Abstract—The hidden terminal problem is a well-known phenomenon in wireless networks. Mechanisms to eliminate the effect of hidden terminals have been proposed and standardized. These mechanisms are usually used, however, without being aware of the existence of any hidden terminals. Hidden terminal detection, on the other hand, provides such awareness, which is useful when deciding whether or when these standardized mechanisms should be used. This paper proposes two detection mechanisms: active detection and passive detection. Simulations based on IEEE 802.11-based ad hoc networks are carried out to evaluate the proposed schemes.

Index Terms—wireless networks, hidden terminal, passive detection, active detection, simulation.

I. INTRODUCTION

The hidden terminal (or node) problem [1] in wireless networks has been studied extensively in the literature ever since the 1970s. Various kinds of mechanisms have also been proposed to reduce or eliminate the effects of hidden terminals. One of these uses the exchange of Request-To-Send (RTS) and Clear-To-Send (CTS) frames to reduce the probability of collision of the subsequent DATA transmission and the final acknowledgement (ACK) frames. This four-way frame exchange protocol [2] is adopted as the solution for the IEEE 802.11 standard [3]. It is used optionally, as an enhancement to the mandatory “minimal frame exchange protocol” where only DATA and ACK frames are exchanged.

From our understanding, the four-way handshake represents only a hidden-node-unaware solution for hidden terminal elimination, in the sense that the RTS/CTS packets are exchanged no matter whether there is a hidden terminal in the vicinity or not. Hence, this mechanism is used preemptively “to be on the safe side”.

However, using RTS/CTS can be costly in terms of bandwidth consumption, and using it in situations where there are actually no hidden terminals, is wasteful. Moreover, as pointed out by many recent papers, e.g., [4-5], the problem of hidden terminals still remains more or less unsolved after employing the RTS/CTS mechanism.

Therefore, a better solution would be a preventive mechanism where a node knows explicitly whether there are hidden terminals or not and where the RTS/CTS mechanism is used only when necessary. To obtain the knowledge of hidden terminals in its vicinity, a node has to do hidden terminal detection. This issue is the main focus of this paper.

Even though hidden terminal detection is currently under discussion within the 802.11k (TGk) group of the IEEE [6], this topic has not received much attention in the wireless network research community at large. Furthermore, the solution proposed in 802.11k is what we refer to as a “receiver-initiated” scheme. We argue that this method is not very suitable for ad hoc networks, where a “sender-initiated” detection scheme is preferable.

In this paper, we propose two mechanisms to detect hidden terminals in 802.11-based wireless networks, namely, passive detection and active detection. After describing the detection principles in different situations, we conduct a series of simulations to evaluate the effectiveness and verify the appropriateness of these two mechanisms. The pros and cons of each mechanism are also discussed.

The rest of the paper is organized as follows. The principles of these two mechanisms are presented in Section II, followed by discussions on some relevant issues in the same section. The simulation scenarios and results are presented in Section III, and the concluding remarks are given in Section IV.

II. HIDDEN TERMINAL DETECTION

A. Hidden Terminal Problem Reiteration

Consider three wireless nodes depicted in Figure 1. Node 1 can hear both node 0 and node 2, but the two peripheral nodes, node 0 and node 2, cannot hear each other. Collisions may easily happen, at node 1, as soon as the two peripheral nodes send packets at the same time.

Figure 1. Illustration of the hidden terminal problem.

B. Initiation of Hidden Node Detection

The detection of hidden terminals can be initiated either by the receiver or the sender of the potentially colliding packets.

For example, if node 1 in Figure 1 wants to avoid collisions of traffic it is receiving, it initiates detection of neighbors that are hidden to each other. This is referred to as “receiver-initiated” detection. On the other hand, if node 0 in Figure 1 wants to avoid collision of traffic it is sending to node 1, it...
initiates detection of two-hop neighbors that are hidden to it. This is referred to as “sender-initiated” detection.

