Evaluation of the Masked Node Problem in Ad-Hoc Wireless LANs

Saikat Ray, Jeffrey B. Carruthers, and David Starobinski

Department of Electrical and Computer Engineering, Boston University

Abstract

IEEE 802.11 wireless networks employ the so-called *RTS*/*CTS* mechanism in order to avoid *DATA* packet collisions. The main design assumption is that all the nodes in the vicinity of a sender and a receiver will hear the *RTS* or *CTS* packets, and defer their transmission appropriately. This assumption happens not to hold in general, even under perfect operating conditions. Often, neighboring nodes are "masked" by other on-going transmissions nearby and, hence, are unable to receive the *RTS* or *CTS* packets correctly. We refer to such nodes as masked nodes. In this paper, we describe the masked node problem and show scenarios leading to *DATA* packet collisions. We evaluate the impact of masked nodes through mathematical analysis and real experiments on a small IEEE 802.11 ad-hoc network. The analytical and experimental data closely match and reveal that the presence of a masked node in a network can result in an order of magnitude increase in *DATA* packet loss compared to a network without masked nodes. These results are further validated by extensive simulations on a large-scale network, which show that masked nodes also significantly affect delay and throughput performance. Therefore, masked nodes severely limit the effectiveness of the *RTS*/*CTS* mechanism in preventing performance degradation in wireless LANs.

Index Terms

Wireless communication, experimental design, performance evaluation, queuing theory, simulation.

This work was supported in part by the National Science Foundation under CAREER grant ANI-0132802 and grant ANI-0240333 and by a SPRInG award from Boston University.

I. INTRODUCTION

IEEE 802.11 wireless local area networks (LANs) are rapidly gaining widespread acceptance [1, 2]. Due to their low cost of installation and maintenance, wireless LANs have become very popular in home, campus, and business environments. Furthermore, in settings such as construction sites and disaster-torn areas, wireless networks are often the only means for providing network connectivity.

The performance of a wireless local area network (WLAN) heavily depends on its medium access control scheme [3]. The IEEE 802.11 WLAN protocol uses a medium access control mechanism based on the "Carrier Sense Multiple Access" (CSMA) protocol [4]. In CSMA, a node is allowed to transmit only if it determines the medium to be idle. However, CSMA cannot prevent packet collisions caused by nodes that are located within the transmission range of the receiver, but not of the sender. Such nodes are called *hidden nodes* [5]. To prevent *DATA* packet collisions due to hidden nodes, IEEE 802.11 supports the *RTS*/*CTS* mechanism [6–8]. In this protocol, a pair of small control packets, called *RTS* and *CTS*, are transmitted initially to avoid costly *DATA* packet collisions.

The *RTS*/*CTS* mechanism can prevent *DATA* packet collisions when every node in the vicinity of the sender and the receiver hears at least one control packet and defers transmission appropriately. In ad hoc networks, however, this assumption does not hold in general. Neighboring nodes are often unable to receive the control packets because they are masked by on-going transmissions from other nodes near them. This means that the *RTS*/*CTS* mechanism does not generally prevent *DATA* packet collisions, even under perfect operating conditions, such as negligible propagation delay, no channel fading and no node mobility. Note that the masking effect, whereby a node may be unable to hear its neighbor's transmissions, does not necessarily impact wireless LANs

that use other mechanisms to avoid the hidden node problem.

In this work, we refer to a node that is supposed to receive an *RTS* or a *CTS* packet, but cannot interpret it correctly because of another on-going transmission, as a *masked node*. A masked node can subsequently cause *DATA* packet collisions, even if the *RTS*/*CTS* handshake is performed successfully between a sender and a receiver. Since *DATA* packet collisions reduce throughput and increase delay, masked nodes may significantly affect network performance. Thus, understanding the impact of masked nodes is essential to evaluate the performance of IEEE 802.11 wireless LANs that use the *RTS*/*CTS* mechanism.

Masked nodes are as fundamental as hidden nodes. Although the hidden node problem has been studied extensively in the literature, the masked node problem has received very little attention. The problem was briefly alluded to in [1, 7]. The present paper contributes to the understanding of masked nodes in several ways. First, we describe the masked node problem and show examples of sequences of events that lead to *DATA* packet collisions. Second, we assess the existence and impact of masked nodes on a small ad-hoc network consisting of four WLAN-equipped laptops, by conducting real experiments backed up by a queueing-theoretic analysis. The analytical and experimental data are in excellent agreement. They provide evidence that the presence of a masked node in a network can result in *DATA* packet loss higher than 10%, which represents an order of magnitude increase over the packet loss measured in a network without masked nodes. Third, we quantify the effect of masked nodes on IEEE 802.11 network performance through detailed simulations. The simulations allow us to consider larger networks, and estimate important performance metrics such as delay and throughput. In particular, the simulations show that masked nodes can double the average packet delay (because of retransmissions), rendering, in some situations, a wireless network unsuitable for multimedia traffic. These results show that

masked nodes significantly affect the overall performance of wireless LANs.

Outline of rest of the paper. Section II provides background on the IEEE 802.11 protocol and discusses related work. In Section III, we elaborate on the concept of masked nodes. We consider a linear topology network and show scenarios that lead to the formation of masked nodes and subsequent *DATA* packet collisions. We then conduct an analysis to derive the probability of a packet collision as a function of the traffic load in this network. In Section IV, we describe the design of our testbed and provide experimental results showing the significant impact of a masked node on a real wireless network. We compare these experimental results with those predicted by our analysis. In Section V, we present our simulation results and quantify the effect of masked nodes by contrasting the results against a hypothetical simulation mode where the formation of masked nodes is eliminated. We conclude the paper in Section VI.

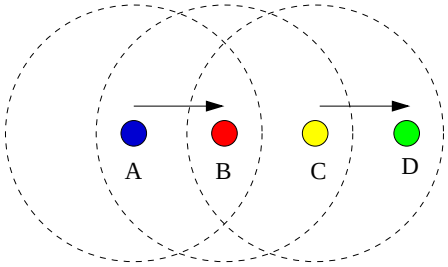
II. Background

A. Overview of the IEEE 802.11 MAC Protocol

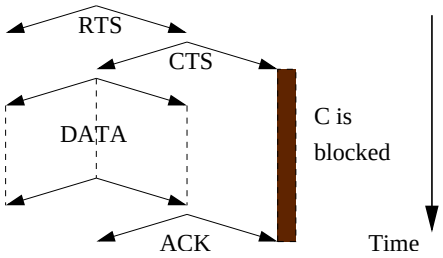
In this section, we briefly describe some of the salient features of the IEEE 802.11 Wireless LAN protocol that are most relevant to the rest of this paper. The protocol is described in detail in [2].

The IEEE 802.11 MAC protocol supports two types of access mode: *Point Coordination Function* (PCF) and *Distributed Coordination Function* (DCF). The DCF mode is more commonly used. In this mode, a node may transmit a packet using either the *basic access* method or the *RTS*/*CTS* method.

The basic access method is essentially equivalent to CSMA. A node transmits a *DATA* packet if it senses the channel to be idle. The receiver, upon receiving an error-free packet, returns an *Acknowledgment* (*ACK*) packet. If the transmitting node does not get the *ACK* back, it enters into



(a) The physical configuration.



(b) *RTS*/*CTS* mechanism in IEEE 802.11 MAC.

Fig. 1. IEEE 802.11 MAC. 1(a) shows the physical configuration of the nodes and 1(b) depicts the time-line. The dark bar below node C indicates its NAV. The hidden node problem is solved in this scenario by prohibiting node C from transmitting during node A's transmission.

backoff and retransmits after the backoff period. The basic access method suffers from the well-known *hidden node* problem [5]. As an example, consider the topology shown in Figure 1(a). In this topology, node C does not hear packet transmissions from node A. Thus, node C may transmit a packet to node D, while node A transmits a packet to node B. These simultaneous transmissions lead to a collision at node B, destroying the packet sent by node A, since node C's transmissions propagate in all directions[1]. In this scenario, node C is referred to as an *hidden* node with respect to node A.

In order to address this issue, the IEEE 802.11 MAC protocol also supports an *RTS*/*CTS* access control method. Figure 1(b) illustrates this scheme. When node A wants to send a packet to node B, it initially sends a small packet called *Request-to-Send* (RTS). Upon correctly receiving

[1]We do not consider directional antennas in this paper.

the *RTS*, node B responds with another small packet called *Clear-to-Send* (*CTS*). After receiving the *CTS*, node A sends the *DATA* packet to node B. If node B receives the *DATA* packet correctly, it sends an *ACK* back to node A. Any node that hears an *RTS* or a *CTS* is prohibited from transmitting any signal for a period that is encoded in the *duration* field of the *RTS* and *CTS*. The duration fields in *RTS* and *CTS* are set such that nodes A and B will be able to complete their communication within the prohibited period. The deferral periods are managed by a data structure called the Network Allocation Vector (NAV). The *RTS*/*CTS* mechanism solves the hidden node problem shown in Figure 1(a), since node C is notified by a *CTS* when node A initiates a transmission.

