The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks

Wenyuan Xu, Wade Trappe, Yanyong Zhang and Timothy Wood
Wireless Information Network Laboratory (WINLAB)
Rutgers University, 73 Brett Rd., Piscataway, NJ 08854
wenyuan, trappe, yyzhang, twood@winlab.rutgers.edu

ABSTRACT

Wireless networks are built upon a shared medium that makes it easy for adversaries to launch jamming-style attacks. These attacks can be easily accomplished by an adversary emitting radio frequency signals that do not follow an underlying MAC protocol. Jamming attacks can severely interfere with the normal operation of wireless networks and, consequently, mechanisms are needed that can cope with jamming attacks. In this paper, we examine radio interference attacks from both sides of the issue: first, we study the problem of conducting radio interference attacks on wireless networks, and second we examine the critical issue of diagnosing the presence of jamming attacks. Specifically, we propose four different jamming attack models that can be used by an adversary to disable the operation of a wireless network, and evaluate their effectiveness in terms of how each method affects the ability of a wireless node to send and receive packets. We then discuss different measurements that serve as the basis for detecting a jamming attack, and explore scenarios where each measurement by itself is not enough to reliably classify the presence of a jamming attack. In particular, we observe that signal strength and carrier sensing time are unable to conclusively detect the presence of a jammer. Further, we observe that although by using packet delivery ratio we may differentiate between congested and jammed scenarios, we are nonetheless unable to conclude whether poor link utility is due to jamming or the mobility of nodes. The fact that no single measurement is sufficient for reliably classifying the presence of a jammer is an important observation, and necessitates the development of enhanced detection schemes that can remove ambiguity when detecting a jammer. To address this need, we propose two enhanced detection protocols that employ consistency checking. The first scheme employs signal strength measurements as a reactive consistency check for poor packet delivery ratios, while the second scheme employs location information to serve as the consistency check. Throughout our discussions, we examine the feasibility and effectiveness of jamming attacks and detection schemes using the MICA2 Mote platform.

MobiHoc'05, May 25–27, 2005, Urbana-Champaign, Illinois, USA.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and Protection*

General Terms

Security

Keywords

Denial of Service, Jamming, Jammer detection

1. INTRODUCTION

Wireless networks are progressively becoming more affordable, and consequently are being deployed in a variety of different modalities, ranging from wireless local area networks to mesh and sensor networks. As these networks gain popularity, providing security and trustworthiness will become an issue of critical importance. Many wireless security threats may be addressed through appropriately designed network security architectures [1, 10, 11, 13, 24, 27, 34], which are essentially modifications of traditional security services, such as confidentiality, authentication, and integrity to the wireless domain. Wireless networks, however, are susceptible to threats that are not able to be adequately addressed via cryptographic methods. One serious class of such threats are attacks of radio interference.

The shared nature of the wireless medium, combined with the commodity nature of wireless technologies and an increasingly sophisticated user-base, allows wireless networks to be easily monitored and broadcast on. Adversaries may easily observe communications between wireless devices, and just as easily launch simple denial of service attacks against wireless networks by injecting false messages. Traditionally, denial of service is concerned with filling user-domain and kernel-domain buffers [12]. However, in the wireless domain, the adversary is empowered to launch more fundamentally severe types of denial of service that block the wireless medium and prevents other wireless devices from even communicating.

Radio interference attacks are not addressable through conventional security mechanisms. An adversary can simply disregard the medium access protocol and continually transmit on a wireless channel. By doing so, he either prevents users from being able to commence with legitimate MAC operations, or introduces packet collisions that force repeated backoffs, or even jams transmissions. Such MAC and PHY-layer security threats for wireless networks have been known for some time, and the issue of MAC-layer weaknesses in 802.11 has been revisited by a recent announcement by the Australian CERT [2].

In order to ensure the dependability of future deployments of wireless networks, mechanisms are needed that will allow wireless networks of all types to cope with the threat of attacks of radio interference, or simply RF jamming attacks. The first stage to defending a wireless network is to understand what types of attacks are feasible, and how these attacks may be diagnosed. This paper examines how radio jamming may be conducted, and explores the task of detecting jamming attacks. The ability of wireless devices to detect that they are jammed allows the wireless network to identify regions of poor radio conditions, and therefore take an appropriate response to such threats, such as routing around these regions or more restorative mechanisms, such as channel surfing and spatial retreats [33].

We begin in Section 2 by presenting an overview of the jamming problem, as well as introducing several different adversarial models for jamming regions of a wireless network. In Section 3, we discuss different measurements that might be used to detect a radio interference attack, and explore the situations in which these attacks can and cannot be accurately identified as a jamming attack. In order to address the insufficiency of the individual measurements for detecting a jamming attack, in Section 4 we introduce two detection schemes that build upon packet delivery ratio measurements by incorporating signal strength readings or location information to serve as the basis for consistency checking in detecting the presence of jamming. We review related literature in Section 5 and present conclusions in Section 6.

2. JAMMING ATTACK MODELS AND THEIR EFFECTIVENESS

In this section, we introduce radio interference attacks that may be launched against wireless networks. The adversary (or the malicious wireless device) that launches such attacks is referred to as the *jammer* in this paper. We first define the characteristics of a jammer's behavior, and then enumerate metrics that can be used to measure the effectiveness of a jamming attack. These metrics are closely related to the ability of a radio device to either send or receive packets. We then introduce four typical jammer attack models, though by no means all-inclusive, which represent a broad range of attack strategies, and will serve as the basis for our discussion throughout the remainder of the paper. Throughout this paper, we will use the Berkeley MICA2 Mote platform for conducting our experiments with jammers. The observed characteristics of the jammers and the detection schemes presented later should hold for different wireless platforms, such as 802.11.

2.1 Jamming Characteristics and Metrics

Although several studies [23,31–33] have targeted jamming-style attacks, the definition of this type of attack remains unclear. A common assumption is that a jammer continuously emits RF signals to fill a wireless channel, so that legitimate traffic will be completely blocked [32, 33]. We believe, however, that a broader range of behaviors can be adopted by a jammer. For example, a jammer may remain quiet when there is no activity on the channel, and start interference as soon as it detects a transmission. The common characteristic for all jamming attacks is that their communications are not compliant with MAC protocols. Therefore, *we define a jammer to be an entity who is purposefully trying to interfere with the physical transmission and reception of wireless communications.*

The objective of a jammer is to interfere with legitimate wireless communications. A jammer can achieve this goal by either preventing a real traffic source from sending out a packet, or by preventing the reception of legitimate packets. Let us assume that A and B denote two legitimate wireless participants, and let us denote X to be the jammer. A legitimate participant may be unable to send out packets for many reasons. To name just a couple, X can continuously emit a signal on the channel so that A will never sense the channel as idle, or X can keep sending out regular data packets and force A to receive junk packets all the time. On the other hand, however, even if A successfully sends out packets to B, it is possible for X to blast a radio transmission to corrupt the message that B receives. We thus define the following two metrics to measure the effectiveness of a jammer:

- **Packet Send Ratio (PSR):** The ratio of packets that are successfully sent out by a legitimate traffic source compared to the number of packets it intends to send out at the MAC layer. Suppose A has a packet to send. Many wireless networks employ some form of carrier-sensing multiple access control before transmission may be performed. For example, in the MAC protocol employed by Mica2, the channel must be sensed as being in an idle state for at least some random amount of time before A can send out a packet. Further, different MAC protocols have different definitions on an idle channel. Some simply compare the signal strength measured with a fixed threshold, while others may adapt the threshold based on the noise level on the channel. A radio interference attack may cause the channel to be sensed as busy, causing A's transmission to be delayed. If too many packets are buffered in the MAC layer, the newly arrived packets will be dropped. It is also possible that a packet stays in the MAC layer for too long, resulting in a timeout and packets being discarded. If A intends to send out n messages, but only m of them go through, the PSR is $\frac{m}{n}$. The PSR can be easily measured by a wireless device by keeping track of the number of packets it intends to send and the number of packets that are successfully sent out.

- **Packet Delivery Ratio (PDR):** The ratio of packets that are successfully delivered to a destination compared to the number of packets that have been sent out by the sender. Even after the packet is sent out by A, B may not be able to decode it correctly, due to the interference introduced by X. Such a scenario is an unsuccessful delivery. The PDR may be measured at the receiver B by calculating the ratio of the number of packets that pass the CRC check with respect to the number of packets (or preambles) received. PDR may also be calculated at the sender A by having B send back an acknowledge packet. In either case, if no packets are received, the PDR is defined to be 0.