802.11k is particularly designed to accommodate the “receiver-initiated” detection. This scheme fits well to the infrastructure mode where all traffic within the Basic Service Set (BSS) goes between the Access Point (AP) and the other stations (STAs). The rationale behind this scheme is that no nodes in the BSS are hidden to the AP (i.e., node 1 in Figure 1), and all collisions due to hidden nodes will happen at the AP. Hence, it makes sense that the AP observes the possibilities for collisions of traffic it is receiving from hidden node pairs amongst its neighbors, and determines whether or not RTS/CTS should be used within the BSS.

In ad hoc networks, however, collision due to hidden terminals can occur at any node in the network, and the need for RTS/CTS protection may vary from situation to situation. Due to bandwidth scarcity in wireless networks, eliminating unnecessary use of RTS/CTS can be beneficial. We argue therefore that the “sender-initiated” solutions would often be preferable in ad hoc networks.

Note, moreover, that even in the receiver-initiated scheme of 802.11k, they are the senders of the potentially colliding packets (i.e., the neighbors of the AP) that in the end have to detect the nodes that are hidden to them. How this should be done is not specified by the standard. Hence, the sender-initiated schemes presented in the following — although primarily presented in the context of ad hoc networks - can also be beneficial with 802.11k in an infrastructure BSS.

C. Detection Mechanisms for Ad Hoc Networks

In the context of sender-initiated hidden terminal detection, a node that wishes to perform hidden terminal detection is hereafter referred to as a detecting node. Two detection mechanisms are presented in this subsection: passive detection and active detection.

1) Mechanism I: Passive Detection. Passive detection is only possible when there is ongoing traffic exchange between the detecting node’s one-hop neighbors and the potential hidden terminals of the detecting node. With passive detection, a detecting node does not generate any traffic itself, but solely relies on monitoring the ongoing traffic in the neighborhood to obtain the picture of its hidden terminals. The only requirement for conducting a passive detection is that the detecting node’s network interface has to be set in promiscuous mode in which all packets, regardless of destinations, are processed by the detecting node. The process of setting the network interface to promiscuous mode can simply be done in the configuration of the network interface and this process can also easily be reversed.

Figures 2 and 3 show two different cases of passive detection, depending on whether RTS/CTS is being used or not by the transmitting neighboring nodes. In both figures, nodes 0 and 2 are hidden terminals to each other. When RTS/CTS is disabled (Figure 2), the detecting node (node 0) is able to hear the DATA frame sent by its immediate neighbor (node 1), but not the ACK frame sent by node 2. (According to 802.11, a node that has successfully received a unicast DATA packet destined for it, must reply with an ACK.) Since it does not receive the ACK, node 0 suspects that node 2 is a hidden terminal to it. Node 2 is confirmed as a hidden node after having recorded several subsequent missing ACKs by node 0. The MAC address of the hidden terminal is extracted from the receiver field of the DATA frame.

Figure 2. Passive detection – without RTS/CTS.

Figure 3 illustrates the other case where RTS/CTS is used in the neighborhood of the detecting node. Here node 2 is attempting to send DATA packets to node 1. As depicted in the figure, the detecting node (node 0) can only hear the CTS and the ACK frames sent by node 1, which are supposed to arrive after the RTS and DATA frames. Based on this information, node 0 concludes that there is a hidden terminal in its vicinity and extracts correspondingly the MAC address of the hidden terminal. If the background traffic is initiated from node 1 and destined at node 2, the detecting node receives the RTS and DATA frames, but not the CTS and ACK frames. The hidden terminal can also be easily detected in the same way.

Figure 3. Passive detection – with RTS/CTS.

2) Mechanism II: Active Detection. Ongoing background traffic in the neighborhood of the detecting node is an obvious prerequisite for passive detection. If this is not the case, active detection is the only workable option.

Figure 4 illustrates the active detection mechanism, where node 0 is the detecting node. Two new types of packets, referred to as ‘detection request’ and ‘detection probe’ packets, have been introduced. With active detection, each node that wishes to know its potential hidden terminals will actively generate a detection request to all its one-hop neighbors. (Neighbor information can be obtained by for instance exchanging Hello messages defined in ad hoc routing.
protocols.) All neighbor nodes that receive this request will then start sending a sequence of unicast probe packets to their neighbors, for a time interval specified in the detection request. The detecting node will then perform measurements on the traffic generated by the neighbors. Based on the received packets, the detecting node is able to establish a complete list of all its hidden terminals, including those that may not be discovered by passive detection. Comparing Figure 4 with Figures 2 and 3, the difference is that active detection is triggered by the detecting node itself. Other operational procedures, such as receiving traffic in promiscuous mode and performing address extraction, are basically the same as for the passive detection mechanism.