B. Related Work

Several works have previously shown that the *RTS*/*CTS* mechanism may fail due to various non-ideal operating conditions. In [6], the author points out that since the interference range may be larger than the communication range in open areas, *CTS* packets may not be received by some nodes that can subsequently interfere. The work in [9] evaluates this effect and in [10], the authors consider the effectiveness of power-control protocols. In [11], the authors show that if two nodes simultaneously transmit an *RTS* and a *CTS* packet, then the *CTS* packet will not be received and a subsequent data packet collision is likely. The probability of such an event becomes non-negligible if the propagation delay is significant. A similar situation (*RTS*/*CTS* collision) is also reported in [12]. In [13], the author mentions that if nodes are mobile, then a node that did not hear an *RTS* or *CTS* may migrate into the footprint of a receiver and destroy a *DATA* packet with its own transmission. The probability of such a scenario increases with the mobility of the nodes.

Our contribution substantially differs from these previous works in at least two major aspects:

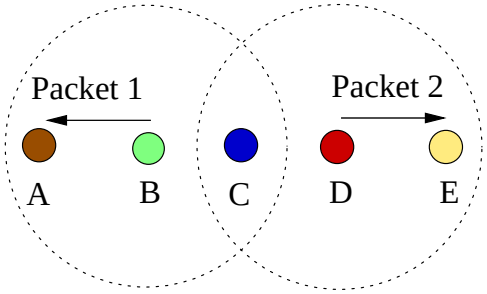


Fig. 2. The masked node. Node B transmits `Packet 1` to node A and node D transmits `Packet 2` to node E at the same time. Since node C receives signals from two different sources, it cannot decode either of the packets. Node C is said to be a masked node because each transmission masks the other.

1) We show that the *RTS*/*CTS* mechanism fails to avoid a significant number of packet collisions *even under perfect operating conditions*, due to the masked node problem.
2) In addition of performing detailed mathematical analysis and simulations, we validate our findings by conducting carefully designed experiments on a *real* IEEE 802.11 ad-hoc network.

III. MASKED NODES

The main idea behind using an *RTS*/*CTS* handshake is that nodes within the transmission range of either a sender or a receiver will hear at least one of the control packets and hence defer their transmission. However, it is incorrect to assume that *all* nodes within the sender's transmission range can hear the *RTS* and *all* nodes within the receiver's transmission range can hear the *CTS*, even under perfect operating conditions. The fact that node A is within the transmission range of node B does not necessarily guarantee that node A will be able to decode every packet originating from node B, due to other on-going transmissions near node A[2]. More specifically, if a node receives two or more signals simultaneously, then it generally cannot decode any of the corresponding packets, even if it wishes to. We refer to such a node as a *masked node*.

Figure 2 illustrates this problem. In this figure, node B transmits a packet to node A. Shortly

[2]By decode we mean that all the bits of the packets are interpreted without any error.

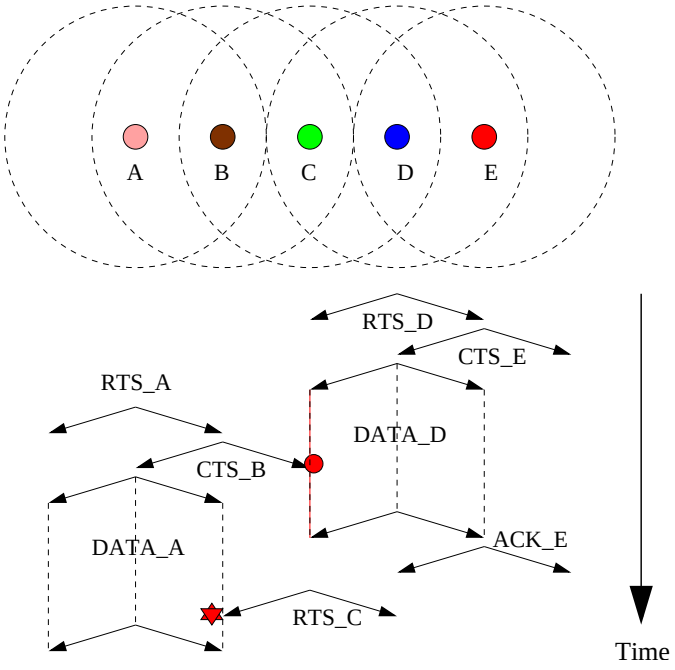


Fig. 3. A *DATA* packet collision due to a masked node. The circle symbol indicates the unheard *CTS* packet from node B, and the star symbol indicates the resulting packet collision.

thereafter, node D starts transmitting a packet to node E. Since node C receives signals from two different sources at the same time, it cannot decode either of the packets. Node C is unable to hear the transmission of node D to node E because it is masked by the on-going transmission from node B to node A (and vice-versa). In such a situation, node C is a masked node.

Masked nodes are as fundamental as hidden nodes. Similar to hidden nodes, masked nodes do not necessarily cause *DATA* packet collisions. However, if one of the packets a masked node was supposed to receive is a *CTS* or an *RTS*, then the masked node may subsequently cause a collision. We next present a typical scenario where the presence of a masked node leads to a *DATA* packet collision. We subsequently derive an analytical expression for the probability of such a collision. The analysis shows that the collision probability is quite high, even under moderate traffic load. The results of our analysis are corroborated by experimental results, presented in Section IV.

A. DATA *packet collision caused by a masked node*

The scenario is illustrated in Figure 3. Assume that initially all the nodes are idle and none of them is prohibited from transmitting. Now, node D and node E exchange an *RTS*/*CTS* dialog successfully and node D starts sending a *DATA* packet to node E. Node C receives the *RTS* sent by node D and updates its NAV appropriately. After node D starts transmitting the *DATA* packet, node A sends an *RTS* to node B. Since node B is not within node D's transmission range, it does not sense any carrier and responds with a *CTS*. This *CTS* should reach node C. However, node C is masked by the signal from node D. Thus, node C cannot decode the *CTS* packet. Node A, on the other hand, does receive the *CTS* and, thus, starts sending its *DATA* packet. In the mean time, nodes D and E complete their communication and node C becomes free to transmit. Node C now transmits an *RTS* destined for one of its neighbors. This *RTS* reaches node B and destroys the data packet node B is receiving.

Another possibility is that after nodes D and E complete their communication, an RTS is sent by node D, or by another node, to node C. Since node C is free to transmit as per its NAV, it responds with a *CTS* which collides with the *DATA* packet that node B is currently receiving.

As shown by our analysis and experiments, the likelihood of this scenario is relatively high because node C remains masked during an entire *DATA* packet transmission, which represents a substantial amount of time.

B. Analysis

In this section, we evaluate the impact of the masked node on the performance of the network shown in Figure 3. Our goal is to derive the probability of a packet collision as a function of the traffic load in the network. In order to introduce our analysis technique, we start with the simpler case of the hidden node depicted in Figure 1(a). The same network configurations will

be used later for our experiments.

1) Hidden Node: We consider the linear network shown in Figure 1(a). In this configuration, node A sends packets to node B and node C sends packets to node D.

We now introduce our notation and assumptions for the analysis. Both node A and C maintain infinite buffer queues. The exogenous arrivals to these queues are independent Poisson processes, each with rate λ (number of packets per unit of time). The *DATA* packet size is fixed and the transmission time of each packet is T. The transmission time of *ACK* packets is negligible. We define the load on each queue to be $\rho = \lambda T$, with $\rho < 1$. We assume that the system is started at time $t = -\infty$, so that it reaches its steady-state at time $t = 0$. The channel is noise-free, so that packets are lost only because of collisions.

In the configuration considered, node C's packets are all correctly received at node D and, thus, never retransmitted. Therefore, based on our assumptions, the statistical behavior of node C corresponds to an $M/D/1$ queue with service time T. We denote the steady-state number of packets in this queue (including the packet in service) by the random variable q_C. The distribution of this random variable is well-known (see [14], page 220).

For simplicity, we assume that the state of node C's queue at the time when node A starts a packet transmission is the same as at any random points of time. In other words, node A's packets see the system in its time average.