2.2 Jamming Attack Models

There are many different attack strategies that a jammer can perform in order to interfere with other wireless communications. As a consequence of their different attack philosophies, these various attack models will have different levels of effectiveness, and may also require different detection strategies. While it is impractical to cover all the possible attack models that might exist, in this study, we discuss a wide range of attacks that have proven to be effective in disrupting wireless communication. Specifically, we have designed and built the following jammers:

- **Constant jammer:** The constant jammer continually emits a radio signal. We have implemented a constant jammer using two types of devices. The first type of device we used is a waveform generator which continuously sends a radio signal. The second type of device we used is a normal wireless device. In this paper, we will focus on the second type, which we built on the MICA2 Mote platform. Our constant jammer continuously sends out random bits to the channel without following any MAC-layer etiquette. Specifically, the constant jammer does not wait for the channel to become idle before transmitting. If the underlying MAC protocol determines whether a channel is idle or not by comparing the signal strength measurement with a fixed threshold, which is usually lower than the signal strength generated by the constant jammer, a constant jammer can effectively prevent legitimate traffic sources from getting hold of channel and sending packets.

- **Deceptive jammer:** Instead of sending out random bits, the deceptive jammer constantly injects regular packets to the channel without any gap between subsequent packet transmissions. As a result, a normal communicator will be deceived into believing there is a legitimate packet and will be duped to remain in the receive state. For example, in TinyOS, if a preamble is detected, a node remains in the receive mode, regardless of whether that node has a packet to send or not. Hence, even if a node has packets to send, it cannot switch to the send state because a constant stream of incoming packets will be detected. Further, we also observe that it is adequate for the jammer to only send a continuous stream of preamble bits (0xAA in TinyOS) rather than entire packets.

- **Random jammer:** Instead of continuously sending out a radio signal, a random jammer alternates between sleeping and jamming. Specifically, after jamming for t_j units of time, it turns off its radio, and enters a "sleeping" mode. It will resume jamming after sleeping for t_s time. t_j and t_s can be either random or fixed values. During its jamming phase, it can either behave like a constant jammer or a deceptive jammer. Throughout this paper, our random jammer will operate as a constant jammer during jamming. The distinction between this model and the previous two models lies in the fact that this model tries to take energy conservation into consideration, which is especially important for those jammers that do not have unlimited power supply. By adjusting the distribution governing the values of t_j and t_s, we can achieve various levels of tradeoff between energy efficiency and jamming effectiveness.

- **Reactive jammer:** The three models discussed above are active jammers in the sense that they try to block the channel irrespective of the traffic pattern on the channel. Active jammers are usually effective because they keep the channel busy all the time. As we shall see in the following section, these methods are relatively easy to detect. An alternative approach to jamming wireless communication is to employ a reactive strategy. For the reactive jammer, we take the viewpoint that it is not necessary to jam the channel when nobody is communicating. Instead, the jammer stays quiet when the channel is idle, but starts transmitting a radio signal as soon as it senses activity on the channel. As a result, a reactive jammer targets the reception of a message. We would like to point out that a reactive jammer does not necessarily conserve energy because the jammer's radio must continuously be on in order to sense the channel. The primary advantage for a reactive jammer, however, is that it may be harder to detect.

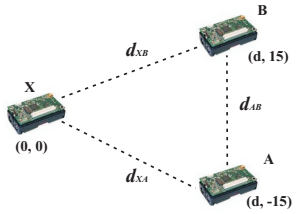


Figure 1: Placement of the Motes during jammer effectiveness experiments.

We have implemented the above four jammer models using Berkeley Motes that employ a ChipCon CC1000 RF transceiver and use TinyOS as the operating system. We disabled channel sensing and back off operations to bypass the MAC protocol, so that the jammer can blast on the channel irrespective of other activities that are taking place. The level of interference a jammer causes is governed by several factors, such as the distance between the jammer and a normal wireless node, the relative transmission power of the jammer and normal nodes, and the MAC protocol employed by normal nodes. The closer a jammer is to a node, or the higher transmit power it employs, the greater the impact it will have on network operation. The MAC protocols employed by the network also play a role. Usually, MAC protocols decide the channel is idle if the measured signal strength value is lower than a threshold. Many MAC protocols, such as the one in TinyOS release 1.1.1, uses a fixed threshold value. Some MAC protocols, however, such as BMAC [25], adapt the threshold value based on the measured signal strength values, i.e. they choose the minimum signal strength among the most recent n readings when channel is idle as the current threshold value. Consequently, if a constant jammer transmits at a constant power, and both the jammer and the nodes are static, these adaptive MAC protocols will consider the channel as idle since they will regard the energy emitted by the jammer as ambient noise. In addition to these network configuration parameters, the impact of a jammer is also affected by jammer-specific parameters, such as the sleep interval for a random jammer. In order to understand the interactions of these parameters and quantify the impact of a jammer in different scenarios, we conducted a set of experiments involving three parties: A, B, and X, where A and B are normal wireless nodes with A being the sender, B the receiver, and X a jammer using one of our four models. The transmission power levels employed by A, B, X are all $-4dBm$. These three nodes are carefully placed so that X has the same impact on both A and B. In particular, we set d_{XA}, the distance between X and A, equal to d_{XB}, the distance between X and B, and we fixed the distance between the sender A and the receiver B at $d_{AB} = 30$ inches, as depicted in Fig. 1.

The resulting PSR and PDR for each jammer model are summarized in Table 1. As the Table 1 shows, if A employs 1.1.1 MAC, a constant jammer that is reasonably close to A can completely block A, from sending out packets, re-

Constant Jammer					
d_{XA} (inch)		BMAC		1.1.1 MAC	
		PSR (%)	PDR (%)	PSR (%)	PDR (%)
38.6		74.37	0.43	1.00	1.94
54.0		77.17	0.53	1.02	2.91
72.0		99.57	93.57	0.92	3.26
Deceptive Jammer					
d_{XA} (inch)		BMAC		1.1.1 MAC	
		PSR (%)	PDR (%)	PSR (%)	PDR (%)
38.6		0.00	0.00	0.00	0.00
54.0		0.00	0.00	0.00	0.00
72.0		0.00	0.00	0.00	0.00
Random Jammer					
d_{XA} (inch)		BMAC		1.1.1 MAC	
		PSR (%)	PDR (%)	PSR (%)	PDR (%)
t_j = U[0,31] t_s = U[0,31]	38.6	79.45	0.26	70.19	16.77
	44.0	80.15	17.48	70.30	21.95
	54.0	80.43	99.00	76.98	99.75
t_j = U[0,31] t_s = U[1,8]	38.6	60.47	0.06	56.49	0.00
	44.0	60.72	47.41	56.00	0.41
	54.0	61.77	96.75	100.0	99.64
Reactive Jammer					
d_{XA} (inch)		BMAC		1.1.1 MAC	
		PSR (%)	PDR (%)	PSR (%)	PDR (%)
m = 7bytes	38.6	99.00	0.00	100.0	0.00
	54.0	100.0	99.24	100.0	99.87
	72.0	100.0	99.35	100.0	99.97
m = 33bytes	38.6	99.00	0.00	100.0	0.00
	44.0	99.00	58.05	100.0	87.26
	54.0	99.25	98.00	100.0	99.53

Table 1: The resulting PSR and PDR for different jammer models under various scenarios.

sulting in a very low PSR. However, if A employs BMAC, which adapts the threshold based on the surrounding signal strength, A can still manage to send out a large portion of the packets, i.e, with PSR being 74.37% even when X is only 38.6 inches away from A. The reason why A cannot send out all of the packets is that the signal strength produced by X varies with time. The corresponding PDR in both cases, however, is poor because most of the packets are corrupted by the constant jammer, especially when the constant jammer is close to the sender.

However, the same trend cannot be observed for a deceptive jammer. Since a deceptive jammer continuously sends out packets with valid preamble, both A and B are forced to constantly stay in the reception mode no matter which MAC protocol they use. Hence, A and B cannot send out any packets at all and the PSR are 0% all the time. PDR in this case is defined as 0.

