Capabilities of Low-Power Wireless Jammers

Lifeng Sang
Department of Computer Science & Engineering
The Ohio State University
sangl@cse.ohio-state.edu

Anish Arora
Department of Computer Science & Engineering
The Ohio State University
anish@cse.ohio-state.edu

Abstract

How can a node *J* predictably jam the wireless communications from a node *S* to a node *R*? More specifically, how can *J* choose its location, power level, or jamming pattern so that communications from *S* are not received by *R*? Can *J* jam communications from *S* so that corrupted values are delivered at *R*? Can *J* jam so that the corrupted values are predictable? Conversely, can *R* discriminate an uncorrupted communication from *S* from a corrupted one or an uncorrupted communication from *J*? Can *R* recover the value communicated by *S* from a corrupted value resulting from jamming?

In this paper, motivated by the goal of modeling these fine-grain capabilities of jammers for the context of security in low-power wireless networks, we experimentally characterize jamming in networks of *CC2420* radio motes and *CC1000* radio motes. Our findings include that it is easy to locate *J* (relative to *S* and *R*) and choose its power level so that *J* can corrupt *S*'s messages with high probability as well as corrupt individual *S*'s bits with nontrivial probability. Internal jammers are however limited in at least two ways: One, it is hard for them to prevent *R* from detecting that it has received an uncorrupted message from *S*. And two, the outcome of their corruptions are not only not deterministic, even the probabilities of corrupted outcomes are time-varying. We therefore conclude that it is hard to predict the value resulting from colliding *S*'s messages (bits) with *J*'s messages (bits) and, conversely, to deduce the value sent by *S*'s or *J*'s from the corrupted value received by *R*.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks:**]: Network Protocols; D.4.6 [**Operating Systems:**]: [Security and Protection]

General Terms

Security, Design

Keywords

Wireless Sensor Network, Jamming Model

1 Introduction

Interference can be exploited in both adversarial and cooperative settings to prevent and even hide communications in wireless networks. In this paper, we study the fine-grain abilities of intentional interference (i.e., jamming) in low power wireless networks to successfully prevent and hide communications. Preventing communication has several forms: a sender may be deceived into not sending at all, its communication may be lost in the sense that the receiver(s) may not be able to frame a value from the received signal, its communication may be corrupted in the sense that the framed value may differ from that which was sent, or its communication may be dominated by the jammer in the sense that the framed value may be that which is communicated by the jammer. Likewise, hiding communication has several forms: a sender may communicate so that only the receiver(s) are able to receive the communicated value, the communicated value can be recovered (or deciphered) from the received value, or the presence of the sender's or jammer's signal can remain undetected.

Since jamming that is external to a network can be arbitrarily complex, we limit our attention in this paper to the case of self-jamming, where one or more nodes internal to the network collude to jam communications. The limited case is still of practical interest, not only because the resulting jamming abilities can be inherited by external environments, but also because in-network attacks are becoming increasingly plausible as wireless networks evolve to supporting applications that are launched remotely and may therefore be compromised by remote attacks or applications that are not fully trusted.

The literature on jamming in wireless networks is extensive. While we review this literature in the next section, we recall here that it is known that it is feasible for a jammer to prevent communications in low-power networks. At the same time, using multimodal signatures, such as a combination of statistics of received signal strength, the packet delivery ratio, and the violation of error detection codes, it is feasible for a receiver to detect the jammer [29, 31, 30]. Our experimental results replicate and extend these findings: specifically, we show that it is easy to choose a node *J* and a power level so that communication from a node *S* to a node *R* is predictably prevented in spite of hardware variability, environment, and communication content. Moreover, we show that jamming detection capabilities can be strengthened so that it is hard for *J* to get *R* to falsely accept a corrupted message or a message not sent by *S* as being sent by *S*.

The fine-grain jamming capabilities we study in this paper focus on questions of corruption: How can a node *J* predictably corrupt the wireless communications from a node *S* to a node *R*? Can *J* force the corruption, with or without knowledge of the communications of *S*, so that the resulting value at *R* is predictable? Likewise, can *J* force the corruption so that the value sent by *S* or *J* can be recovered from the resulting value at *R*? Answers to these questions have important roles to play in the integrity, authentication, and confidentiality of communications in low-power wireless networks.