Without any loss of generality, let the time at which node A starts a packet transmission be $t = 0$ (a packet transmission can either be a first attempt or a retransmission). Our goal is to compute the probability that this packet collides at the receiver, i.e., node B. For this purpose, we condition on the state of node C at time $t = 0$. We distinguish between the cases where

node C's queue is empty, i.e., $q_C = 0$, and non-empty, i.e., $q_C > 0$. We thus have

$$\Pr\{\text{Collision}\} = \Pr\{\text{Collision}|q_C > 0\} \cdot \Pr\{q_C > 0\} + \Pr\{\text{Collision}|q_C = 0\} \cdot \Pr\{q_C = 0\}. \quad (1)$$

We now compute the conditional probabilities. If $q_C > 0$, then node C is currently transmitting. Therefore, node A's packet collides with probability 1, i.e.,

$$\Pr\{\text{Collision}|q_C > 0\} = 1. \quad (2)$$

Now suppose $q_C = 0$ at time $t = 0$. A collision will happen only if a packet arrives at node C before $t = T$ (note that this packet is immediately transmitted). Since the arrival process to node C is Poisson with rate λ, we obtain

$$\Pr\{\text{Collision}|q_C = 0\} = 1 - e^{-\lambda T} = 1 - e^{-\rho}. \quad (3)$$

Since $\Pr\{q_C = 0\} = 1 - \rho$, we obtain the following expression for the collision probability in the hidden node case from Eqs. (1), (2), and (3):

$$\Pr\{\text{Collision}\} = 1 \cdot \rho + (1 - e^{-\rho}) \cdot (1 - \rho) = 1 - e^{-\rho}(1 - \rho) \quad (4)$$

As shown by our experiments later, this simple analysis somewhat underestimates the collision probability in a real system (although it does capture the right order). The main reason is as follows. When a packet sent by node A collides, it is retransmitted after a short back-off time period[3]. The chance that this retransmission collides is higher than other packets since node C may not have yet completed its transmission. Notice that the same problem does not arise in the masked node case. Nodes A and B cannot perform a successful *RTS*/*CTS* handshake as long as node C is transmitting, since node B is unable to hear any *RTS* sent by node A during this time. Thus, node A will not start retransmitting a *DATA* packet before node C completes its transmission.

[3]Note that back-off time is usually negligible in comparison to transmission time.

2) The Masked Node Scenario: We consider the network configuration of Figure 3. In this configuration, node A sends packets to node B, node C to node D, and node D to node E. All the pairs communicate by first initiating an *RTS*/*CTS* handshake, which we assume to be instantaneous. The arrival process of new packets at each sending nodes is an independent Poisson process with mean rate λ. We note that this network configuration will occur repeatedly at different times and locations in any large multi-hop network. Thus, we consider this network and the traffic patterns to be typical local snapshots of larger network behavior. This viewpoint will be validated when we examine simulation results for a large wireless network in Section V.

The analysis of the masked node in this network is similar to that of the hidden node carried out in the previous section. The strategy is first to calculate the collision probabilities conditioned on the states of nodes C and D and then combine these conditional probabilities by approximating the steady-state joint probabilities of the states of node C and D by two independent $M/D/1$ queues. This approximated analysis becomes asymptotically exact at low load, that is, as $\rho \to 0$. In practice, the analysis remains fairly accurate over a wide range of values of ρ, as shown by our experiments in the next section.

We first compute the conditional probabilities. Suppose that node A starts transmitting a *DATA* packet at time $t = 0$, following a successful *RTS*/*CTS* handshake. Let the queue lengths in nodes C and D at $t = 0$ be denoted by the random variables q_C and q_D, respectively. We next compute the probability that this *DATA* packet collides conditioned on the following possible states of the queues of nodes C and D at this time: (a) $q_C = 0, q_D = 1$, (b) $q_C = 0, q_D \geq 2$, (c) $q_C \geq 1, q_D = 1$ and (d) $q_C \geq 1, q_D \geq 2$. These four cases are mutually exclusive and the probability of collision in other states is 0. Note that node C cannot transmit at time $t = 0$, otherwise the *RTS*/*CTS* handshake between nodes A and B would have been unsuccessful.

Moreover, node D must be transmitting at this time in order for node C to be masked and not hear the *CTS*.

We now consider the four cases:

Case (a): At time $t = 0$, $q_C = 0$ and $q_D = 1$. Suppose that node D completes its transmission at time τ, where $0 \leq \tau \leq T$. The packet sent by node A will be destroyed only if node C initiates a packet transmission during the time interval $[\tau, T]$. To compute the probability of this event, we need to distinguish between the following three sub-cases:

- Event A_a: There has been at least one new arrival to both of the queues of node C and node D during the interval $[0, \tau]$. Thus, at time $t = \tau$, both nodes C and D have a packet to transmit. In such a case, we assume that node C has a probability $1/2$ to win the channel contention (an assumption justifiable at low load). So,

$$\Pr\{\text{collision}, A_a | \tau, q_C = 0, q_D = 1\} = \frac{1}{2}\left(1 - e^{-\lambda\tau}\right)^2 \tag{5}$$

- Event B_a: One or more new packets have arrived to the queue of node C during the interval $[0, \tau]$, but none has joined node D. Therefore, only node C has a packet to send at time $t = \tau$. This packet collides with node A's packet with probability 1. Thus,

$$\Pr\{\text{collision}, B_a | \tau, q_C = 0, q_D = 1\} = \left(1 - e^{-\lambda\tau}\right) e^{-\lambda\tau} \tag{6}$$

- Event C_a: No packet joins any of the queues of nodes C and D during the interval $[0, \tau]$. Therefore, both queues are empty at time $t = \tau$. For a collision still to take place, a new packet needs to arrive to node C during the interval $[\tau, T]$ and precede an arrival to node D (if any) during this interval. The probability of this event is given by

$$\begin{aligned} \Pr\{\text{collision}, C_a | \tau, q_C = 0, q_D = 1\} &= e^{-2\lambda\tau} \int_{\tau_2=0}^{T-\tau} \int_{\tau_1=\tau_2}^{\infty} \lambda e^{-\lambda\tau_1} \lambda e^{-\lambda\tau_2} d\tau_1 d\tau_2 \\ &= \frac{1}{2}\left[e^{-2\lambda\tau} - e^{-2\lambda T}\right] \end{aligned} \tag{7}$$

The random variable τ is uniformly distributed over the interval $[0, T]$, based on the assumption that the state of node D at time $t = 0$ is the same as at any random points of time (an assumption again valid at low load). Combining the expressions for the sub-cases A_a, B_a and C_a and integrating over τ, we obtain the final expression for the conditional collision probability in case (a):

$$\Pr\{\text{collision}|q_C = 0, q_D = 1\} = \frac{1}{2}\left(1 - e^{-2\rho}\right) \tag{8}$$

Case (b): At time $t = 0$, $q_C = 0$ and $q_D \geq 2$. Node D completes its transmission at time $t = \tau$. For a collision to occur, at least one new packet needs to arrive to node C during the interval $[0, \tau]$ and node C must subsequently win the channel contention (probability 1/2). The final expression for the conditional collision probability in this case is given by

$$\Pr\{\text{collision}|q_C = 0, q_D \geq 2\} = \frac{1}{2} - \frac{1}{2\rho} + \frac{e^{-\rho}}{2\rho} \tag{9}$$

Case (c): At time $t = 0$, $q_C \geq 1$ and $q_D = 1$. Node D completes its transmission at time $t = \tau$. We now need to consider two sub-cases:

- Event A_c: If there are no arrivals to node D during the interval $[0, \tau]$, then node C will transmit a packet at time τ which leads to a collision. Therefore,

$$\Pr\{\text{collision}, A_c|q_C \geq 1, q_D = 1\} = \frac{1}{\rho} - \frac{e^{-\rho}}{\rho} \tag{10}$$

- Event B_c: If there is at least one arrival to node D during the interval $[0, \tau]$, then both node C and D have a packet to transmit at time τ. Node C wins the contention with probability $1/2$. We then have

$$\Pr\{\text{collision}, B_c|q_C \geq 1, q_D = 1\} = \frac{1}{2} - \frac{1}{2\rho} + \frac{e^{-\rho}}{2\rho} \tag{11}$$

By adding the probabilities of these two mutually exclusive events, we get

$$\Pr\{\text{collision}|q_C \geq 1, q_D = 1\} = \frac{1}{2} + \frac{1}{2\rho} - \frac{e^{-\rho}}{2\rho} \tag{12}$$

Case (d): At time $t = 0$, $q_C \geq 1, q_D \geq 2$. When node D finishes its transmission at time $t = \tau$, both nodes C and D have a packet to transmit. With probability $1/2$, node C is the one that transmits, causing a collision. Thus,

$$\Pr\{\text{collision}|q_C \geq 1, q_D \geq 2\} = \frac{1}{2} \tag{13}$$

We next compute the joint queue length probabilities of nodes C and D. As mentioned earlier, we assume a low load regime in which the queues of these two nodes behave as two independent $M/D/1$ queues. The load on these queues is ρ_C and ρ_D, respectively. Thus, the probabilities of the conditioning states can be found by using the standard $M/D/1$ formula from [14]. Combining all the conditioned cases, we obtain the following expression for the probability of a collision in the masked node scenario:

$$\begin{aligned}
\Pr\{\text{Collision}\} &= \frac{1}{2}\left[1 - e^{-2\rho}\right](1-\rho_C)(1-\rho_D)(e^{\rho_D} - 1) \\
&+ \left[\frac{1}{2} - \frac{1}{2\rho} + \frac{e^{-\rho}}{2\rho}\right](1-\rho_C)\left[1 - (1-\rho_D)\,e^{\rho_D}\right] \\
&+ \left[\frac{1}{2} + \frac{1}{2\rho} - \frac{e^{-\rho}}{2\rho}\right]\rho_C\,(1-\rho_D)\,(e^{\rho_D} - 1) \\
&+ \frac{1}{2}\,\rho_C\left[1 - (1-\rho_D)\,e^{\rho_D}\right]
\end{aligned} \tag{14}$$

The only remaining part to complete our derivation is to find expressions for ρ_C and ρ_D. For this purpose, we invoke Little's Law which states that $\rho_C = \lambda \bar{T}_C$ and $\rho_D = \lambda \bar{T}_D$, where $\bar{T}_C$ (resp., $\bar{T}_D$) represents the mean service time of a packet at the head of queue C (resp., D).