For the random jammer, in addition to studying the impact of network configuration parameters, such as the distance between the jammer and the nodes, and the MAC protocol on the effectiveness of the jammer, we also look at jammer-specific parameters, such as the on-off periods. Specifically, we studied two random jammers. For the first random jammer, the duration of the jamming period t_j is a uniform random number between 0 and 31 spibus interrupts in TinyOS [9], denoted by t_j = U[0,31], and the duration of the sleeping period t_s is a uniform random number between 0 and 31 as well, denoted by t_s = U[0,31]. For the second random jammer, t_j = U[0,31], and t_s = U[1,8]. On average, the second jammer sleeps less, and switches to the jamming mode more often. Thus, the PSR measured in the second jammer scenario is less than the PSR in the first jammer scenario. Additionally, since the random jammer alternates between jamming and sleeping, BMAC, which always chooses the minimum signal strength value among the recent readings, cannot increase the threshold quickly enough to consider the channel idle. Thus, BMAC considers the channel as busy when the random jammer is jamming, resulting in a lower PSR.

A reactive jammer starts interference as soon as it hears a transmission on the channel. Consequently, the effectiveness of a reactive jammer is also dependent on size of legitimate network packets as well as the size of packet the jammer emits. In Table 1, we explore the behavior of the reactive jammer for network packets of size $m = 7$ and $m = 33$ bytes, where the jammer emitted a 20 byte jamming packet. First, we observe that in all cases the sender is able to reliably send out its packets. Ideally, if m is short, one would infer that there may not be enough time for a reactive jammer to corrupt a network packet in transmission. However, as we see in Table 1, for different network packet sizes, although there is a difference in the resulting PDR, the difference is in fact negligible. Hence, even for short packets of a few bytes in length, a jammer employing the reactive strategy is able to effectively disrupt network communication.

3. BASIC STATISTICS FOR DETECTING JAMMING ATTACKS

Detecting jamming attacks is important because it is the first step towards building a secure and dependable wireless network. It is challenging because jammers can employ different models, and it is often difficult to differentiate a jamming scenario from legitimate scenarios. Specifically, we need to differentiate a jamming scenario from various network conditions: congestions that occur when the aggregated traffic load exceeds the network capacity so that the packet send ratio and delivery ratio are affected; the interrupt of the communication due to failures at the sender side, etc.

In this section, we present several measurements that may be employed by wireless devices for the purpose of detecting jamming attacks. We explore these measurements in detail and present scenarios where they may not be effective in detecting a jamming attack, and in fact could cause false detections. For each of these measurements, we develop statistics upon which to make decisions. Since statistics built upon individual measurements may lead to false conclusions, in Section 4 we develop two improved detection strategies. These two detection strategies are both built upon the fundamental assumption that communicating parties should have some basis for knowing what their characteristics should be if they are not jammed, and consequently can use this as a basis for differentiating jammed scenarios from mere poor link conditions.

3.1 Signal Strength

One seemingly natural measurement that can be employed to detect jamming is signal strength, or ambient energy. The rationale behind using this measurement is that the signal strength distribution may be affected by the presence of a jammer. In practice, since most commodity radio devices do not provide signal strength or noise level measurements that are calibrated (even across devices from the same manufacturer), it is necessary for each device to employ its own empirically gathered statistics in order to make its decisions. Each device should sample the noise levels many times during a given time interval. By gathering enough noise level measurements during a time period prior to jamming, network devices can build a statistical model describing normal energy levels in the network.

We now explore two basic strategies that employ signal strength measurements for detecting a jamming attack. The first approach uses either the average signal value or the total signal energy over a window of N signal strength measurements. This is a simple approach that extracts a single statistic for basing a hypothesis test upon. Since a single

statistic loses most of the shape characteristics of the time series, a second strategy would seek to capture the shape of the time series by representing its spectral behavior. The second strategy that we discuss uses N samples to extract spectral characteristics of the signal strength for the basis of discrimination. In the discussion below, we assume that we have measured the channel's received energy levels $s(t)$ at different times and collected N of these samples to form a window of samples $\{s(k), s(k-1), \cdots, s(k-N+1)\}$.

3.1.1 Basic Average and Energy Detection

We can extract two basic statistics from signal strength readings, namely, the average signal strength and the energy for detection. In both cases, the statistical hypothesis testing problem is binary and essentially involves deciding between signal absent and signal present hypotheses.

The use of the signal average arises naturally when the jammer emits a constant amplitude signal. In this case, the detection statistic is $T(k) = (\sum_{j=k-N+1}^{k} s(j))/N$. The use of the signal energy arises when the jammer emits a powerful noise-like signal, such as a white Gaussian process. Here, the detection statistic is $T(k) = (\sum_{j=k-N+1}^{k} s(j)^2)/N$. In either case, the detection decision is made by comparing $T(k)$ to a threshold γ that is suitably chosen by considering tradeoffs between probability of detection and false alarm, such as through application of Neyman-Pearson theorem [14, 26].

3.1.2 Signal Strength Spectral Discrimination

The average signal strength or the signal energy over a window of N samples does not reflect the fact that there may be many different received signal sample paths that could have led to the same mean or energy value. For example, a signal that has half of its ADC values as 50 and half as 150 would be considered the same as a signal whose samples are all 100 if we use the average signal strength as our decision statistic.

In order to have more robustness to false decisions and enhance the ability to classify scenarios, it is natural to use spectral discrimination techniques to classify the signal. One possible spectral discrimination mechanism is to employ higher order crossings (HOC). We refer the reader to the treatise on HOC [15] for explicit definition of HOC statistics. We have chosen to study higher order crossings since the calculation of these statistics only involves differences between samples, and is thus simple and practical to implement on resource-constrained wireless devices, such as sensor nodes. More complicated spectral techniques that involve the estimation of power spectral densities are possible and yield comparable performance but require more computational complexity.

Effectiveness Analysis: In order to understand the effect that a jammer would have on the received signal strength, we performed six experiments. In the first two experiments, we have two Motes, a sender A and a receiver B, which are 30 inches apart from each other. In the first case, A transmits 20 packets per second, corresponding to a traffic rate of 5.28kbps, which we refer to as a CBR source. In the second case, A transmits at its maximum rate; as soon as the send function returns to the application level asynchronously, either because the packet is successfully sent or because the packet is dropped (the packet pumping rate is larger than the radio throughput), it posts the next send function. Such a sender is referred to as a MaxTraffic source, and corresponds to a raw traffic rate of 6.46kbps. In the following four experiments, in addition to A and B, we introduced the jammer X, which was placed 54 inches away from B, with X em-

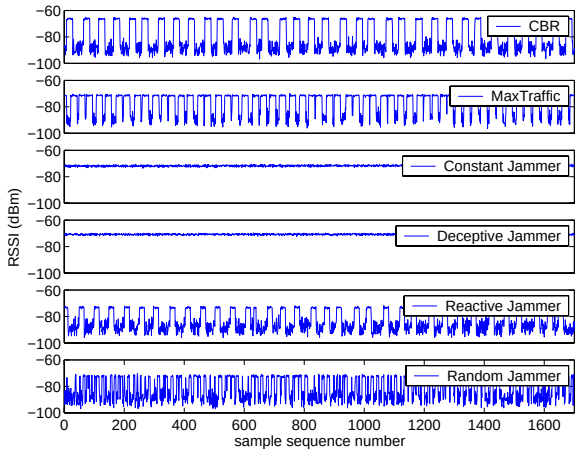


Figure 2: RSSI readings as a function of time in different scenarios. RSSI values were sampled every 1msec.

ploying our four jammer models. When X behaves as a random jammer, it uses the following parameters: t_j = U[0,31] and t_s = U[0,31]. In these four jamming scenarios, A is a CBR source. In each of these six experiments, the receiver B obtains the RSSI values by posting the `RSSIADC.getData()` function on the port `TOS_ADC_CC_RSSI_PORT` every millisecond. The reported RSSI values in Fig. 2, in dBm, are converted from the raw values following the analog-to-digital conversion of the received voltage levels [6]. We present time series data for each of the six scenarios in Fig. 2. From these results, we observed that the average values for the constant jammer and the MaxTraffic source scenario, are roughly the same. Further, the constant jammer and deceptive jammer have roughly the same average values, with the slight difference in the plot resulting from experimental setup. Additionally, the signal strength average from a normal CBR source does not differ much from that measured for the reactive jammer scenario. Similar statements can be made for using the signal energy. These results suggest the following important observation: we may not be able to use simple statistics, such as average signal strength or energy, to discriminate jamming scenarios from normal traffic scenarios because it is not straightforward to devise a threshold that can separate these two scenarios.