Contributions. Our answers are based on both analysis and experiments on corruption at the level of bits as well as of packets. Our experiments use a simple protocol to implement internal jamming in *CC2420* radio motes and *CC1000* radio motes, and study jamming in the presence of different transmission powers, locations, and communication content. Our main findings are summarized as follows:

1. It is easy for *J* to choose a location and a power level so that it can corrupt a bit or a packet from *S* to *R*. It is hard for *R* to detect corruption at the level of individual bits, but it is easy for *R* to detect corruption at the level of packets. More specifically, it is easy for *R* to authenticate an uncorrupted packet from *S* even in the presence of jamming.
2. The probability of corrupting a bit via jamming is nontrivial and fluctuates dramatically over time. More specifically, the probability of corrupting a bit-value b with a bit-value b' to obtain a bit-value b" is non-trivial and fluctuates dramatically over time. By the same token, the probability of corrupting a packet via jamming is substantial and fluctuates significantly over time.
3. It is hard for *J* to jam so that the corrupted value resulting from colliding with *S*'s bit or packet is predictable. Conversely, even if *J* uses a known protocol and values for jamming, it is hard for *R* to recover the original value sent by *S* from the corrupted value received at *R*. In fact, the probability of successful recovery at *R* is close to the probability of random guessing, even when multiple *R* cooperate.

The rest of the paper is organized as follows. In Section 2, we discuss related work in jamming. In Section 3, we present the system model, the problem statement, and our experimental methodology. We then present the analysis and experiments leading to our findings in Section 4. We discuss the implications of our findings in Section 5 and make concluding remarks in Section 6.

2 Related Work

Communication and information theory. A number of researchers have studied the impact of jamming at the physical layer, asking for instance what the amount of information is that can be communicated in the presence of signal jamming. For example, in [7], the authors obtain bounds on the coding capacity of Gaussian channels, for the case where jamming mismatches the power constraint on the signal is mismatched with respect to channel noise. They also obtain an exact expression for the coding capacity for the case where the noise has a Crammer-Hida representation of finite multiplicity.

Other studies [11, 5] have characterized the effect of different sorts of physical jamming signals (wide-band, partial-band, tone, multi-modal) with respect to the type of modulation technique (direct sequence spread spectrum, frequency hopping, frequency shift keying, hybrid) and demodulation strategy (coherent or non coherent, with or without error correcting codes, with or without side information). These works provide a basis for choosing modulation techniques and spreading codes for different jamming environments. An illustrative result, due to Evaggelos [5], is that multiple noise or tone jammers have no advantage over a single noise or tone jammer that has equivalent spectral density (energy per symbol) of jamming but jams smaller fractions of the band (their particular setup assumes frequency hopped spread-spectrum signaling, frequency shift keying modulation with noncoherent demodulation and Reed-Solomon coding).

Many "intelligent" jamming studies have focused on the design of strategies to optimize payoff in a model/game setting. In [25], for instance, Basar considers the problem of transmitting a sequence of identically distributed independent Gaussian random variables through a Gaussian memoryless channel in the presence of an intelligent jammer. The jammer's optimal strategy is either to choose a linear function of the measurement received through channel-tapping, or to additively choose an independent Gaussian noise sequence (which depends upon the region here the parameters lie). Dually, the strategy of the transmitter is to amplify the input sequence to the given power level by a linear transformation, whereas the optimal policy for the receiver is to use a Bayes estimator. In [15], the authors model jamming as a two-person zero-sum noncooperative dynamic game, and find that when the payoff matrix is lower than a threshold, the optimal steady-state strategies are mixed and the payoff increment is constant over time. When it is greater than the threshold, the strategies are pure, and the payoff increment exhibits oscillatory behavior.

A particularly elegant application of jamming strategy design is to achieve secret communication between cooperating nodes [10, 20, 12]. These constructions complement a substantial literature on the rates of achievable secure communication which date back to the origins of information theory.