The computation of the mean service time is difficult in general, since the wireless channel is shared between multiple nodes. However, at very low load, the channel is free almost all the time and therefore $\bar{T}_C = \bar{T}_D = T$. Consequently, we obtain as a first-order approximation $\rho_C = \rho_D = \lambda T = \rho$. The collision probability is found by substituting these values into Eq. (14). We next derive a second-order approximation for ρ_C and ρ_D that helps in improving the accuracy

of the analysis at moderate load.

3) Second Order Approximation of the Queue Length Probabilities: In this section, we refine the analysis of the packet collision probability in the masked node scenario by deriving second-order approximations for ρ_C and ρ_D, the load on queues C and D respectively. We first remind that any analytic function $f(x)$ can be expanded about $x = 0$ in the form of a MacLaurin (Taylor) series

$$f(x) \quad = \quad a_0 + a_1 x + a_2 x^2 + \cdots = \sum_{i=0}^{\infty} a_i x^i. \tag{15}$$

The polynomial $f^{(n)}(x) = \sum_{i=0}^{n} a_i x^i$ is called the $n-$th order approximation of $f(x)$. We will also use the standard asymptotic notations $g(x) = o(x^n)$ if $\lim_{x\to 0} g(x)/x^n = 0$, and $g(x) = \Theta(x^n)$ if $K_1 < \lim_{x\to 0} g(x)/x^n < K_2$, where K_1 and K_2 are positive constants.

We denote by $\bar{T}_C^{(n)}$ and $\bar{T}_D^{(n)}$ the n-th order approximation about $\rho = 0$ of the mean service time at the queues of node C and D respectively. Similarly, $\rho_C^{(n)}$ and $\rho_D^{(n)}$ represent the n-th order approximation of the load (utilization) at each of these queues.

Our derivation is based on an iterative procedure [15]. We compute the n-th order approximation of the service times $\bar{T}^{(n)}$ based on the n-th order approximation of the queue utilization $\rho^{(n)}$. Then, using Little's Law, we derive $\rho^{(n+1)}$ from $\bar{T}^{(n)}$, and repeat the procedure.

As explained in Section III-B.2, when $\rho \to 0$ all the queues in the system behave as independent queues with deterministic service time T. Thus, we have $\bar{T}_C^{(0)} = \bar{T}_D^{(0)} = T$. From Little's Law, we then obtain $\rho_C^{(1)} = \rho_D^{(1)} = \rho$.

Our next step is to compute a first-order approximation for the mean service time at queues C and D. We note that we need only to take into consideration events that occur with probability $\Theta(1)$ or $\Theta(\rho)$. Consider the system that consists of the queues of nodes A, C and D (node B does not have *DATA* packets to transmit). From our first-order approximation of the queue utilization,

the steady-state probabilities of each queue correspond to those of an independent $M/D/1$ queue with load ρ. Thus, the probability that, at a random point of time, there is more than one packet in the system is $o(\rho)$. At the instant of a new packet arrival to nodes C or D we therefore have to account only for the four following events[4]: H_1: All queues are empty; H_2: Queue A contains one packet under transmission; H_3: Queue C contains one packet under transmission; H_4: Queue D contains one packet under transmission.

We can then compute the first order approximation of the mean service time by conditioning on these events, i.e.,

$$\bar{T}_C^{(1)} = E[T_C^{(1)}|H_1]\Pr\{H_1\} + E[T_C^{(1)}|H_2]\Pr\{H_2\} + E[T_C^{(1)}|H_3]\Pr\{H_3\} + E[T_C^{(1)}|H_4]\Pr\{H_4\}. \tag{16}$$

The first order approximation of the mean service time at queue D can be computed similarly.

Since the queues behave quasi-independently, the probabilities of the conditioning events can be expressed as follows:

$$\Pr\{H_1\} = 1 - 3\rho + o(\rho) \tag{17}$$

$$\Pr\{H_2\} = \Pr\{H_3\} = \Pr\{H_4\} = \rho + o(\rho). \tag{18}$$

We next compute the conditional expectations appearing in Eq. (16). We recall that the service time extends from the point when a packet reaches the head of queue until the packet's transmission is completed. If the system is empty when a new packet arrives to node C (event H_1), then the service time is simply T. Therefore, $E[T_C^{(1)}|H_1] = T$. Now consider the case where node A is transmitting when node C's packet arrives (event H_2). In that case, node C's packet needs first to wait for the residual time of the packet transmission at node A, the average

[4]Note that due to the PASTA (Poisson Arrivals See Time-Average) property, packet arrivals see the system in the same state as at any random point of time.

of which is $T/2$. The probability of any packet arrival to the system during this residual time is $\Theta(\rho)$. This event can be ignored since $\Pr\{H_2\}$ is already itself on the order of $\Theta(\rho)$. Thus, once node A completes its transmission, node C will immediately transmit its packet, and we obtain $E[T_C^{(1)}|H_2] = 3T/2$. A similar argument holds when a packet arrives to node C while node D is transmitting (event H_4). Hence, $E[T_C^{(1)}|H_4] = 3T/2$. The last case is when a packet arrival to node C needs to wait in the queue of that node (event H_3). Since queueing is not part of the service time, we simply have $E[T_C^{(1)}|H_3] = T$. Note that the probability of any packet arrival to the system while the packet is queued at node C is again $\Theta(\rho)$.

The derivation of the conditional expectations for the mean service time at node D is very similar. The only subtle difference is for event H_2. In that case, we have $E[T_D^{(1)}|H_2] = T$, because nodes A and D do not interfere and can transmit simultaneously. The remaining expressions are as follows: $E[T_D^{(1)}|H_1] = E[T_D^{(1)}|H_4] = T$ and $E[T_D^{(1)}|H_3] = 3T/2$.

Substituting the above derived expressions for the conditional expectations and conditioning probabilities into Eq. (16), we get

$$\bar{T}_C^{(1)} = T(1+\rho); \qquad \bar{T}_D^{(1)} = T(1+\frac{\rho}{2}). \tag{19}$$

Finally, using Little's Law, we obtain the following second-order approximation for the queue load at nodes C and D:

$$\rho_C^{(2)} = \lambda \bar{T}_C^{(1)} = \rho + \rho^2 \tag{20}$$

$$\rho_D^{(2)} = \lambda \bar{T}_D^{(1)} = \rho + \frac{\rho^2}{2}. \tag{21}$$

A refined, second-order approximation for the packet collision probability is obtained by substituting $\rho_C^{(2)}$ and $\rho_D^{(2)}$ into Eq. (14). In Section IV, we show that this refined expression predicts very well the packet loss probability measured in real experiments.