There is a practical issue that arises from the locations the nodes and jammers relative to each other. Nodes that are very close to each other will naturally lead to high signal strength measurements, while nodes separated by more distance will yield lower signal strength measurements.

From the time series in Fig. 2, we observe that there are some differences in the shapes underlying the time series for these scenarios. For example, the measured signal strength for the constant jammer and the deceptive jammer exhibit a much lower variation (the time series curve is almost flat) compared to the signal strengths for MaxTraffic source.

We next examined the issue of whether spectral discrimination techniques would be able to distinguish between normal and jammed scenarios. We calculated the first two higher order crossings for the time series, D_1 and D_2, using a window of 240 samples. We plot D_1 versus D_2 in Fig. 3. From the Fig. 3 (a), we observe that the points gather in two clusters, one cluster corresponding to the constant and deceptive jammers, while the other cluster corresponding to normal CBR and MaxTraffic sources. Hence, using

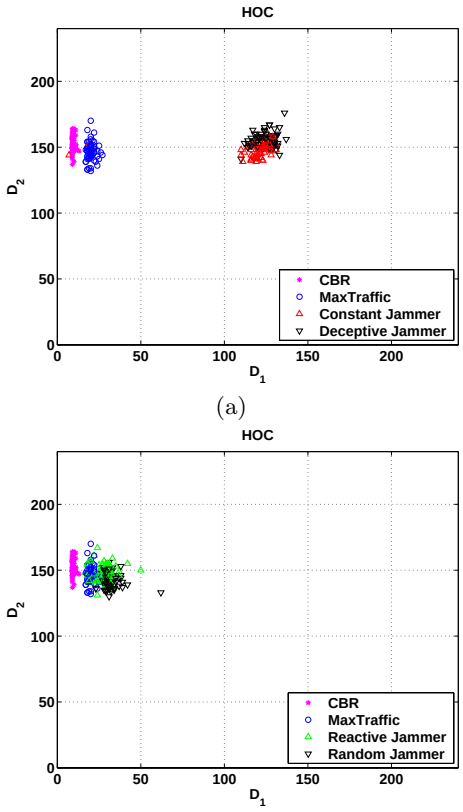


Figure 3: Plot of the first two higher order crossings, D_1 vs. D_2, for different jammer and communication scenarios.

HOC, we can distinguish normal traffic scenarios from the constant and deceptive jammer. However, examining Fig. 3 (b) we see that we cannot distinguish the reactive or random jammer from normal traffic scenarios. The reason for this is that a reactive jammer or random jammer causes the channel state to alternate between busy and idle in much the same way as normal traffic behaves. In particular, because the reactive jammer does not change the underlying busy and idle periods for a normal traffic scenario, it is particularly difficult to distinguish between signal readings for a reactive jammer and signal readings from the underlying traffic.

Hence, based on these observations, we conclude that employing HOC (or even other spectral methods), will work for some jammer scenarios, but are not powerful enough to detect all jammer scenarios.

3.2 Carrier Sensing Time

As discussed in Section 2, a jammer can prevent a legitimate source from sending out packets because the channel might appear constantly busy to the source. In this case, it is very natural for one to keep track of the amount of time it spends waiting for the channel to become idle, i.e. the carrier sensing time, and compare it with the sensing time during normal traffic operations to determine whether it is jammed. We would like to emphasize that this is only true if the legitimate wireless node's MAC protocol employs a fixed signal strength threshold to determine whether the channel is busy or idle. For protocols that employ an adaptive threshold, such as BMAC, after the threshold has adapted to the ambient energy of the jammer, the carrier sensing time will be small even when a jammer is blasting on the channel. Consequently, in the rest of this section, we only focus on MAC protocols that employ a fixed threshold, such as the MAC in TinyOS 1.1.1.

In most forms of wireless medium access control, there are rules governing who can transmit at which time. One popular class of medium access control protocols for wireless devices are those based on carrier sense multiple access (CSMA). CSMA is employed in MICA2 Motes as well as in both infrastructure and infrastructureless (ad hoc) 802.11 networks. The MAC-layer protocol for 802.11 additionally involves an RTS/CTS handshake. During normal operation of CSMA, when A (the sender) tries to transmit a packet, it will continually sense the channel until it detects the channel is idle, after which it will wait an extra amount of time (known as the propagation delay) in order to guarantee the channel is clear. Then, if RTS/CTS is used it will send the RTS packet, or otherwise will send the data packet. Suppose we assume that the adversary X continuously emits radio signal on a channel and that A attempts to transmit a packet. Then, since the channel is occupied by X, A will either time-out the channel sensing operation (if a time-out mechanism is available in the MAC protocol) or be stuck in the channel sensing mode.

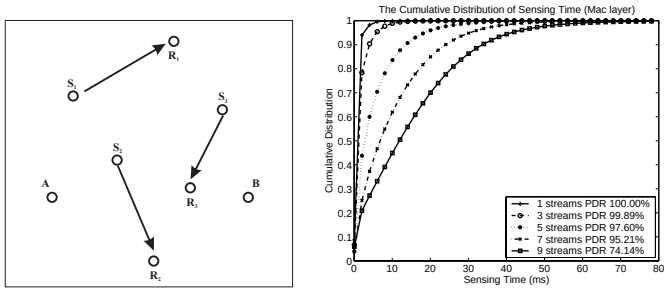
Unfortunately, a large carrier sensing time could have occurred in non-jammed scenarios as well, such as congestion. It is therefore important to have some mechanism to distinguish between normal and abnormal failures to access the channel. In order to do so, a thresholding mechanism based on the sensing time can be used to identify jamming: Each time A wishes to transmit, it will monitor the time spent sensing the channel, and if that time is above a threshold (or if it is consistently above the threshold), it will declare that a jamming is occurring. The threshold may be determined theoretically based on a simple channel occupancy model, or empirically. The problem with theoretically calculating the threshold is that it is extremely difficult to build a complete mathematical model that captures a realistic MAC protocol. A well-known $M/M/1/1$ queuing model may be used to describe the MAC protocol [16, 17, 33], but doesn't capture the notion of collisions, or retransmissions. Therefore, we focus on the second approach to determining the threshold, which involves each network device collecting statistics regarding the amount of time D that a device must wait before it can start transmission during normal, or even somewhat congested, network conditions. With a distribution $f_D(d)$ describing carrier sensing times during acceptable network conditions, we may classify any new measured sensing time as either normal or anomalous by employing significance testing [26]. In this case, our null hypothesis is that the measured delay D corresponds to the distribution $f_D(d)$. If we reject the null hypothesis, then we conclude the network is experiencing a jamming attack. Since it is undesirable to falsely conclude the presence of jamming when the network is merely experiencing a glitch, we need to use a conservative threshold to reduce the probability of a false positive.

Effectiveness Analysis: In order to quantify the validity of detecting jamming at the MAC-layer using carrier sensing time, we carried out several simulation based studies using the ns-2 simulator with 802.11 extensions. We modified ns-2 by disabling the MAC layer retransmission so that we could focus our investigation on the channel sensing behavior. In our experiments we have two nodes, A and B. Once every 19 msecs, node A senses the channel by trying

to send out a beacon to node B. We obtain the channel sensing time D by calculating the difference between the time when beacon packets reach the MAC-layer and the time when the MAC successfully senses the channel as idle and sends out RTS. In order to capture the statistical behavior of the sensing time, we calculate the corresponding cumulative distribution for several scenarios involving different levels of background traffic loads. As shown in Fig. 4(a), we introduce several streams (from sender S_i to receiver R_i) that are within the radio range of A and B in order to increase the background traffic. Each stream's traffic represents an MPEG-4 video stream suitable for a wireless video application. We use traffic statistics corresponding to the movie Star Wars IV [8], where packet sizes are governed by an exponential distribution with a mean size of 268 bytes, and the packet inter-arrival times following an exponential distribution with mean 40msecs, resulting in each stream having an average traffic rate of 53.6Kbps. The corresponding cumulative distributions of D are shown in Fig. 4(b). These observations can be explained as follows. When there are only a few streams, there are few nodes competing for the channel, and node A can get the channel quickly with high probability. As the number of streams increases, the competition for the channel becomes more intense, thus taking longer for A to acquire the channel.