D. Discussions on a few relevant aspects of detection

1) Missing ACK Frames. The procedure described above raises a number of questions. First, how does the detecting node conclude that an ACK is missing? According to 802.11, a node transmitting a DATA frame assumes that an ACK is received within the time interval of SIFS + ACK_time + DIFS [3]. Thus, if the detecting node does not hear the ACK within the same interval, it concludes that the ACK is missing. Another natural question is: How does the detecting node know that the reason for an ACK being missed is the hidden terminal effect? In fact, the reason for a missing ACK might as well be that the original DATA frame was corrupted so that it never triggered the ACK to be sent. Alternatively, the ACK might be corrupted so that the detecting node never receives it. Thus, if only judging from one single missing ACK, the detecting node might conclude that a number of nodes are hidden terminals despite that they are within the hearing range.

To eliminate the effect of such “false hidden terminals”, the detecting node has to detect a number of missing ACKs before it should conclude that a node is a hidden terminal. To handle this problem adequately, we propose the following rules, which form a basis for our simulation study presented later:

- When the detecting node observes a DATA (or a detection probe) frame from a neighbor, it monitors the channel status while waiting for the ACK. When the waiting time of SIFS + ACK_time + DIFS expires without having received the ACK, it only concludes that the ACK is missing if the channel has been idle during the entire waiting time. Otherwise, it cannot conclude that the ACK was sent outside its radio range, because the ACK might as well have been corrupted while the channel was sensed busy.

- When the detecting node observes a DATA (or a detection probe) frame from a neighbor with a missing ACK it does not conclude immediately that the destination of the DATA frame is hidden. Instead, it waits until the reception of a subsequent DATA frame sent from the neighbor to the same destination. If the retry field in the subsequent frame shows it is a new DATA frame, the detecting node concludes that the missing ACK was received correctly by its neighbor and increases the counter for missing ACKs by one. Having recorded a certain number of missing ACKs, the node adds the address of the destination node to its hidden terminal list.

- If the retry field shows that a DATA frame is a retransmitted frame, the detecting node waits for the same period of time, and won’t increase the counter for missing ACKs until the subsequent frame is a new DATA frame. If a missing ACK was not detected prior to the new DATA frame, the detecting node cannot conclude that the ACK was missing.

2) Missing DATA Frames. If a detecting node (e.g., node 0 in Figure 3) receives only ACKs, not DATA (or detection probe) frames, it checks the duration field in the MAC header. If this value is zero, which indicates a completed DATA frame or the final segment of a long packet, the detecting node increases the counter for missing DATA frames by one. Having recorded a certain number of missing DATA frames, the node extracts the receiver address from the ACK frame, and adds it to its hidden terminal list.

3) Hidden Terminal List. As the result of the detection, each detecting node maintains an updated hidden terminal list. A new node is appended to the list once it is detected as a hidden terminal, if it is not already on the list. It may happen sometimes that a detecting node suddenly receives ACKs (or DATA frames) from another node that is already on its hidden terminal list. This means likely that either the location or the transmission power of the hidden terminal has changed. This node should then be removed from the hidden terminal list.

4) Passive Detection versus Active Detection. The advantage of using the passive detection mechanism is that no extra protocol overhead is introduced. It might for example be a useful way of maintaining the hidden terminal list. However, as not all the hidden terminals can be detected in this way, active detection might be required now and then to reconstruct an exhaustive list of hidden terminals. Since active detection introduces additional protocol overhead, it might often make sense to limit the usage of active detection, and complement it with passive detection.

The amount of extra overhead introduced by active detection is actually scenario dependent. For a stationary ad hoc network, each node, either the detecting node or the other involved nodes, needs only to generate dozens of packets for detecting its hidden terminals. For a mobile ad hoc network, especially with high mobility, active detection is required more frequently in order to get a timely picture of all hidden terminals. If all nodes inside an ad hoc network would require the knowledge of hidden terminals, the additional traffic of active detection is expected to be noticeable. Therefore, there is a tradeoff between using active detection on the one hand, and using RTS/CTS under all circumstances (i.e., without the need for hidden terminal detection at all), on the other hand.