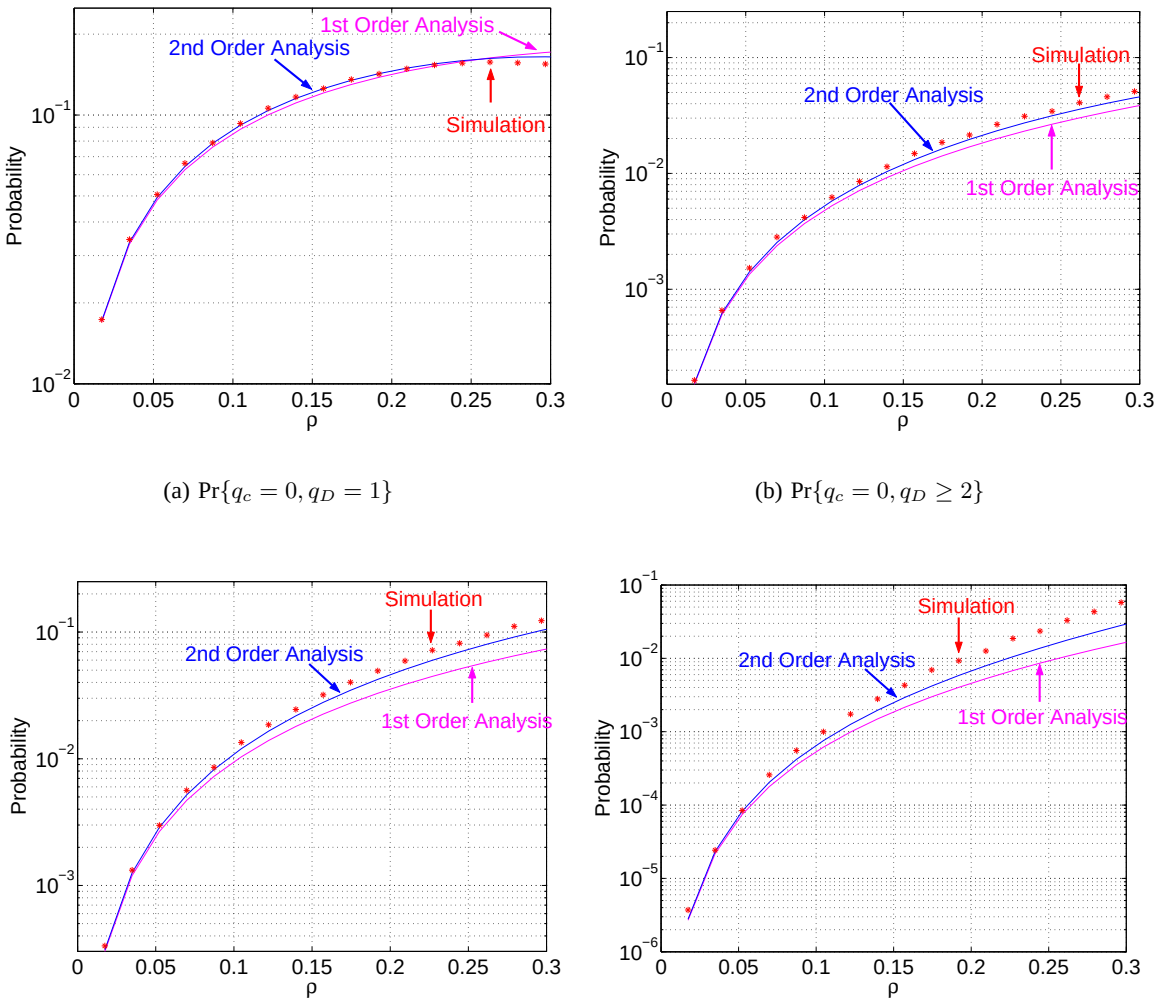


Fig. 4. Joint queue-length probabilities at nodes C and D: Comparison between analytical and simulation results as a function of the traffic load.

We further validate this second-order analysis through simulations. We remind that the analysis approximates the joint distribution of the queue length at node C and D by those of two independent $M/D/1$ queues (it is worth emphasizing that our approximated analysis does not assume that the service time is deterministic but rather that the queue-length distribution of an $M/D/1$ queue can closely approximate the actual queue-length distribution). Fig. 4 compares the analytical and simulation results obtained for the steady state probabilities of the four

conditioning states of queues C and D at time $t = 0$ (see Section III-B.2 and also note the different y-axis scale in each figure). From the figure, it is clear that both the first and the second-order approximations perform well as $\rho \to 0$, although the second-order approximation is more accurate at moderate values of ρ, as expected. The largest discrepancy is observed for case (d), i.e., for the state $\{q_c \geq 1, q_D \geq 2\}$. However, at low and moderate load, the probability that the system is in this state is relatively low compared to the other states.

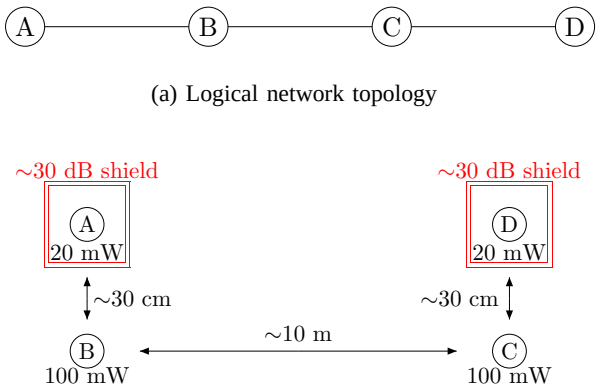
In the next section, the first-order and second-order expressions derived for the collision probability are compared with experimental results obtained on a real wireless network.

IV. Experimental Results

We designed and performed two suites of experiments on an actual wireless ad hoc network. In the first suite, we measure the packet collision probability in several different test cases to assess the existence and importance of the masked node problem in wireless LANs. In the second suite, we obtain experimental results for the packet collision probability as a function of the traffic load, under the same network configurations used in Section III-B for the analysis of the hidden and masked nodes.

A. Experiment design

The equipment consisted of four laptop computers running RedHat Linux 8.0. Each laptop was equipped with a Cisco Aironet 350 IEEE 802.11b PCMCIA card. The laptops, denoted by the letters A, B, C and D, were arranged according to the configuration shown in Figure 5. In this setup, each laptop could only receive packets or sense the carrier from immediate neighbors. This was done by shielding the wireless LAN cards on nodes A and D which ensures adequate power reception from neighbors but isolates non-neighbors. An alternate way to create the specified



(a) Logical network topology

(b) Physical configuration

Fig. 5. Experimental setup

topology is to place the nodes on separate floors of a building. This way, the nodes separated by a single floor can have strong connections while nodes separated by two or more floors are effectively isolated from one another.

In order to provide clear evidence of the masked node problem, we considered the following five scenarios:

a) **Point-to-Point**: Node A transmits to node B without *RTS*/*CTS*-protection; other nodes do not transmit.

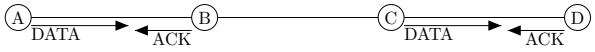
b) **Hidden node**: Node A transmits to node B without *RTS*/*CTS*-protection; node C (the hidden node) transmits to node D without *RTS*/*CTS*-protection; node D does not transmit.

c) **Hidden node with *RTS***: Node A transmits to node B with *RTS*/*CTS*-protection; node C transmits to node D with *RTS*/*CTS*-protection; node D does not transmit.

d) **Masked node**: Node A transmits to node B with *RTS*/*CTS*-protection; node C (the masked node) transmits to node D with *RTS*/*CTS*-protection; node D transmits broadcast packets.

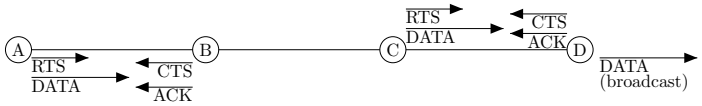
e) **Control**: Node A transmits to node B without *RTS*/*CTS*-protection; node C does not transmit; node D transmits broadcast packets.

(a) Point-to-point: A transmits to B without *RTS*/*CTS*.

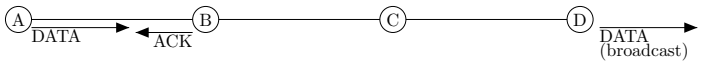


(b) Hidden node: A transmits to B and C transmits to D; neither use *RTS*/*CTS*.

(c) Hidden node with *RTS*: A transmits to B and C transmits to D; both use *RTS*/*CTS*.



(d) Masked node: A transmits to B and C transmits to D; both use *RTS*/*CTS*; D broadcasts.



(e) Control: A transmits to B without *RTS*/*CTS* and D broadcasts.

Fig. 6. The five test cases of our experiments.

The test cases are outlined graphically in Figure 6. In all cases, the reference traffic stream is from node A to node B. Case (a) is used to test the quality of this link without the use of an *RTS*/*CTS* exchange. Case (b) is the hidden node case without the protection of an *RTS*/*CTS* exchange, as analyzed in Section III-B. Since node C cannot effectively sense the transmission of node A, it is a hidden node. In case (c), we test the same configuration, but with *RTS*/*CTS* exchange which is supposed to solve the hidden node problem. Case (d) is the masked node case analyzed in Section III-B, where C represents the masked node. Finally, case (e) is a control experiment, which ensures that the experimental results truly represent the impact of masked nodes. We note that having node D broadcast packets is essentially equivalent to having node D

exchange packets with a fifth node E, as long as the channel error rate between the two nodes is low and the size of control packets is negligible compared to that of *DATA* packets. Both conditions hold in our analysis and experiments.

The link performance between node A and node B was determined by measuring the number of data packets transmitted for which no acknowledgment was received (N_{noACK}) and measuring the total number of data packets transmitted (N_{all}). The ratio N_{noACK}/N_{all} is called the *fraction of unreplied ACK*, or more simply just the packet error rate. This quantity corresponds to the *DATA* packet collision probability, if no *ACK* packet is lost (which is typically the case, since *ACK* packets are very short). Note that when an *RTS*/*CTS* exchange is not successful, this event is not captured since no data packet is generated in that case. The measurements were derived from those reported by the card in its statistics resource identifier (RID) as Tal.NoAck and Tal.RxAck (fields 12 and 14) as follows: $N_{noACK} =$ Tal.NoAck and $N_{all} =$ Tal.NoAck + Tal.RxAck. [16]

In all test cases, *DATA* packets are generated following an homogeneous Poisson arrival process. The size of each packet is fixed. The packet generation is performed by a Java program which creates UDP packets for transmission by the card. Therefore, the only retransmissions are those attempted by the cards at the MAC level: there are no transport-level acknowledgments or retransmissions. The relevant configuration parameters are presented in Table I(a). Channel 6 was chosen as it provided the channel with the least interference. The beacon interval was increased in order to minimize collision events between the Poisson traffic flows and beacons.