From this figure, we can observe that when the number of streams is less than 7, the curves approach 1 quickly before D equals 40 msecs. Even in the case of 9 streams, which has an average PDR of 74.1% and corresponds to a very poor quality of service, over 99% of all observed transmission delays occur within 60 msecs. However, for the constant jammer, the time taken to acquire the channel will be large relative to normal MAC-sensing times, or even the times observed for poor QoS conditions. Choosing an appropriate threshold for the MAC-sensing time will allow the algorithm to be robust to false detections. For example, if we would like to ensure, with 99% confidence, that our sensing time is a jamming attack and not a result of a normal background with a PDR of 75%, we should choose the threshold as 60 msecs.

To study the effect of different jammers on the carrier sensing time in a real wireless network, we performed an experiment using two Motes, X and A. Here, Mote A corresponds to a network node trying to send out a 33-byte packet every 100msecs, and which measures the sensing time while doing so. Mote A employed the MAC protocol from TinyOS release 1.1.1, which used a fixed threshold for determining idleness. Mote X cycles through the four different types



(a) (b)

Figure 4: The MAC-layer sensing time experiment: (a) basic underlying experimental setup, (b) cumulative distributions of D for different traffic scenarios and the corresponding packet delivery ratio.

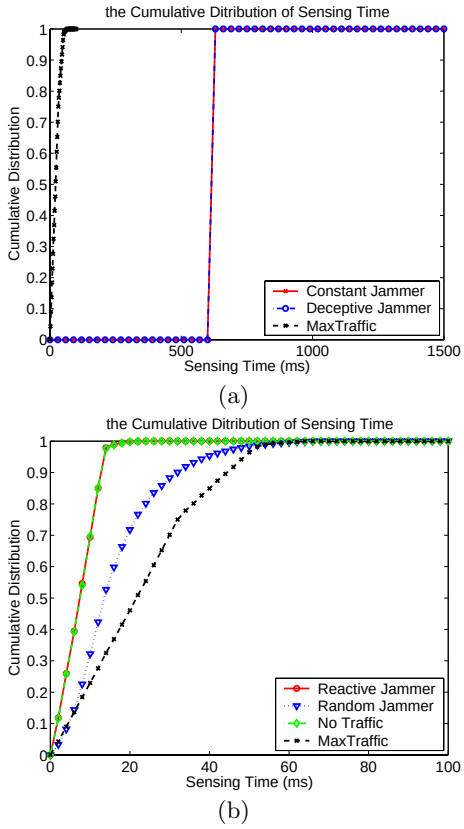


Figure 5: The cumulative distribution for the carrier sensing times measured using MICA2 Motes.

of jammers, as well as the MaxTraffic source. Additionally, we measured the sensing time when there is no background traffic, i.e. X does not send any traffic.

Fig. 5 depicts the cumulative distribution of the sensing time for the six different scenarios. Fig. 5 (a) shows that the cumulative distribution of the constant jammer and the deceptive jammer jumps at the point where the sensing time equals to 640msces. This is caused by a timeout we added to the TinyOS. In our experiment, if the device does not start to send the packet within 640msecs after the packet was passed to the MAC-layer from application layer, a timeout will occur, the packet will be discarded, and its sensing time will be counted as 640msecs.

The drawback of carrier sensing time is that it exhibits significant missed detections in the presence of other types of jammers. As Fig. 5 (b) shows, most of the sensing time in other jammer scenarios is smaller than the sensing time in a congested scenario. The reactive jammer will exhibit normal carrier sensing times because the jammer will not attempt to jam until another node has successfully started transmission. As a result, the transmitting node A will observe normal carrier sensing times. In particular, in our experiment the reactive jammer produces sensing time cumulative distributions that overlap completely with the case of no background traffic.

We note that, if the MAC protocol employs an adaptive threshold for determining channel idleness, instead of the fixed threshold in our experiment, then the cumulative distribution of the sensing time for the constant jammer would have shifted to the left, while there would have been no difference for the deceptive jammer since node A would still

have been locked in a received state. The reactive jammer would have exhibited the same characteristics. Similar to the constant jammer, the random jammer also shifts the cumulative distribution to the left. We verified these observations through identical experiments to the ones described above where we used BMAC instead of the MAC protocol from TinyOS release 1.1.1.

In summary, both signal strength and carrier sensing time, under certain circumstances, can only detect the constant jammer and deceptive jammer. Neither of these two statistics is effective in detecting the random jammer or the reactive jammer.

3.3 Packet Delivery Ratio

A jammer may not only prevent a wireless node from sending out packets, but may also corrupt a packet in transmission. Consequently, we next evaluate the feasibility of using packet delivery ratio (PDR) as the means of detecting the presence of jamming. The packet delivery ratio can be measured in the following two ways: either by the sender, or by the receiver. At the sender side, the PDR can be calculated by keeping track of how many acknowledgements it receives from the receiver. At the receiver side, the PDR can be calculated using the ratio of the number of packets that pass the CRC check with respect to the number of packets (or preambles) received. Unlike signal strength and carrier sensing time, PDR must be measured during a specified window of time where a baseline amount of traffic is expected. If no packet is received over that time window, then the PDR within that window is zero.

Since a jamming attack will degrade the channel quality surrounding a node, the detection of a radio interference attack essentially boils down to determining whether the communication node can send or receive packets in the way it should have had the jammer not been present. More formally, let us use π_0 to denote the PDR between a sender and a receiver, who are within radio range of each other, assuming that the network only contains these two nodes and that they are static. As shown in Table 1, any one of the four jammers, if placed within a reasonable distance from the receiver, can cause the corresponding PDR to become close to 0. In the cases shown in Table 1, π_0 is 100%. From these results, we can conclude that a jammer can cause the PDR to drop significantly. We would like to point out that a non-aggressive jammer, which only marginally affects the PDR, does not cause noticeable damage to the network quality and does not need to be detected or defended against.

Next, we need to investigate how much PDR degradation can be caused by non-jamming, normal network dynamics, such as congestion, failures at the sender side, etc. In order to study the impact of congestion on PDR, we introduced 3 MaxTraffic sources, resulting in a raw offered traffic rate of 19.38kbps[1], to model a rather highly congested scenario. Even under such a congestion level, the PDR measured by the receiver is still around 78%. As a result, a simple thresholding mechanism based on the PDR value can be used to differentiate a jamming attack, regardless of the jamming model, from a congested network condition.

Though PDR is quite effective in discriminating jamming from congestion, it is not as effective for other network dynamics, such as a sender battery failure, or the sender moving out of the receiver's communication range, because these dynamics can result in sudden PDR drop in much the same way as a jammer does. Specifically, if the sender's battery drains out, it stops sending packets, and the corresponding PDR is 0%.

Consequently, compared to signal strength and carrier sensing time, PDR is a powerful statistic in that it can be used to differentiate a jamming attack from a congested network scenario, for different jammer models. However, it still cannot differentiate the jamming attack from other network dynamics that can disrupt the communication between the sender and the receiver.

[1]At 100% duty cycle, the MICA2 radio's maximum bandwidth capacity is 12.364kbps, though the effective maximum throughput is typically much less than that.

4. JAMMING DETECTION WITH CONSISTENCY CHECKS

In the previous section we saw that no single measurement is capable of detecting all kinds of jamming attacks. Since the purpose of a jammer is to influence the channel quality between a node and its neighbors, it is not reasonable, or needed, to try to detect a jammer if that jammer does not effectively interfere with the receipt/send of packets at a node. While a node losing its sending ability is a clear sign that it is being jammed, a weak reception capability (i.e. a low PDR) can be caused by several factors besides jamming, such as a low link quality due to the relatively large distance between the sender and the receiver.

We observed in the previous section that PDR is a powerful measurement that is capable of discriminating between jammed and congested scenarios, yet is unable to identify whether an observed low PDR is due to natural causes of poor link quality. In order to compensate for this drawback, and enhance the likelihood of detection, we will examine two strategies that build upon PDR to achieve enhanced jammer detection. We augment the use of PDR by applying signal strength measurements to conduct consistency checking to determine whether low PDRs are due to natural causes or due to radio interference. Later, in Section 4.2, we discuss a complementary technique that uses location information to augment PDR measurements for jamming detection.

