number of nodes or traffic.

Fig. 2 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 higher resolution 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 $10\mu s$.

We now present an analysis of the amount of overhead associated with sending this sampled busy-idle signal. For a rough 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 0.0014 bits per sample, or 1.75 kilobytes per second.

Even though this is more overhead than sending only busy and idle slot counts, it is still relatively insignificant compared to the total load of the network. If the AP were to send this much information at the lowest possible modulation rate, i.e. 1 Mbps for 802.11b, the overhead would amount to 3% of the available time.

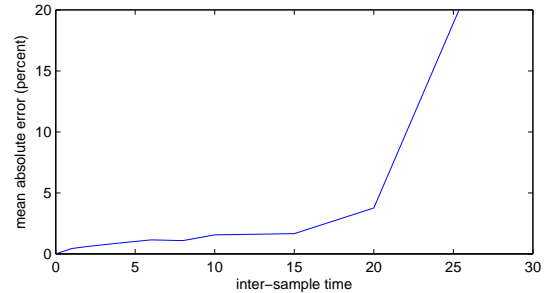


Fig. 2. Absolute error percentage for P_{SC2} as a function of sampling period averaged over 300 nodes

B. P_{SC1} estimation in a more complex network

While the busy-idle waveform at the AP can be used to estimate P_{SC2} in a complex scenario, it is insufficient for estimating P_{SC1} . This is because to estimate P_{SC1} , the station must estimate the probability that one of its hidden nodes starts to transmit during its own transmission. However, the behavior of the hidden nodes during this period is observable by neither the station nor the AP. Nevertheless, we propose an appropriate estimation process using the same information, namely the busy-idle signal at the station and at the AP. To do this, we need to address the following three issues:

- 1) Time-variation of τ_h , the rate at which hidden nodes send.
- 2) Non-uniformity of L , the length of the packet in virtual slots as observed by the hidden nodes.
- 3) Coupling of transmission times.

We address these in the following subsections.

B1. Time-variation of τ_h

A key parameter in computing P_{SC1} is the probability that one of the hidden nodes to the station starts to send in a given virtual slot while the station is transmitting. The difficulty is that this probability, denoted by τ_h^* , is greater than the average probability that one of the hidden nodes to the station start to send in a given virtual slot while the station is idle, denoted by τ_h^{idle} . This is because while the station is sending, it silences all the nodes in its interference range. This includes some of the nodes which typically contend with the hidden nodes for the channel. An example of this can be seen in Fig. 3 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.

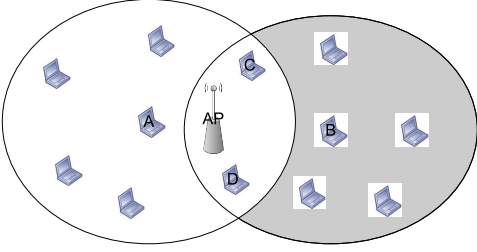


Fig. 3. The silencing of competing nodes: While node A is sending, only nodes in the shaded region can send, because the other nodes hear node A and are silenced.

Our approach to compute τ_h^* consists of two steps; we first estimate τ_h^{idle} ; then we scale it by the relative increase in send probability between times when the station is sending and when it is idle. We now elaborate on this process for a given hidden node B as shown in Fig. 3. 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 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. 3 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 crescent-shaped shaded region to that of the circle, where d_B is the distance between A and B; then the rate at which node B sends while node A is sending is given by:

$$\tau_B^* = \tau_B^{idle} R(d_B) \quad (11)$$

where τ_B^{idle} denotes the rate at which node B sends when the station is not sending.

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

$$\tau_h^* = \sum_{i \in \mathcal{H}} \tau_i^{idle} R(d_i) \approx \tau_h^{idle} \bar{R}(d) \quad (12)$$

where \mathcal{H} is the set of hidden nodes to the station A, and $\bar{R}(d)$ is the average of $R(d)$ over all these nodes; τ_h^{idle} is estimated in a similar way to τ_h in Eq. (5), with τ replaced by the rate of rising edges in the busy-idle signal of the AP, and τ_l replaced by the rate of simultaneous rising edges in the busy-idle signal of the AP and station. The approximation in Eq. (12) assumes that there are no hidden nodes with significantly different distances and sending rates than 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. 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 a slim, crescent-shaped region, shown in Fig. 4.

The average distance from the station, A, to the hidden nodes is an unknown quantity, and can be estimated by integrating over the distribution of d . This distribution is dependent on the distance between the station and the AP as well as the location of the hidden nodes within their admissible region. For our simulations in this paper, we assume the distance between the station and the AP to be known through some protocol. To estimate $R(d)$, we further assume that hidden nodes are uniformly distributed in the admissible region. If the distance between the station as the AP is not known, one can assume a distribution on this distance as well. $R(d)$ can then be pre-computed and stored at the station.