Wireless network protocols. Knowledge of protocols can be exploited to jam intelligently at higher layers of the network, with potentially severe effect. Theunte and Acharya [26], for instance, exploit knowledge of crucial timings and control packets to show that the effect of periodic jamming on throughput for an 802.11b network can be similar to that of continuous jamming. Law et al [13] show how packet interarrival times can be exploited to energy-efficiently jam communications with commonly used wireless sensor network MAC protocols (S-MAC, L-MAC, and B-MAC). Wood and Stankovic [29] overview denial of service attacks from physical layer attacks to transport layer attacks.

[29] also presents protocol countermeasures to jamming attacks, including for instance the ability of nodes to reduce their duty cycle upon detecting a jamming attack, services to map jamming regions by exploiting network density and thereby the presence of nodes at the fringe of jamming re-

gions, and rerouting packets around jamming regions. Channel surfing (and power/code-rate control) techniques for resilience to jamming attacks have been studied [18] in 802.11 networks and in wireless sensor networks [30, 31].

Detection of jamming has also received diverse attention. Wood and Stankovic's overview [29] suggests that jamming detection can be based on factors such as inability to access wireless channel, bad framing, CRC failures, address corruptions, protocol violations, excessive RSSI values, low SNR, repeated collisions, etc. Xu et al [30, 31] present a detailed analysis for different sort of jamming traffic; they find that higher order counting statistics may suffice for distinguish normal traffic from some jamming traffics (e.g., constant or deceptive) but not all jamming traffics (e.g., random or reactive); they also find that multimodal detection (e.g. based on signal strength and packet delivery rates) work well for detection. In a sense, the work of Xu et al is closest to our work; on one hand, they consider more diverse jamming strategies than we do, on the other hand, they do not consider the implications of the probability and value of corruption as we do. (For the specific case of RSSI, we note that Srinivasan and Levis [24] conduct evaluations of signal measures for the *CC2420* radios, and find that RSSI measures for a given link has very small variation over time, whereas LQI measures vary over a wider range. Their measurements focus on RSSI and link quality in the absence of concurrent transmission; our study complements their work by measuring RSSI in the presence of jamming.) McCune et al [16] study detection of jamming for the problem of reliably broadcasting a packet from a base station to nodes in a network with an efficient Secure Implicit Sampling technique.

Interference in wireless sensor networks. There has been considerable interest in recent years in modeling interference for low power links. Going beyond the early literature on the behavioral complexity of low-power links [24, 1, 9, 28, 32, 23, 4, 27, 22], recent investigations have frequently considered the effect of concurrent transmissions. Using the SINR model for interference, Son et al [22] study concurrent packet transmissions for mica2 motes. They show that different devices may have different SINR thresholds, and that concurrent transmissions can be successful if the received strength exceeds the receiver threshold. (We note by way of contrast that while they focus on concurrent transmissions intended for different receivers such that all receivers can correctly receive the packets, our focus in jamming is where none of the potential receivers can correctly receive the messages.)

Whitehouse et al [27] also address wireless link quality in the presence of concurrent transmissions. They evaluate a technique to detect and recover messages from packet collisions by exploiting the *capture effect*. They exploit detection of preambles even during packet reception, and allow the packet with the strongest signal strength to be received correctly.

SINR is often used to model packet interference [19] in 802.11 networks, others use geography-based models for instance based on the conflict graph [8]) concept where packets are lost if the range of two concurrent transmitting signals intersect.

Our work. By way of contrast with the work described above, our work addresses the effects of intentional interference in terms of the specific properties of corruption, the ability to predictably corrupted messages, the ability to control the value of corruption, and the ability to recover one or more of the messages from the corrupted result. To the best of our knowledge, these issues have received little consideration thus far. We consider corruption at physical layer (bit level) as well as higher layer (packet level) corruption, and accommodate both intelligent as well as dumb jamming strategies. We limit ourselves to "internal" corruption attacks in low-power single channel wireless sensor networks, but consider diverse radios within this space.