Throughout this section, we assume that a node is only responsible for detecting whether it is jammed, and is not responsible for detecting the jammed condition of its neighbors. This follows from the fact that a wireless node is the best source of information regarding its local radio environment and is a less reliable predictor of the radio condition at distant locations. We assume that each node maintains a neighbor list, obtained from the routing layer, which will assist in making more reliable detection decisions. Additionally, we assume that the deployment of the network is sufficiently dense to guarantee that each node has several neighbors. All legitimate nodes in the network will participate in the detection protocol by transmitting a baseline amount of traffic, e.g. by sending heartbeat beacons. This allows each node to reliably estimate PDR over a window of time, and conclude that the PDR is 0 if no packets are observed during that time period.

4.1 Signal Strength Consistency Checks

The packet delivery ratio serves as our starting point for building the enhanced detector. Rather than rely on a single PDR measurement to make a decision, we employ measurements of the PDR between a node and each of its neighbors. In order to combat false detections due to legitimate causes of link degradation, we use the signal strength as a consistency check. Specifically, we check to see whether a low PDR value is consistent with the signal strength that is measured. In a normal scenario, where there is no interference or software faults, a high signal strength corresponds to a high PDR. However, if the signal strength is low, which

```
Algorithm: PDRSS_Detect_Jam
{PDR(N) : N ∈ Neighbors} = Measure_PDR()
MaxPDR = max{PDR(N) : N ∈ Neighbors}
if MaxPDR < PDRThresh then
    SS = Sample_Signal_Strength()
    CCheck = SS_ConsistencyCheck(MaxPDR, SS)
    if CCheck == False then
        post NodeIsJammed()
    end
end
```

Algorithm 1: Jamming detection algorithm that checks the consistency of PDR measurements with observed signal strength readings.

means the strength of the wireless signal is comparable to that of the ambient background noise, the PDR will be also low. On the other hand, a low PDR does not necessarily imply a low signal strength. It is the relationship between signal strength and PDR that allows us to differentiate between the following two cases, which were not possible to separate using just the packet delivery ratio. First, from the point of view of a specific wireless node, it may be that all of its neighbors have died (perhaps from consuming battery resources or device faults) or it may be that all of a node's neighbors have moved beyond a reliable radio range. A second case would be the case that the wireless node is jammed. The key observation here is that in the first case, the signal strength is low, which is consistent with a low PDR measurement. While in the jammed case, the signal strength should be high, which contradicts the fact that the PDR is low. Table 2 summarizes typical network scenarios that can cause low PDR values and how the signal strength measurements can help further isolate the cause of the low PDR values.

Based on these observations we propose the detection protocol shown in Algorithm 1. In the `PDRSS_Detect_Jam` algorithm, a wireless node will declare that it is not jammed if at least one of its neighbors has a high PDR value. However, if the PDRs of all the neighbors are low, then the node may or may not be jammed and we need to further differentiate the possibilities by measuring the ambient signal strength. Rather than continually sample the ambient signal levels, which may use precious energy and processor cycles, the function `Sample_Signal_Strength()` instead reactively measures the signal strength values for a window of time after the PDR values fall below a threshold (the threshold we have identified in Section 3.3), and returns the maximum value of the signal strengths during the sampling window[2], which is denoted as SS. We note that the duration of the sampling window should be carefully tuned based upon the traffic rate, the jamming model, the measuring accuracy, and the desired detection confidence level.

The function `SS_ConsistencyCheck()` takes as input the maximum PDR value of all the neighbors, denoted as $MaxPDR$, and the signal strength reading SS. A consistency check is performed to see whether the low PDR values are consistent with the signal strength measurements. If the signal strength SS is too large to have produced the observed $MaxPDR$ value, then `SS_ConsistencyCheck()` returns False, else it returns True.

The consistency check may be conducted empirically as follows. During deployment, or during a guaranteed time of non-interfered network operation, a table (PDR, SS) of

[2]In order to prevent spurious readings and have improved stability, in practice we use the average of the top three signal strength readings.

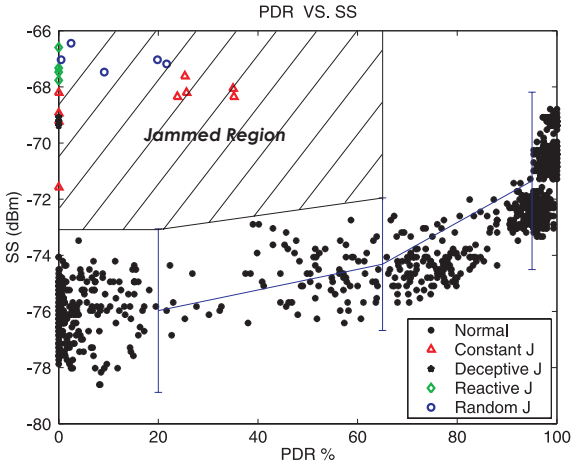


Figure 6: The (PDR, SS) measurements, indicating the relationship between PDR and signal strength. Also presented are the (PDR, SS) values measured for different jammers. The data was binned into three PDR regions, $(0, 40)$, $(40, 90)$ and $(90, 100)$, and the corresponding 99% confidence intervals are presented. The shaded region is the jammed-region, and corresponds to (PDR, SS) values that are above the 99% signal strength confidence intervals and whose PDR values are less than 65%.

packet delivery ratios and signal strength values are measured. We may divide the data into PDR bins and calculate the mean and variance for the data within each bin. Or, we may conduct a simple regression to build a relationship between PDR and SS. The output of the binning or the regression is a relationship from which we may calculate an upper bound for the maximum SS that would have produced a particular PDR value in a non-jammed scenario. Using this bound, we may partition the (PDR, SS) plane into a benign-region and a jammed-region.

We conducted an experiment using MICA2 Motes to validate Algorithm 1. We gathered (PDR, SS) values for a source transmitting to a receiver node at a power level of roughly -5dBm. The PDR values were calculated using a window of 200 packets, while the SS values were sampled every 1msec for 200msecs in order to provide sufficient resolution to capture the jammer behavior during a reactive jammer attack. The packets were 33 byte long and transmitted at a rate of 20 packets per second. The source receiver separation was varied in order to produce a full spectrum of normal (PDR, SS) values, as depicted in Fig. 6. Using these values, we found the 99% SS confidence bars values for $(0, 40)$ $(40, 90)$ and $(90, 100)$ PDR regions. We depict these confidence bars, and define the corresponding jammed-region to be the region of (PDR, SS) that is above the 99% signal strength confidence intervals and whose PDR values are less than 65%. The jammed-region is shaded and appears in the upper-left corner of Fig. 6. We then performed experiments where we introduced the different jammers. The reactive jammer that we used sent out a 20-byte long interference packet as soon as it detects activities on the channel, while the random jammer had $t_j = \mathrm{U}[0,31]$ and $t_s = \mathrm{U}[0,31]$. We varied the source-receiver configurations as well as the location of the jammer, and measured the resulting PDR and SS values. As can be seen in Fig. 6, the

Observed PDR	Observed signal strength	Typical scenarios
PDR = 0 (no preamble is received)	low signal strength	non-jammed: neighbor failure, neighbor absence, neighbors being blocked, etc.
PDR = 0 (no preamble is received)	high signal strength	node jammed
PDR low (packets are corrupted)	low signal strength	non-jammed: neighbor being faraway
PDR low (packets are corrupted)	high signal strength	node jammed

Table 2: A combination of PDR and signal strength improves jamming detection accuracy.

(PDR, SS) values for all jammers distinctively fall within the jammed-region.

It is to be noted that the jammer in this experiment had a transmission power level of roughly -4dBm, which is stronger than that of the source. In fact, in order for the jammer to be more effective, it needs to operate at a relatively higher power level. However, a jammer using higher power will further decrease the PDR value and increase the SS measurement, thus pushing the resulting (PDR, SS) pair further towards the upper left corner, making it more distinct the benign-region. On the other hand, a jammer that operates on a lower power level is not as effective in interfering with the network operations. As a result, the combination of PDR and signal strength is quite powerful in discriminating a jammed scenario from various network conditions.

4.2 Location Consistency Checks

We now discuss a second consistency checking algorithm for detecting the presence of a radio interference attack. Whereas `PDRSS_Detect_Jam` employs signal strength to validate PDR measurements, the `LOC_Detect_Jam` algorithm employs location information. In addition to the assumptions listed earlier, for `LOC_Detect_Jam` we also assume that all legitimate neighbor nodes transmit with a fixed power level, such as the default settings when the sensor or ad hoc network was originally deployed. While this assumption holds for many real network settings, we would like to point out that scenarios where nodes have varying transmission powers can be addressed by easy extensions to our algorithm.