B. Results

We performed the first suite of tests with a packet size of 1500 bytes and a 20 packet per second average transmission rate for all data streams. This represents a net data rate of 240 kb/s between transmitting nodes, not including framing overhead and control packets. Hence, when

both node A and node C are transmitting, the net load at node B is about 48% of the nominal data rate of the receiver. Each test case was evaluated in ten separate runs, and each test run was stopped when 3000 acknowledged packets were transmitted from node A to node B.

The results are summarized in Figure 7. The baseline quality of the link between node A and node B is established by the point-to-point tests shown in Figure 6(a). For this case, the measured packet error rate ranged from 0.48% to 0.93% with a mean of 0.73%. Thus, the quality of the link is excellent and few packets are lost due to bit errors.

In the hidden node scenario (Figure 6(b)), node A is transmitting a traffic stream to node B and node C is interfering with this data since it is transmitting to node D. Neither uses an *RTS*/*CTS* exchange to protect its transmissions, and neither node can sense the transmissions of the other. In such a scenario, the measured packet error rates becomes very high, ranging from 40.0% to 45.4% with a mean of 43.4%

Hidden node mitigation is tested in the scenario shown in Figure 6(c). Both nodes A and C now protect their transmissions by sending an *RTS* and waiting for a *CTS* response. We obtain an error rate of about 0.78%, very similar to the range observed in the point-to-point case. This clearly demonstrates the effectiveness of the *RTS*/*CTS* exchange in eliminating collisions and hence packet errors.

The masked node case is tested as shown in Figure 6(d). Nodes A and C are transmitting to nodes B and D respectively and are protecting their transmissions using an *RTS*/*CTS* exchange. However, node D is now transmitting broadcast packets. Therefore, during a packet transmission from node D, node C is prevented from hearing any traffic from node B. Thus, node C is a masked node and will create packet collisions as previously discussed. The result is shown in Figure 7; the average packet error rate for this masked node case is 13.0%, with error rates

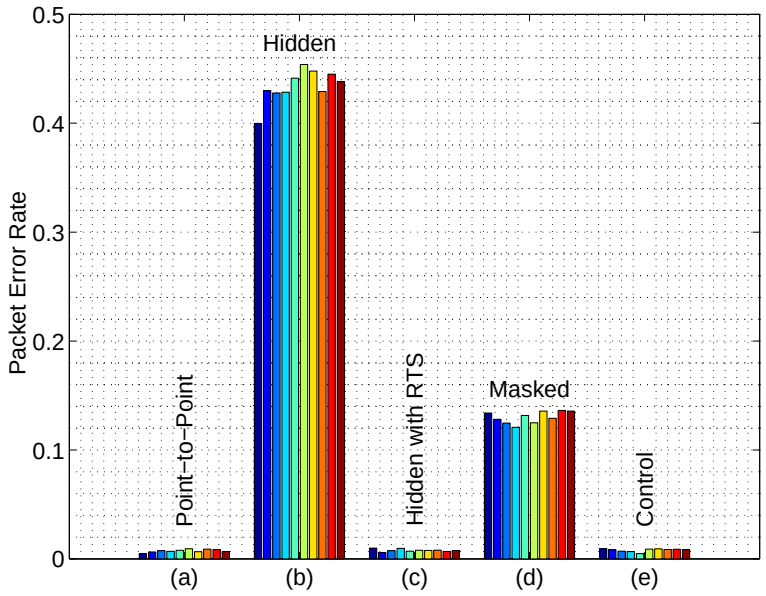


Fig. 7. Experimental results for the packet error rate in each test case. Each case is evaluated using ten separate runs.

measured in individual runs ranging from 12.1% to 13.6%. This is more than an order of magnitude increase over the packet error rate for the "protected" hidden node case. This clearly demonstrates that *RTS*/*CTS* exchanges fail to eliminate a significant amount of packet collisions, and that the failure in this case is due to the presence of masked nodes.

The control experiment, shown in Figure 6(e), is included to verify that the marked increase in the packet error rate is due to transmissions by the masked node C, and not due to transmissions by the broadcasting node D. This is indeed the case since the measured error rate (mean 0.80%) in this experiment is almost identical to the rate in the point-to-point experiment.

In the second suite of experiments, we focus on the effect of the traffic load on the packet error rate in the hidden node scenario (Figure 6(b)) and the masked node scenario (Figure 6(d)). We recall that the load ρ is defined as the product of the arrival rate with the service time. We increase the load by correspondingly increasing the packet arrival rate of the Poisson process (the packet size is kept fixed). Each test case is evaluated in three separate runs for each load value,

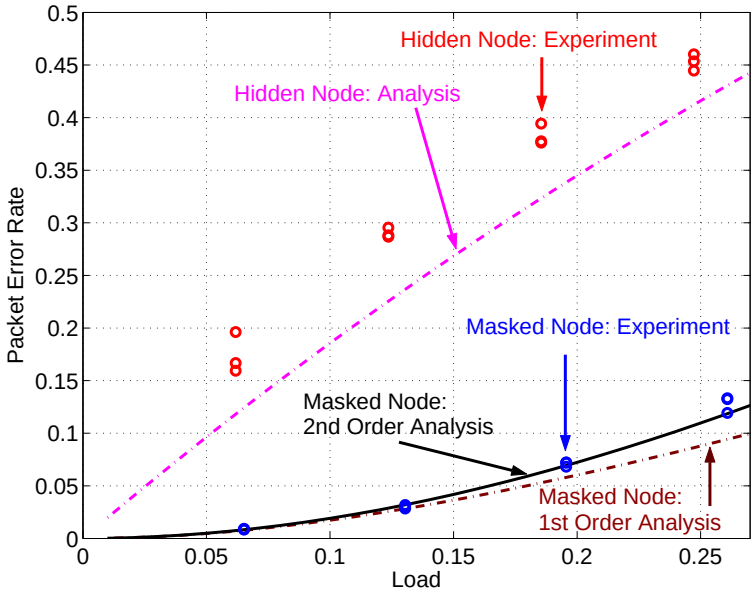


Fig. 8. Packet error rate versus load for the hidden node case and masked node case. Each case is evaluated using three separate runs.

and each test run is stopped when 3000 acknowledged packets are transmitted from node A to node B.

Figure 8 depicts the experimental results together with the analytical values derived in Section III-B and III-B.3. The expression for the packet error rate in the hidden node case is given by Eq. (4). For the masked node case, the expression for the packet error rate is given by Eq. (14). For the first-order analysis, we use $\rho_C = \rho_D = \rho$. For the second-order analysis, we use $\rho_C = \rho + \rho^2$ and $\rho_D = \rho + \rho^2/2$ (see Eqs. (20) and (21)).

We observe that the measured packet error rates in the hidden node case is significant, even at low load. The *RTS*/*CTS* mechanism helps in mitigating the problem. However, the results for the masked node case show that this mechanism is still insufficient in preventing a substantial amount of packet loss.

As expected, our simple analysis of the hidden node somewhat underestimates the actual value of the packet collision probability, although it captures the right order (see explanation in

Section III-B). On the other hand, our analysis of the masked node agrees very well with the measured data. In particular, the second-order analysis exhibits remarkable accuracy over the entire range of traffic load values used in our experiments.

V. Large Network Simulation

In the previous sections, we have presented both analytical and experimental evidences that masked nodes can lead to significant packet loss. An extension of our analysis to networks of arbitrary topology does not appear immediate. Therefore, we use simulation to quantify the impact of masked nodes on a large IEEE 802.11 ad hoc wireless network. Furthermore, the simulations enable us to evaluate the impact of masked nodes on other important performance metrics, such as throughput and delay.

A. Simulation Models

We have developed a discrete event simulator in MATLAB [17]. The simulator simulates a two-dimensional network. We assume that every node transmits with the same power in the same channel, all transmissions experience the same path loss vs. distance profile, and each node has the same antenna gain and receiver sensitivity. Thus, the range of each node is the same (called the *footprint*) and chosen to be 5 units of distance. If, for instance, 1 unit of distance corresponds to 100 m, then the range of each node is 500 m. A receiver decodes a packet if and only if the packet does not overlap with any other packet transmitted by a node within its range. The propagation delay is assumed to be negligible. Thus, every node within a sender's footprint senses a busy channel immediately after the transmission begins. Nodes are also static. These simulation settings allow us to separate the effects of masked nodes from other causes of packet collisions, such as channel fading, propagation delay, and mobility.