III. SIMULATION SCENARIOS AND RESULTS

To validate the feasibility and the accuracy of the two detection mechanisms, three sets of simulations for hidden terminal detection are carried out. The ns2 network simulator [7] is used, with some code modifications such as setting the interfaces in promiscuous mode. The MAC addresses of the hidden terminals, represented by their node IDs in the simulation, can thus be read from the packet headers.
The goal of this set of simulations is to verify whether a node can passively detect any, and possibly all, potential hidden terminals around itself. The result of the detection is a hidden terminal report at each node, which summarizes the nodes that are detected as hidden to it.

Figure 5 shows a static network topology used in the simulations, with five stationary nodes located in a two dimensional area. The distance between nodes 1 and 4 is slightly higher than 150 meters. The distance between any other two neighboring nodes is exactly 150 meters. With the sensing range set to 200 meters, any pair of nodes that are separated by two hops are hidden to one another. As background traffic, three flows are generated by nodes 1, 3 and 4 respectively, and destined to nodes 2, 2 and 5 correspondingly (Figure 5). All three flows are UDP sessions sent at a Constant Bit Rate (CBR) of 59 Kbps (Table 1). Two variations of the passive detection mechanism are studied separately as follows.

1) Without RTS/CTS. In this case, a node can detect the existence of a hidden terminal based on its incomplete reception of a DATA and ACK pair.

Of the detection mechanism described in Section 2, there is a parameter on the number of missing DATA or ACK frames for the detecting node to confirm there is a hidden terminal. This parameter is empirically set as 8 in our simulations. In practice, however, this parameter should be adjustable. An optimal value for this parameter could be achieved by joint consideration of factors such as network topology, traffic pattern and intensity, mobility, and so on.

Take node 1 in Figure 5 as an example for the case where a node receives only ACK frames for detection. It receives only the ACK frames sent from node 2 in response to the DATA frames received from node 3, not the DATA frames directly from node 3 that triggered the ACK frames. In this way, node 1 is able to detect node 3 as a hidden terminal to itself.

The established hidden terminal reports from the simulation are summarized in Table 2, for all nodes listed in Figure 5.

2) With RTS/CTS. As expected, the results in this case are exactly the same as above, as listed in Table 2. The only difference is that the IDs of the hidden terminals are obtained from the RTS or the CTS frames when RTS/CTS is enabled.

B. Simulation Set II – Active Detection

From the simulated results on passive detection described in the previous subsection, we can easily observe that the obtained hidden terminal reports are not exhaustive. A hidden terminal cannot be discovered if background traffic in the neighborhood is insufficient. For example, node 5 in Figure 5 is not able to detect node 1 as its hidden terminal since there is no ongoing traffic between node 4 and node 1.

In our simulation, the lengths of the detection request packet and the detection probe packet are empirically set as 100 bytes and 60 bytes respectively. Upon receiving any request packet from a detecting node, the requested node generates a sequence of 20 probe packets, at an interval of 50 ms. Again in practice the optimal values for the number of probe packets and the length of the interval should be scenario-dependent.

The simulation scenario for active detection is the same as the one for passive detection, as illustrated in Figure 5, except that the ongoing packets are request and probe packets in this case. Take node 2 as an example this time. Node 2 notices that no hidden terminals are reported by passive detection and decides to send a hidden node request to all its neighbors. From the corresponding traffic generated, node 2 finds that it has a hidden node with node ID 4. Table 3 shows the simulation results for active detection for all nodes in Figure 5. It is observed that - in contrast to the passive detection mechanism - the active detection mechanism detects all of the hidden terminals. The cost of detecting all hidden terminals is obviously the extra overhead imposed by the request and probe packets.