In `PDRSS_Detect_Jam`, the sampling granularity and the window length for measuring signal strength are two parameters that must be carefully set based upon the assumed jammer models as well as the underlying network traffic conditions. As noted earlier, it may not be practical to sample the signal strength with a fine granularity over a long window of time, and for this reason `PDRSS_Detect_Jam` employs a reactive consistency checking strategy in the sense that signal strength measurements are performed after PDR measurements fall below a threshold.

Instead of employing a reactive consistency check, the `LOC_Detect_Jam` algorithm uses a proactive consistency check. Rather than a node reacting to conduct measurements, the location consistency checking scheme involves information that is already made available to the wireless node prior to determining that PDR values are suspicious. As a consequence of this, the granularity and window length at the detector is no longer an issue. We note, in our specification of `LOC_Detect_Jam` that, although we require each node to transmit a location advertisement message, the issue of window length and granularity of signal strength sampling has been translated from a complicated issue involving assumptions regarding the adversary's attack model into an issue regarding a node's mobility. As shall be seen, the analogous notion of position message frequency may be simply addressed using knowledge of node mobility and an assumption regarding the nominal packet delivery ratio of the network.

The `LOC_Detect_Jam` protocol requires the support of a localization infrastructure, such as GPS [7], or other localization techniques [3, 19, 22], which provides location information to wireless devices. We assume that this localization infrastructure is not able to be attacked or exploited by potential adversaries. Recently, countermeasures have been proposed to protect localization services from being exploited by adversaries [5, 20, 21]. In the `LOC_Detect_Jam` protocol, we again use PDR as the metric indicating link quality. A node will decide its jamming status by checking its PDR and deciding whether the observed PDR is consistent with what it should see given the location of its neighbor nodes. Conceptually, neighbor nodes that are close to a particular node should have high PDR values, and if we observe that all nearby neighbors have low PDR values, then we conclude that the node is jammed.

In our protocol, we let every node periodically advertise its current location and further let each node keep track of both the PDR and the location of its neighbors. Due to node mobility, it is necessary that the location advertisements occur with sufficient frequency to be able to reliably capture the migration of neighbors from regions of high PDR near node A to regions of lower PDR further from node A. If a jammer suddenly comes into the network near node A, then the location information that node A has will correspond to the location of the neighbors prior to the start of the interference. Analogous to `PDRSS_Detect_Jam`, if node A finds that the PDR values of all of its neighbors are below the threshold $PDRThresh$, then node A will perform a consistency check by using the position P_n of the neighbor who had the maximum PDR. The distance between P_n and P_0 (i.e. the location of node A) is calculated, and together $MaxPDR$ and d are used as input into `LOC_ConsistencyCheck()` to conduct a location-based consistency check.

The function `LOC_ConsistencyCheck()` operates in a manner similar to `SS_ConsistencyCheck()`. During deployment, a table of (PDR, d) values are gathered to represent the profile of normal radio operation for node A. As in `SS_ConsistencyCheck()`, we may define a jammed-region and a benign-region using either a binning procedure or regression to obtain lower bounds on the PDR that should be observed for a given distance under benign radio conditions using measured data. If the point $(MaxPDR, d)$ falls in the jammed-region, then the node declares it is jammed.

```
Algorithm: LOC_Detect_Jam
{PDR(N) : N ∈ Neighbors} = Measure_PDR()
(n, MaxPDR) = (arg max, max){PDR(N) : N ∈ Neighbors}
if MaxPDR < PDRThresh then
    P_0 = (x_0, y_0) = GetMyLoc()
    P_n = (x_n, y_n) = LookUpLoc(n)
    d = dist(P_0,P_n)
    CCheck = LOC_ConsistencyCheck(MaxPDR, d)
    if CCheck == False then
        post NodeIsJammed()
    end
end
```

Algorithm 2: Jamming detection algorithm that checks the consistency of PDR measurements with location information.

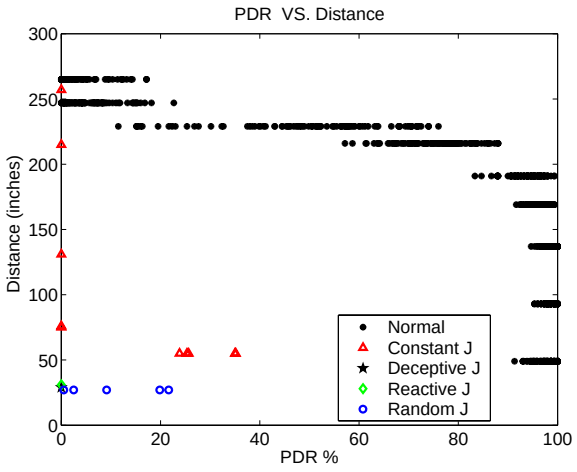


Figure 7: The (PDR, d) measurements, indicating the relationship between PDR and distance between source and receiver. Also presented are the (PDR, d) values measured for the different jammer models.

We note that, just as in the operation of `PDRSS_Detect_Jam`, the assumption that every legitimate node transmits a minimal baseline amount of traffic with which to estimate PDR is paramount to the operation of the `LOC_Detect_Jam` protocol. This baseline amount of traffic may coincide with the transmission of location advertisements in order to reduce the overhead of the protocol. The baseline traffic assumption allows us to declare the PDR to be 0 when no packets are received from a neighbor node within a given time period. This assumption is particularly important for handling scenarios where every neighbor node is jammed, as it allows `LOC_Detect_Jam` to pass into the location-based consistency check, which will allow the algorithm to declare that the node is jammed since its neighbors should have delivered at least a minimal amount of packets. Finally, we note that we have disregarded the extremely unlikely event that all neighboring devices have faulted or depleted their power resources.

We conducted an experiment to validate Algorithm 2. The setup of the experiment was the same as the experiment used to validate Algorithm 1. We gathered (PDR, d) values for normal operation as well as for scenarios involving the different jammers, as depicted in Fig. 7. As can be seen in Fig. 7, the (PDR, d) values for the jammer scenarios, where the source-receiver separation was small, are distinctly separated from normal operation values, and hence fall in the jammed-region. Again, we would like to point out that, for a reasonably dense network where every node has one or more neighbors that are close to itself, a jammer's presence can be easily identified, as shown in Fig. 7. If a node, on the other hand, does not have a nearby neighbor, then the PDR of that node, even without the jammer, is rather poor (Fig. 7). For these nodes, the effect of a jammer will not be noticeable anyway.

We now address the frequency of node position advertisement. There are two factors that affect the frequency: first, nodes may move towards or away from each other, and second, position messages may be missed, especially for neighbors farther away from node A. We may address the first factor by setting a requirement that a node announces its location whenever it has moved a distance δ from its previous position. By using the device's velocity v, we find that a device should update its position at least every $\tau = \delta/v$ seconds. To address the second issue, we assume that each device seeks a guarantee of η that its position announcement will arrive to neighbors who are sufficiently close to have at least a nominal packet delivery ratio of q. Assuming independence of successive transmissions of position announcement messages, the cumulative distribution for the amount of transmissions T before the first successful delivery is

$$F_T(T) = 1 - (1-q)^T, \quad \text{for } T \in \{1, 2, 3, \cdots\} \qquad (1)$$

From the cumulative distribution, we may find the amount of transmissions, $\tilde{T}$, needed to have a guarantee of η that the position announcement will have been heard. Combining the two factors, a node should announce its position every $\tau/\tilde{T}$ seconds. The frequent announcement of position information guarantees that nodes will have knowledge of their neighbor's position.

5. RELATED WORK

Radio interference attacks are a serious threat to the operation of a wireless network, regardless of the type of wireless network. In order to cope with the threat of jamming attacks, it is important to understand the different threat models that may be employed by adversaries, the methods that are needed to diagnose these threats, and the countermeasures that may be employed to defend against jamming attacks.

The traditional literature on jamming primarily focuses on the design of physical layer technologies, such as spread spectrum, that are resistant to RF jamming [28,30]. It should be realized that the physical layer technologies needed to reliably resist jamming have not found widespread deployment in commodity wireless devices, such as wireless LANs and sensor networks. Our work takes the viewpoint that rather than replace existing systems with more complicated radio platforms, it is instead desirable to understand the modes of attacks that may be launched against existing platforms, and be able to detect them. Following detection, appropriate countermeasures may be employed.