3 System Model & Experimental Methodology

Table 1 contains the notations that we will use in the rest of the paper.

m, m'	a message
m_S	a message sent by S
m_J	a message by J
m_R	a message received at R
$m_S \vert\!\lhd m_J$	corrupted message yielded by jamming m_S with m_J
m^b	a message containing 1 or more b-valued bits, $b \in \{0,1\}$
$m^\dagger$	a message containing a mix of both 0 and 1 bits
$N(m)$	number of messages with same type as that of m
p^b_m	probability of occurrence of b-valued bit in m, $b \in \{0,1\}$

Table 1. Notation

3.1 System Model and Problem Statement

The system consists of a network of resource-constrained wireless nodes, which we refer to as motes. One mote in the system is labeled as sender S, one or more other motes are labeled as receiver R, and the designer may label one node as an internal jammer J.

We assume the following system properties:

- Motes may choose from more than one power level for their communications. Their communication range at any power level need not be isotropic.
- The location and power level of motes S and R are fixed and known to J, whereas the location and power level of J may be chosen (from the available choices) to make its jamming successful.
- The content, format, and traffic pattern of the communication from S to R may be fixed and known to J. The format and traffic pattern may be known to all other motes as well (including R).
- The energy of motes is limited, so efficient jamming strategies are desirable. For instance, jammers should not blindly used highest power level for their communications.
- J may sense the channel and accordingly activate its transmitter, so as to disrupt communications between S and R energy-efficiently.

- J may also synchronize in a fine-grain manner with S and accordingly activate its transmitter to hide S's communications.

Note that the model accommodates the possibly of the internal jammer J being malicious as well as benign and that the communications are single bits or more complex packets.

Our goal is to investigate a model of jamming in low-power wireless networks that provides guidance to the design of both security or attack protocols. Specifically, we will consider the simple protocol:

$$S \rightarrow R: \ m_S \quad || \quad J \rightarrow R: \ m_J$$

where S and J concurrently send message $m_S$and m_J respectively to R. Concurrency may be realized via channel sensing if J works independently of S or more precisely via synchronization if S and J work cooperatively.

Specific problems in the scope of our study of fine-grain jamming capabilities are: Can J predictably jam so as to: lose both m_S and m_J at R? deliver a corrupted value at R? deliver only m_J at R? In particular, can J control the value resulting from corruption of m_S, even if it knows m_S a priori? Conversely, can one or more R detect whether the received value is indeed m_S or not? Moreover, can one or more R recover m_S upon receiving a corrupted value, even if they know the jammer's protocol and choice of m_S?

3.2 Experimental Methodology

To answer these problems as well as to verify our model of corruption, we carried out various experiments on both *CC1000* and *CC2420* motes. We describe our experimental method and setup below.

We used Tmote Sky [17] motes with *CC2420* radio and MICA2 motes with *CC1000* radio. The lowest power level on *CC2420* radio is 3, corresponding to -24 dBm, while the highest power level is 31, corresponding to 0 dBm [21]. While space reasons limit us to presenting more experiments on the *CC2420* motes than on the *CC1000* motes in this paper, we note that in general similar observations hold on both the *CC1000* and *CC2420* motes.

We considered the following aspects in our experiments:

- **Jamming Power:** Varying jamming power may result in different loss and corruption properties: when jamming power is too low, S's signal may dominate, while when jamming power is too high, J's signal may dominate. We varied jamming power to see its impact on corruption.
- **Location:** The difference between $distance(R,S)$ and $distance(R,J)$ plays an important role in jamming, since path loss is typically assumed to follow a log-normal fading [22]. We varied the location of R from S towards J to see the effect of distance, and also considered extremal cases for $distance(R,S)$=0, i.e., when R and S sit side-by-side and one-atop-the-other.
- **Communication Content:** The values of the communication being jammed and the jamming communication itself can affect corruption output. For example, bit zero may produce different output when jammed by bit zero or bit one. We studied jamming using different communication content. we note that we also overwrote the TinyOS message header with the designated value to avoid miscalculation.
- **Synchronization:** For realizing intentional interference, we synchronized J's jamming with S's transmission. Because of hardware variance, we eschewed the use of a hardware timer alone. Instead, we let R to trigger the communication, i.e., both S and J started sending a message immediately after they received a control message from R.