B2. Non-uniformity of L

Another important parameter used for the estimation of P_{SC1} is the length of the packet in virtual slots as observed by the hidden nodes, L . In the simple scenario of Section III, L is straightforward to compute, because during a successful transmission by the station, all nodes must be silent, and L virtual time slots elapse at a constant rate of one per backoff slot length, namely $20\mu s$ for 802.11b. Thus there is a constant scaling factor between the length of the packet in real time, denoted by l , and the length in virtual time, denoted by L . In the more complex scenario of this section, there may be nodes exposed to the hidden nodes which effectively lengthen some of the virtual time slots for some of the hidden nodes. An example of this is shown in Fig. 4 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 hidden nodes with respect to the station A, namely nodes B through F, which are within the sensing range of the AP yet outside the sensed range of A. If node G starts sending, nodes B, C, and D experience a single busy virtual time slot which, in real time, corresponds to multiple idle virtual time slots experienced by E and F. Thus, depending on the position and activity of exposed nodes, there is a time- and space-varying scaling between virtual time slots and real time at the hidden nodes.

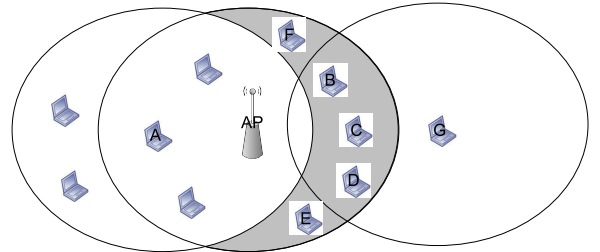


Fig. 4. The silencing of hidden nodes: If node G starts sending, nodes B through D are silenced and cannot cause a SC1 with A. Effectively, they observe a shorter packet, in virtual slots, than nodes E and F.

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

$$P_{SC1}^{(B)} = (1 - \tau_B^*)^{L_B(l)} \quad (13)$$

where $L_B(l)$ is the length of the packet sent by the station in virtual slots as observed by node B. The difficulty in directly applying Eq. (13) 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. (13). We have empirically shown this not to be the case. Specifically, Fig. 5 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 μ s. For each instant the station starts a transmission, we record the time until its next hidden node starts to transmit. Fig. 5 is an example cumulative histogram of these values for a particular station.

The significance of Fig. 5 is that the duration of the packet sent by the station 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 estimate this curve based on the busy-idle signal at the station and AP, and then to look up P_{SC1} based on its packet length.

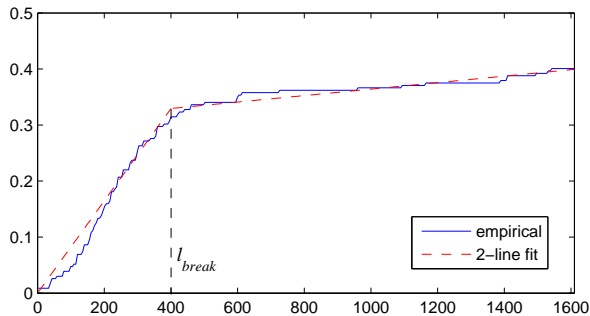


Fig. 5. Plot of the cumulative histogram of the real time, l , in μ s between the start of the station’s packet and the start of the next packet sent by any of its hidden nodes.

For a single hidden node, B, the slope of the curve in Fig. 5 can be estimated as the derivative of P_{SC1} with respect to l in Eq. (13),

$$\frac{\partial}{\partial l} P_{SC1}^{(B)} = -[\ln(1 - \tau_B^*)](1 - \tau_B^*)^{L_B(l)} \frac{\partial L_B(l)}{\partial l}. \quad (14)$$

The key difference between the scenario in this section and the simple one of Section III is that here $\frac{\partial L_B(l)}{\partial l}$ 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 $1/(\text{length of idle slot})$, since node B is observing idle slots. For larger l , eventually a silencing nodes such as G in Fig. 4, starts transmitting, decreasing $\frac{\partial L_B(l)}{\partial l}$ to $1/(\text{length of average busy slot})$. Let l_{break} denote the expected time at which this decrease occurs, i.e. the expected amount of time before a node similar to G in Fig. 4 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 is initially constant, and eventually increases rapidly when the current busy slot expires and idle slots resume. However, since this 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, 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 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, since it has access to the busy-idle signal at the AP.