The issue of jamming detection was briefly studied by Wood and Stankovic in [32] in the context of sensor networks. This study posed the issue of jamming detection in the loose context of the utility of the communication channel, and presented several factors that might affect the channel's utility. The primary focus of this paper, however, was on the issue of mapping the jammed region and did not explore the fact that no single measurement is a *sufficient* statistic for basing decisions upon. Our work has explored the inconsistencies that might arise from naively employing decision processes built upon these factors. Further, our detection algorithms may be viewed as a complement to their work and, when integrated with their mapping algorithm, can lead to enhanced mapping services.

Although not precisely a jamming attack, one may exploit the MAC layer to achieve increased network resources [4,18]. The issue of detecting non-MAC compliancy was recently studied in [29]. This work showed that a greedy user can increase his share of bandwidth by sightly modifying the driver of his network adapter. The greedy user may try to corrupt the RTS and CTS of other users to prevent packet transmission, or may corrupt ACKs to cause the ACK contention window to increase, leading to larger backoff. They proposed DOMINO, a system for detection of such greedy behavior in the MAC layer of IEEE 802.11 public networks.

Countermeasures for coping with jammed regions in wire-

less networks has been studied in [23, 33]. In [23], the use of low density parity check (LDPC) codes is proposed to cope with jamming. Further, an anti-jamming technique is proposed for 802.11b that involves the use of Reed-Solomon codes. In [33], two countermeasures are presented for coping with jamming. The first method, channel surfing, involves a form of on-demand link-layer frequency hopping, where valid participants change the channel they are communicating on when a denial of service attack occurs. The second method, spatial retreats, involves legitimate network devices moving away from the adversary to reestablish connections.

6. CONCLUSIONS

Wireless networks are being deployed in a variety of forms, ranging from ad hoc networks to wireless LANs to sensor networks. The shared nature of the wireless medium will allow adversaries to pose non-cryptographic security threats by conducting radio interference attacks. Therefore, understanding the nature of jamming attacks is critical to assuring the operation of wireless networks. This paper has sought to focus on both sides of the issue by presenting four different jammer attack models that may be employed against a wireless network, as well as exploring techniques for detecting the presence of a jamming attack. We have studied the effectiveness of our four jammer strategies by constructing prototypes using the MICA2 Mote platform and measuring how each of the jammers fared in terms of their effect on the packet send ratio and packet delivery ratio.

We then studied the issue of detecting the presence of jamming attacks, and examined the ability of different measurement statistics to classify the presence of a jammer. We showed that by using signal strength, carrier sensing time, or the packet delivery ratio individually, one is not able to definitively conclude the presence of a jammer. Therefore, to improve detection, we introduced the notion of consistency checking, where the packet delivery ratio is used to classify a radio link as having poor utility, and then a consistency check is performed to classify whether poor link quality is due to jamming. We introduced two enhanced detection algorithms: one employing signal strength as a consistency check, and one employing location information as a consistency check. We evaluated the effectiveness of each scheme through empirical experiments and showed that each of the four jammer models we introduced can be reliably classified using our consistency checking schemes.

7. REFERENCES

[1] IEEE Std 802.11i/d3.0. Available at http://www.cs.umd.edu/ mhshin/doc/802.11/802.11i-D3.0.pdf.

[2] AusCERT. AA-2004.02 - denial of service vulnerability in IEEE 802.11 wireless devices. http://www.auscert.org.

[3] P. Bahl and V. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of IEEE Infocom 2003*, pages 775–784, 2000.

[4] J. Bellardo and S. Savage. 802.11 denial-of-service attacks: Real vulnerabilities and practical solutions. In *Proceedings of the USENIX Security Symposium*, pages 15–28, 2003.

[5] S. Capkun and J. Hubaux. Secure positioning in sensor networks. Technical report EPFL/IC/200444, May 2004.

[6] Chipcon. Chipcon cc1000 radio's datasheet. http://www.chipcon.com/files/CC1000_Data_Sheet_2_1.pdf.

[7] P. Enge and P. Misra. *Global Positioning System: Signals, Measurements and Performance*. Ganga-Jamuna Pr, 2001.

[8] F. Fitzek and M. Reisslein. MPEG-4 and H.263 video traces for network performance evaluation. *IEEE Network*, 15(6):40–54, November/December 2002.

[9] J. L. Hill and D. E. Culler. Mica: A wireless platform for deeply embedded networks. In *IEEE Micro*, pages 12–24, 2002.

[10] Y. Hu, A. Perrig, and D. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. In *8th ACM International Conference on Mobile Computing and Networking*, pages 12–23, September 2002.

[11] Y. Hu, A. Perrig, and D. Johnson. Packet leashes: a defense against wormhole attacks in wireless networks. In *Proceedings of IEEE Infocom 2003*, pages 1976–1986, 2003.

[12] Q. Huang, H. Kobayashi, and B. Liu. Modeling of distributed denial of service attacks in wireless networks. In *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, volume 1, pages 41–44, 2003.

[13] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: attacks and countermeasures. In *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications*, pages 113–127, 2003.

[14] S. Kay. *Fundamentals of Statistical Signal Processing: Detection Theory*. Prentice Hall, 1998.

[15] B. Kedem. *Time Series Analysis by Higher Order Crossings*. IEEE Press, 1994.

[16] L. Kleinrock. *Queueing Systems, Volume 2: Computer Applications*. John Wiley & Sons, 1976.

[17] L. Kleinrock and F. Tobagi. Packet switching in radio channels: Part i–carrier sense multiple-access modes and their throughput-delay characteristics. *IEEE Trans. on Communications*, 23(12):1400 – 1416, 1975.

[18] P. Kyasanur and N. Vaidya. Detection and handling of mac layer misbehavior in wireless networks. In *Proceedings of the 2003 IEEE International Conference on Dependable Systems and Networks*, pages 173 – 182, 2003.

[19] K. Langendoen and N. Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Comput. Networks*, 43(4):499–518, 2003.

[20] L. Lazos and R. Poovendran. SeRLoc: Secure range-independent localization for wireless sensor networks. In *Proceedings of the 2004 ACM Workshop on Wireless Security*, pages 21–30, 2004.

[21] Z. Li, W. Trappe, Y. Zhang, and B. Nath. Securing wireless localization: Living with bad guys. In *DIMACS Workshop on Mobile and Wireless Security*, 2004.

[22] D. Nicelescu and B. Nath. DV based positioning in ad hoc networks. *Telecommunication Systems*, 22(1-4):267–280, 2003.

[23] G. Noubir and G. Lin. Low-power DoS attacks in data wireless lans and countermeasures. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):29–30, 2003.

[24] P. Papadimittratos and Z. Haas. Secure routing for mobile ad hoc networks. In *SCS Communication Networks and Distributed Systems Modeling and Simulations Conference (CNDS 2002), San Antonio*, 2002.

[25] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 95–107. ACM Press, 2004.

[26] H. V. Poor. *An Introduction to Signal Detection and Estimation*. Springer Verlag, 2nd edition, 1994.

[27] B. Potter. Wireless security's future. *IEEE Security and Privacy Magazine*, 1(4):68–72, 2003.

[28] J. G. Proakis. *Digital Communications*. McGraw-Hill, 4th edition, 2000.

[29] M. Raya, J. Hubaux, and I. Aad. Domino: a system to detect greedy behavior in ieee 802.11 hotspots. In *MobiSYS '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 84–97. ACM Press, 2004.

[30] C. Schleher. *Electronic Warfare in the Information Age*. MArtech House, 1999.

[31] A. Wood and J. Stankovic. Denial of service in sensor networks. *IEEE Computer*, 35(10):54–62, October 2002.

[32] A. Wood, J. Stankovic, and S. Son. JAM: A jammed-area mapping service for sensor networks. In *24th IEEE Real-Time Systems Symposium*, pages 286 – 297, 2003.

[33] W. Xu, T. Wood, W. Trappe, and Y. Zhang. Channel surfing and spatial retreats: defenses against wireless denial of service. In *Proceedings of the 2004 ACM workshop on Wireless security*, pages 80 – 89, 2004.

[34] L. Zhou and Z. Haas. Securing ad hoc networks. *IEEE Network*, 13(6):24–30, 1999.