Mode	Ad hoc
Channel	6
Data rate	1 Mb/s only
Power mode	Constantly awake
Beacon interval	993.3 ms
DATA size	1500 Byte
Short retry limit	16
Long retry limit	16
RTS Threshold	10 or 2300
Fragmentation Threshold	2312
Transmit power	See Fig. 5
Scan mode	Active
Probe delay	3.036 ms
Probe Energy timeout	3.036 ms
Card Type	Cisco Aironet 350
Firmware Version	4.25.23
Operating System	Redhat Linux™ 8.0

(a) Key experiment parameters

Data rate	1 Mbps
Preamble length	144 μs
PLCP Header	48 μs
RTS size	20 Byte
CTS size	14 Byte
DATA size	2300 Byte
ACK size	14 Byte
SIFS	10 μs
DIFS	50 μs
Backoff Slot_time	20 μs
turnaround_time	4 μs
`CW_min`	31
`CW_max`	1023
`LongRetryLimit`	7
Radio Range	5 unit
Node Density	10 nodes per footprint
Network Size	900 unit^2

(b) Key simulation parameters.

TABLE I

PARAMETER TABLES FOR EXPERIMENT AND SIMULATION

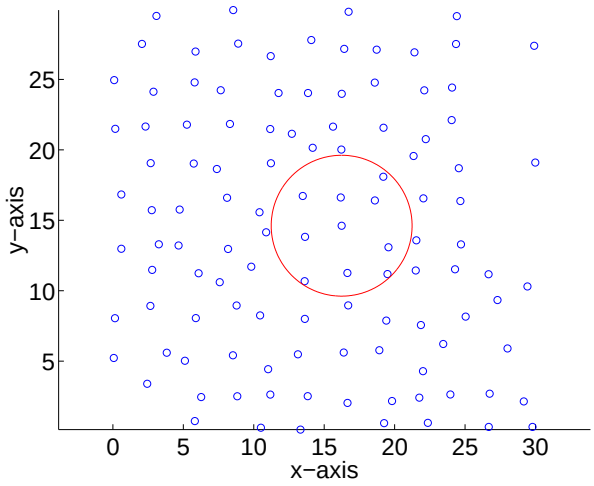
The network is a 30×30 unit^2 area. The *node-density* of the network, η, is defined to be the average number of nodes per footprint. We set $\eta = 10$ and hence the network contains 115 nodes. The nodes are initially distributed over an uniform grid and then the coordinate of each node is perturbed by a Gaussian distributed random number with zero mean and 0.5 variance. Figure 9(a) shows the resultant network. In order to avoid edge effects, we use a wrap-around topology in both the x and the y directions. The same network configuration has been used in all our simulations in order to avoid fluctuations in the simulation outcomes resulting from topology changes. Each node in this network independently generates a traffic of fixed-size packets. Packets at each node are generated independently according to a Poisson process with average rate λ. The *normalized load* on the network $\bar{\rho}$ is defined to be the average load per footprint, i.e., $\bar{\rho} = \lambda \times T \times \eta$, where $T = 19.4$ ms is the time to complete a communication

between any two nodes (this includes the transmission time of the DATA and control packets as well as inter-frame spacings). For each new packet, one of the neighbors of the source node is selected at random (uniformly) to be the destination. In order to isolate the effects of routing mechanisms from medium access issues, the destination of each packet is only one hop away (as in [18]). Each node uses a single *First-In First-Out* (FIFO) queue of infinite size. Therefore, the simulation results are not affected by the issues of finite buffer size. However, if a packet transmission attempt fails for `Max_retransmission_attempt` (`LongRetryLimit`), then the packet is dropped. If the first packet in the queue cannot be transmitted, then all other packets in the queue must wait. For each value of $\bar{\rho}$, the simulation is run for a sufficient amount time so that the network generates $115,000$ packets on aggregate.

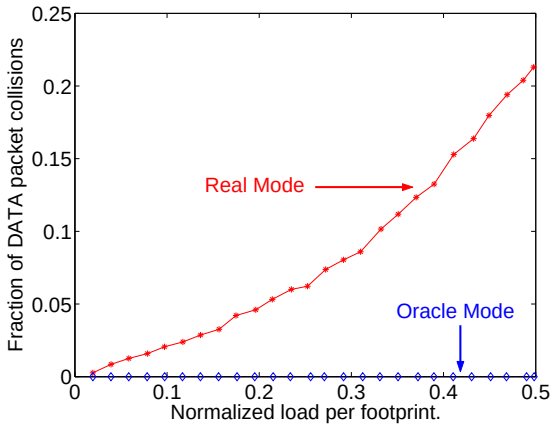
The simulator accurately follows the DSSS PHY specification of IEEE 802.11. The important parameters of the network used in the simulations are listed in Table I(b). In particular, the data rate is set to 1 Mbps and the size of *DATA* packets is fixed at 2300 bytes.

B. The Oracle Mode

In order to evaluate the impact of masked nodes on network performance, the simulator uses two modes: the *real mode* and the *oracle mode*. In the real mode, the simulator simulates the network the way it actually performs. On the other hand, the oracle mode simulates the network as if masked nodes did not exist. In this mode, the *RTS* and *CTS* packets sent by a node are heard by all the nodes in its range. The oracle mode is implemented as follows: every *RTS* or *CTS* sent to a node is stored in a cache, even if the packet would have been destroyed in reality. This cache is called the *oracle cache*. Before a node initiates any transmission attempt, it consults its oracle cache. If the cache contains a *RTS* or *CTS* that prohibits the node to transmit, then the node defers its transmission. This way, the oracle mode avoids all the collisions of *DATA* and



(a) The network used for the simulation. The circle represents the footprint of the node at the center of the circle.



(b) $\frac{\text{Number of } \textit{DATA} \text{ packet collisions}}{\text{Total number of } \textit{DATA} \text{ packets sent}}$

Fig. 9. Simulation results

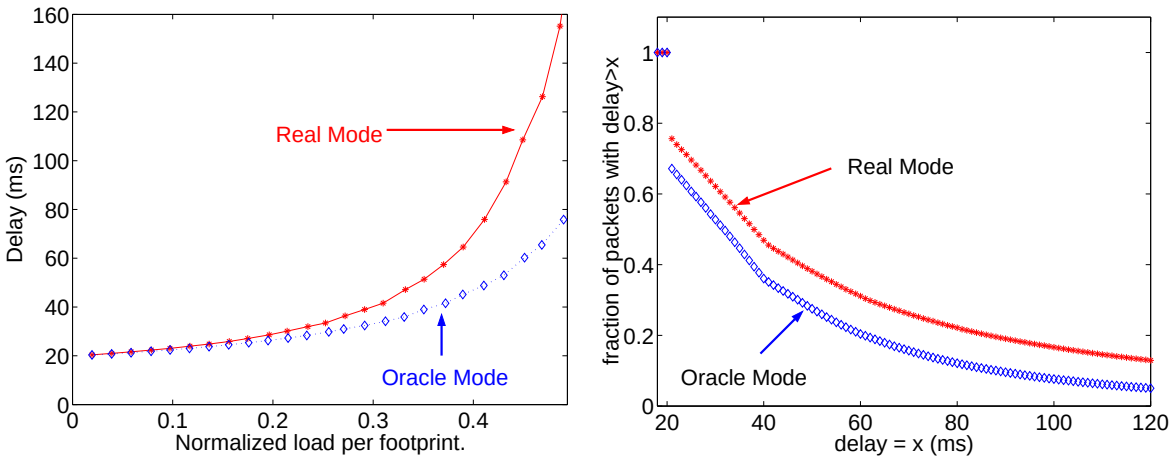
ACK packets.

C. Simulation Results

In this section we present our simulation results. We remind that the load $\bar{\rho}$ is normalized both in time and area. Thus, for instance, if the normalized load is $\bar{\rho} = 0.5$, then each node generates about 2.5 packets per second on average.

1) Collisions: Figure 9(b) shows the fraction of *DATA* packet collisions. If there were no masked nodes in the network, then none of the packets would collide. This is indeed the case as shown by all-zero collisions in the oracle mode.

The correct situation, however, is depicted by the real mode. We observe, for example, that at load $\bar{\rho} = 0.3$, about 10% of the *DATA* packets will collide due to the presence of masked nodes. It is worth pointing out that our analysis and experiments show that the same fraction of packet collisions occurs in a small linear network at a per-node load of about $\rho = 0.25$ (see Fig. 8).



(a) Average delay

(b) Complementary Cumulative Distribution Function at $\bar{\rho} = 0.4$.

Fig. 10. Comparison between the oracle and the real modes

Thus, despite very different node topology and traffic patterns, the simple network analysis is indeed predictive of the behavior of larger networks.

2) Delay: One of the most important impact of *DATA* packet collisions is on the packet delivery time. Suppose a packet enters a transmission queue at time t_1 and let the *ACK* for the packet be received successfully at time t_2. Then, the delivery time (or simply *delay*) for the packet is defined to be $(t_2 - t_1)$. Note that this represents the delay as perceived by the sending node. The delay calculation includes only successfully transmitted packets. If a collision occurs, the node backs off and then retransmits the packet. Retransmissions increase packet delivery time. Intuitively, we expect fewer retransmissions and smaller packet delivery time in the oracle mode.