The empirical curve of Fig. 5 involves 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. (14) can be approximated as

$$\frac{\partial}{\partial l} P_{SC1} \approx -[\ln(1 - \tau_h^*)](1 - \tau_h^*)^{L(l)} \frac{\partial L(l)}{\partial l} \quad (15)$$

where $L(l)$ denotes the average length of station’s packet as observed by all the hidden nodes. Based on the above analysis, it is reasonable to approximate the empirical curve in Fig. 5 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 $l_{break} = 400\mu$ s to result in fairly accurate estimates of P_{SC1} 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} , we now describe a way of estimating the slopes, m_1 and m_2 , for the piecewise linear function of Fig. 5

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. 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

$$m_{avg} = \frac{(m_1 l_{break} + m_2 l_{max} - l_{break})}{l_{max}} \quad (16)$$

where l_{max} is the length of the longest possible packet the station can send. Re-arranging (16), m_2 can be computed as

$$m_2 = \frac{m_{avg} l_{max} - m_1 l_{break}}{l_{max} - l_{break}}. \quad (17)$$

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

$$m_1 = \alpha_0(P(B_{AP})) - \alpha_1(P(B_{AP})) \ln(1 - \tau_h^*) \quad (18)$$

and for the average slope to be

$$m_{avg} = \beta_0(P(B_{AP})) - \beta_1(P(B_{AP})) \ln(1 - \tau_h^*). \quad (19)$$

α_0 , α_1 , β_0 , and β_1 are functions of $P(B_{AP})$ and are looked up from a table by the station as described in Section IV-B4. The first terms in Eqs. (18) and (19), i.e. α_0 and β_0 , are independent of τ_h^* , are absent in Eq. (15), and are discussed in detail in section IV-B3. From Eq. (15) and Fig. 5, α_1 and β_1 are the average value 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 τ_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 α_1 and β_1 are also dependent on B_{AP} . Furthermore, 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, $\frac{\partial L_B(l)}{\partial l} = 1/(\text{length of idle slot})$ for small l . In this case, it is also likely that τ_h^* is small, so $(1 - \tau_h^*)^{L(l)} = 1$ for small l . Therefore, for small $P(B_{AP})$, α_1 , which corresponds to small l , i.e. $l < l_{break}$, is given by $1/(\text{length of idle slot})$. 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 α_1 to decrease. Similarly, β_1 decreases with increasing $P(B_{AP})$ due to a similar phenomenon. These observations are verified via simulations shown in Table I of Section IV-B4.

B3. Coupling of transmission times

We now justify the existence of α_0 and β_0 in Eqs. (18) and (19). 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. 3 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 an excellent proxy for this. For lower-traffic networks, this effect is negligible, but for higher traffic networks with larger values of $P(B_{AP})$ this becomes more pronounced. Thus α_0 and β_0 should increase with $P(B_{AP})$ as verified via simulations shown in Table I of Section IV-B4.

B4. Estimating α_0 , α_1 , β_0 , and β_1

In general, computing closed-form analytical expressions for α_0 , α_1 , β_0 , and β_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 modified NS-2 described in Section III-D. For our simulation setup, we fix the locations of 7 APs with hexagonal cells and randomly place 40-50 nodes according to a spatial poisson process over the cells, associating each node with the closest AP. The nodes all send poisson application layer traffic at a fixed rate, which varies over different simulations. 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. 5. We then optimize m_1 and m_2 for each station to achieve the minimum squared distance between the 2-line approximation and the empirical curve. We generate many random network topologies in this manner with varying traffic load and choose a random subset of the stations for training. 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. (18) and (19) in order to determine the values of α_0 , α_1 , β_0 , and β_1 . In practice, nodes store a table of values for α_0 , α_1 , β_0 , and β_1 and look them up based on the observed $P(B_{AP})$. An example portion of this table is shown in Table I. As expected, α_1 and β_1 decrease with $P(B_{AP})$ while α_0 and β_0 increase with it. Since $\ln(1 - \tau_h^*)$ is on the order of 10^{-2} , all the terms in Eqs. (18) and (19) are of comparable size.

C. Simulation results

Fig. 6 shows the resulting average normalized error in P_{SC1} , defined as $|(P_{SC1}^{estimate} - P_{SC1}^{actual})/P_{SC1}^{actual}|$, for two bins 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 $P(B_{AP})$. Even though each scenario has different α 's and β 's, they have similar performance. 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. Beyond the knee in the 2-line model at $400\mu s$, errors are less than 20%, becoming less than 10% for longer packets.

TABLE I
SAMPLE VALUES OF α 'S AND β 'S

range of $P(B_{AP})$	α_0	α_1	β_0	β_1
.5-.6	0	5e-2	0	2.38e-2
.6-.7	0	5e-2	0	1.93e-2
.7-.75	0	5e-2	0	1.96e-2
.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

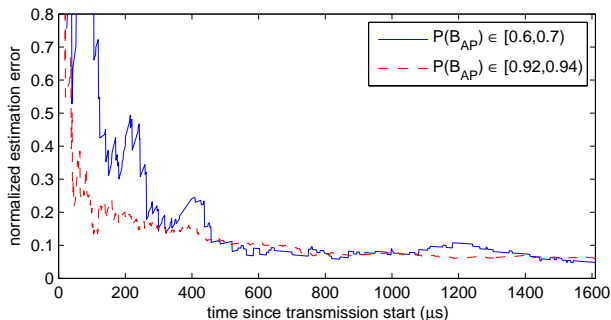


Fig. 6. Average absolute value of normalized estimation error, $|(P_{SC1}^{estimate} - P_{SC1}^{actual})/P_{SC1}^{actual}|$, for two traffic levels.