<table>
<thead>
<tr>
<th>Node</th>
<th>One-hop neighbor</th>
<th>Hidden terminal list</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2, 4</td>
<td>3, 5</td>
</tr>
<tr>
<td>2</td>
<td>1, 3</td>
<td>No hidden node observed</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1, 5</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>No hidden node observed</td>
</tr>
</tbody>
</table>

Table 1. Simulation Configuration.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routing protocol</td>
<td>AODV</td>
</tr>
<tr>
<td>Transmission range</td>
<td>200 meters</td>
</tr>
<tr>
<td>Carrier Sense range</td>
<td>200 meters</td>
</tr>
<tr>
<td>Background Sense range</td>
<td>200 meters</td>
</tr>
<tr>
<td>Background traffic</td>
<td>CBR/UDP</td>
</tr>
<tr>
<td>Packet size</td>
<td>350 bytes</td>
</tr>
<tr>
<td>Packet interval time</td>
<td>0.05 seconds</td>
</tr>
<tr>
<td>Background source rate</td>
<td>59.2 Kbps</td>
</tr>
<tr>
<td>Nominal data rate</td>
<td>11 Mbps</td>
</tr>
<tr>
<td>Simulation duration</td>
<td>100 seconds</td>
</tr>
</tbody>
</table>

Table 2. Hidden terminal reports for passive detection.
Table 3. Hidden terminal reports for active detection.

<table>
<thead>
<tr>
<th>Node</th>
<th>One-hop neighbor</th>
<th>Hidden terminal list</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2, 4</td>
<td>3, 5</td>
</tr>
<tr>
<td>2</td>
<td>1, 3</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1, 5</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

C. Simulation Set III – Detection for Mobile Nodes

Finally, a set of simulations is targeted at investigating how the detection mechanisms work in a mobile scenario. More specifically, the detection is performed on a mobile node that is traveling across a network. Only results with passive detection are presented here due to the 5-page limit.

The mobile scenario is depicted in Figure 6, with 3 fixed nodes (nodes 1, 2, and 3) and 1 mobile node (node 4). There are two ongoing CBR/UDP sessions, one from node 1 to node 2 and another from node 3 to node 2 respectively. The distance between nodes 1 and 2, and between nodes 2 and 3 is 150 meters, and node 4 is 190 meters away from the horizontal line connecting nodes 1-2-3. Node 4 moves from left to right at a constant low velocity, and during the course of the simulation, it is always within the reach of at least one of the other three nodes. Around the left-most position, the mobile node is able to receive the DATA packets sent from node 1 to node 2, and identifies node 2 as its hidden node. At the position in the middle, the mobile node is able receive ACKs from node 2, but not the DATA frames from nodes 1 and 3. It detects two hidden terminals during this period. At the right-most position, similar to at the left-most position, the mobile node is able to detect node 2 as its hidden terminal.

The overall hidden terminal report for node 4 is illustrated in Figure 7, with the number of hidden nodes on the y-axis and simulation time on the x-axis. The result confirms that the detection mechanism works fine also for mobile nodes.

IV. CONCLUSIONS AND FUTURE WORK

Even though the hidden terminal problem is a well-known phenomenon in wireless network, how to detect potential hidden terminals has not received much attention so far. In this paper, we have presented and studied two mechanisms for hidden terminal detection, referred to as passive detection and active detection respectively. The passive detection mechanism does not impose any extra traffic into the network, and relies purely on background traffic between the detecting node’s one-hop neighbors and their neighbors. The disadvantage of this mechanism is that it gives an incomplete picture of the potential hidden terminals. Active detection, on the other hand, generates certain amount of additional traffic into the network and requires collaboration from its neighboring nodes. This mechanism reveals all potential hidden terminals of the detecting node, at a cost of extra bandwidth consumption.

Although the proposed mechanisms for hidden node detection are presented in the context of ad hoc networks, the same mechanisms can also be applied in the infrastructure BSS WLAN configurations. Our proposal complements the 802.11k standardization effort, as it shows how the hidden terminal reports can be established. Any node, including an AP, can use these mechanisms to retrieve hidden node reports from any of its neighbors.

Finally, more analytical and simulation studies regarding detection overhead, parameters optimization, and power consumption for both passive and active detection, as well as the tradeoff mentioned in Subsection II.D, need to be carried out quantitatively, especially for larger-scale networks with various types of mobility.

REFERENCES