Figure 10(a) plots the average delay versus the normalized load $\bar{\rho}$. The offset of about 20 ms at the bottom is due to the minimum time needed to consecutively exchange *RTS*, *CTS*, *DATA* and *ACK* packets. The simulation shows that for some values of traffic load, the average delay in the real mode can be as much as 100% higher than in the oracle mode. For example, at a

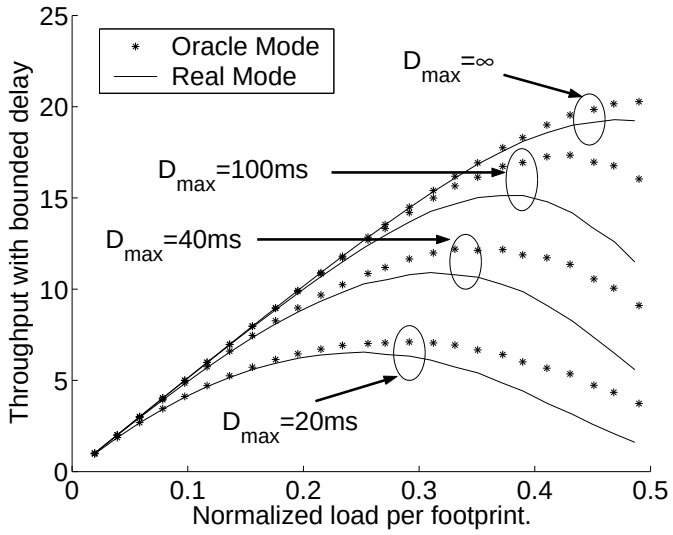


Fig. 11. Throughput, counting only packets delivered within a maximum delay bound. The four sets stand for a maximum delay bound of 20 ms, 40 ms, 100 ms and ∞ respectively.

normalized load of 0.5, the average delay in the real mode is about 160 ms whereas in the oracle mode it is only 80 ms.

Figure 10(b) shows the complementary cumulative distribution function of the delay at $\bar{\rho} = 0.4$. The figure shows that the real mode performs worse than the oracle mode for all threshold values of delay. In particular, the fraction of packets arriving with delay exceeding 120 ms is about 13% in the real mode while it is only 5% in the oracle mode. Masked nodes may, thus, render a wireless network unsuitable for multimedia traffic. The discontinuity in each curve near 20 ms shows that about 25% (35%) of packets arrive with the minimum possible delay for the real (oracle) mode.

3) Throughput: Throughput corresponds to the average number of successful DATA packet transmissions per second. However, in several applications, a late packet is useless. This could be due to the nature of the application, e.g., multimedia applications, or because of the retransmissions triggered by excessive delays, such as in TCP. In order to take packet delivery time into consideration, we define *Throughput(D_{max})* to be the number of packets transmitted per

second per footprint with delay smaller or equal to D_{max}. Every successful packet is counted in the calculation of *Throughput*(∞).

Figure 11 shows the simulation results. The outcomes for *Throughput*(D_{max}) with D_{max} equal to 20 ms, 40 ms, 100 ms and ∞ are plotted together. We observe a significant difference between the throughput obtained in the real and the oracle modes. For example, for $D_{max} = 100$ ms and $\bar{\rho} = 0.45$, the oracle mode achieves a throughput about 40% higher. These results show that masked nodes may also significantly affect the throughput performance of wireless LANs.

VI. Conclusion

In this paper, we have defined the masked node problem for wireless networks that rely on control packets, such as *RTS* and *CTS*, to avoid collisions. A node becomes masked when it is supposed to hear a packet, but cannot interpret it correctly due to other transmissions. We have shown that in an ad hoc network, a successful exchange of *RTS* and *CTS* is not sufficient to prevent *DATA* packet collisions, due to the presence of masked nodes. A masked node is, therefore, a "problematic node" in the same class as hidden nodes.

The impact of masked nodes on the performance of IEEE 802.11 ad hoc networks was thoroughly evaluated through real experiments, mathematical analysis, and simulations. The experiments clearly demonstrated that *RTS*/*CTS* exchanges fail to eliminate a significant number of packet collisions (about 10% to 15%), and that the presence of masked nodes is the major cause of the failure. We also derived a closed-form expression for the probability of collision, as a function of the traffic load. In the case of the masked node, this expression turned out to match very well the experimental data, over a wide range of traffic load values.

Since our analysis and experiments focused on a small (yet realistic) linear network topology, we have performed extensive simulations to also evaluate the impact of masked nodes on a

large network. One of the key aspects of the simulation was the definition of an oracle-mode network, in which every node hears *every RTS* or *CTS* packet it is supposed to receive even if the packet collides in reality. Thus, the oracle mode simulates the network as if masked nodes did not exist. This approach allowed us to appropriately evaluate the impact of masked nodes on network performance. We observed that collisions due to masked nodes result in retransmissions and subsequent increases in packet delivery time. In particular, we showed that the average delay may increase by as much as 100%, when comparing the real mode to the oracle mode. The fraction of *DATA* packet collisions observed in the simulation was on the same order as that observed in our experiments and analysis of the small linear network. Thus, our analysis provides a good insight into the significance of the masked-node problem in general.

We conclude by noting that masked nodes arise in ad hoc networks as a consequence of the fact that the radio receivers cannot decode overlapping signals reliably. Multi-user detections may help in alleviating this problem significantly in the future [19]. However, at present, commercially used WLAN cards generally do not implement multiuser detection techniques. Thus, the issue of whether a comprehensive solution to this problem is achievable with currently used medium access schemes is an open research area.

REFERENCES

[1] Zygmunt J. Haas, Jing Deng, Ben Liang, Panagiotis Papadimitratos, and S. Sajama, "Wireless ad hoc networks," in *Encyclopedia of Telecommunications*, John G. Proakis, Ed. Wiley, 2002.

[2] "ANSI/IEEE Std 802.11-1999 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 1999.

[3] Raphael Rom and Moshe Sidi, *Multiple Access Protocols: Performance and Analysis*, Springer Verlag, 1990.

[4] Leonard Kleinrock and Fouad A. Tobagi, "Packet switching in radio channels: Part 1 - Carrier Sense Multiple-Access modes and their throughput-delay characteristics," *IEEE Transactions on Communications*, vol. COM-23, no. 12, pp. 1400–1416, 1975.

[5] Fouad Tobagi and Leonard Kleinrock, "Packet switching in radio channels: Part 2 - the hidden node problem in carrier sense multiple access modes and the busy tone solution," *IEEE Transactions on Communications*, vol. COM-23, no. 12, pp. 1417–1433, 1975.

[6] Phil Karn, "MACA - a new channel access method for packet radio," in *ARRL/CRRL Amature Radio 9th Computer Networking Conference*, September 22 1990, pp. 134–140.

[7] Vaduvur Bharghavan, Alan Demers, Scott Shenker, and Lixia Zhang, "MACAW: A media access protocol for wireless LANs," in *Proceedings of ACM SIGCOMM '94*. 1994, pp. 212–225, ACM.

[8] Saikat Ray, Jeffrey B. Carruthers, and David Starobinski, "RTS/CTS-induced congestion in ad-hoc wireless LANs," in *IEEE Wireless Communication and Networking Conference (WCNC)*, March 2003, pp. 1516–1521.

[9] Kaixin Xu, Mario Gerla, and Sang Bae, "How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks?," in *IEEE Globecom*. 2002, IEEE.

[10] EunSun Jung and Nitin H. Vaidya, "A power control MAC protocol for ad hoc networks," in *ACM International Conference on Mobile Computing and Networking (MobiCom)*. Sept. 2002, ACM.

[11] Chane L. Fullmer and J.J. Garcia-Luna-Aceves, "Floor acquisition multiple access (FAMA) for packet-radio networks," in *Proceedings of SIGCOMM '95*. 1995, ACM.

[12] Shih-Lin Wu, Yu-Chee Tseng, and Jang-Ping Sheu, "Intelligent medium access for mobile Ad Hoc networks with busy tones and power control," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 9, pp. 1647–1657, September 2000.

[13] Zygmunt J. Haas, "On the performance of a medium access control scheme for the reconfigurable wireless networks," in *Proceedings of MILCOM'97*. 1997, IEEE.

[14] Donald Gross and Carl M. Harris, *Queuing Theory (3rd edition)*, John Wiley & Sons, 1998.

[15] James Murdock, *Perturbations: Theory and Methods*, Wiley, 1991.

[16] Aironet Wireless Communications, *PC4500/PC4800 Developer's Reference Manual*, 1997, Document number: 710-004247, Revision: B1.

[17] "SimEleven: An IEEE 802.11 MATLAB-based simulator.," Available at: `http://netlab1.bu.edu/~saikat`.

[18] Jeffrey P. Monks, Vaduvur Bharghavan, and W. Wen-mei Hwu, "A power controlled multiple access protocol for wireless packet networks," in *IEEE Infocom 2001*, March 2001, pp. 134–140.

[19] Sergio Verdu, *Multiuser Detection*, The Press Syndicate of the University of Cambridge, 1998.