Fig. 7 shows a cumulative distribution of the estimation errors in P_{SC1} for maximum packet length over about 100 nodes from 30 different topologies. As seen, the error is below 20% about 80% of the time. Although there is a non-negligible chance for the error to be large, in practice this does not have a significant contribution to the total collision probability; this is because we have empirically found this to occur in scenarios where P_{SC1} accounts for an insignificant part of P_C .

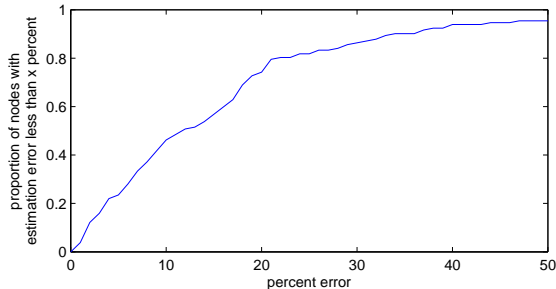


Fig. 7. Cumulative distributions of the estimation errors in P_{SC1} in percent for maximum packet length.

Having estimated the probabilities of each type of collision, we can combine them to find the total P_C using Eq. (1) as shown in Table II. Each row corresponds to a random node from a random scenario, with the average network traffic increasing from top to bottom. The first eight columns show the actual and estimated values of the probabilities, and the last column shows the total relative error: $(P_C^{est} - P_C^{act})/P_C^{act}$ in percent. Even though the P_{DC} estimate seems to be inaccurate for scenarios in which P_{SC2} is large, in practice, this is of little consequence since in these scenarios, a direct collision is a rare event and,

as such, does not significantly contribute to the total collision probability. As seen, the total estimate is always within 10% of the actual, and often much closer.

TABLE II
AGGREGATE P_C ESTIMATION RESULTS

P_{SC2}		P_{DC}		P_{SC1}		P_C		err
act	est	act	est	act	est	act	est	%
.229	.229	.021	.012	.273	.309	.452	.474	4.9
.200	.200	.011	.012	.155	.149	.332	.327	-1.4
.163	.163	.000	.012	.162	.133	.299	.283	-5.2
.343	.343	.000	.000	.356	.392	.577	.601	4.2
.171	.164	.026	.035	.082	.104	.258	.278	7.6
.661	.663	.108	.262	.921	.962	.976	.991	1.5
.902	.873	.035	.255	.860	.931	.987	.993	0.7
.883	.883	.063	.201	.943	.780	.994	.979	-1.4
.732	.732	.140	.265	.903	.874	.978	.975	-0.2

V. CONCLUSIONS AND FUTURE WORK

We have proposed a method to estimate the probabilities of three types of collision based on local and AP busy-idle information only. We have used NS-2 simulations to verify the accuracy of our proposed method. Unlike existing approaches in the literature, our approach does not assume individual nodes' traffic to be saturated; nor does it assume any prior knowledge about the network topology. Future work involves verifications of our approach using an actual implementation using an open source wireless driver.

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] Yi Li, Lili Qiu, Yin Zhang, Ratul Mahajan, and Eric Rozner, "Predictable Performance Optimization for Wireless Networks", in *Proc. of ACM SIGCOMM 2008*, Seattle, Washington, August 2008.
- [7] Cunqing Hua and Rong Zheng, "Starvation Modeling and Identification in Dense 802.11 Wireless Community Networks", in *Proc. of IEEE INFOCOM 2008*, Phoenix, Arizona, April 2008, pp.1022-1030.
- [8] Yong Yang, Jennifer C. Hou, and Lu-Chuan Kung, "Modeling the Effect of Transmit Power and Physical Carrier Sense in Multi-hop Wireless Networks", in *Proc. of IEEE INFOCOM Miniconference 2007*, Anchorage, Alaska, May 2007.
- [9] Xue Yang and Nitin Vaidya, "Spatial Backoff Contention Resolution for Wireless Networks", in *Proc. of IEEE WiMesh 2006*, Reston, Virginia, September 2006, pp.13-22.
- [10] Giuseppe Bianchi, "Performance Analysis of the 802.11 Distributed Coordination Function", *IEEE JSAC*, Vol. 18, No. 3, March 2000, pp.535-547.