3.2.1 *Implementation Issues*

To achieve accurate jamming, our TinyOS implementation addressed the following issues: (I) In TinyOS, by default a transmission is only enabled when the channel is free (this is checked using the CCA register [21]) as the default CSMA protocol backs off if the channel is busy. In contrast, we needed to allow concurrent transmissions to enable jamming. (II) In TinyOS, by default multiple tasks are posted in between an application layer invoking a message send and the time it is communicated by the radio. Similarly, many tasks are posted between a packet being received and being delivered to the application. Because tasks in TinyOS are a form of deferred procedure call (DPC) [2], which enables a program to defer a computation or operation until a later time, they may introduce unexpected delays and have a negative impact on our jamming accuracy. We therefore modified the networking stack for the *CC2420* radio and *CC1000* radio motes so that : (1) The default CSMA protocol was removed so that concurrent transmissions are possible; (2) Data processing is done in the lowest layer as we can touch. The default TinyOS library was modified so that the *send* command and the *receive* event are executed immediately without involving many explicit tasks.

3.2.2 *Background Noise*

Conceptually, a noisy environment is likely to cause more message corruptions. In order to accurately evaluate the corruption values resulting from jamming, we chose a relatively low noise environment for all the experiments. We used channel 26 in *CC2420* motes to minimize potential interference from co-existing *802.11b* network, as suggested by [23].

We first ran an experiment to estimate background noise checking[1] at 1Hz for about 15 minutes. The results were relatively consistent over time. Figure 1 shows a 40-second subset of the data measured in the room and a histogram of the RSSI values over the whole 15-minute period. 45.88% of the samples have a value of -93 dBm, 18.82% have a value of -94 dBm, 12.59% have a value of -92 dBm, and less than 5% have a value below -94 dBm or above -92 dBm. We regard this background noise as being reasonably low, and expect that it not negatively impact our jamming evaluation.

3.2.3 *Experimental Setup*

For the single receiver case, we chose four topologies in our experiments: *(a)*, *(b)*, *(c)* and *(d)*, as shown in Figure 2. R and S are very close to each other in *(a)* and *(b)*: R is

[1] We thank Hyungjune Lee from Stanford for sharing the noise collection code.

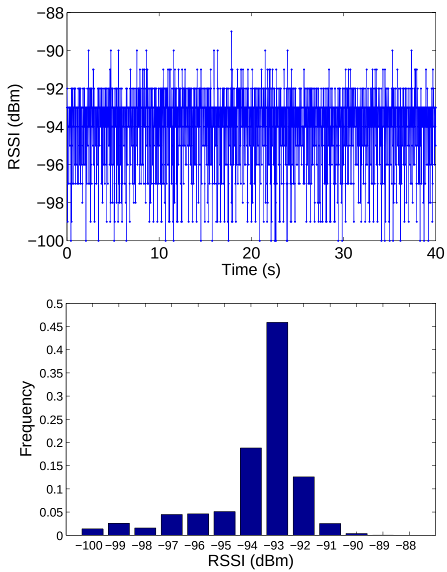
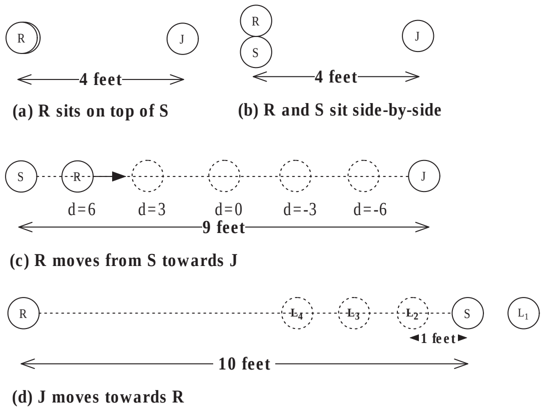


Figure 1. Background noise trace (with 1 Hz sampling frequency) and histogram at channel 26. (a) RSSI (in dBm) measured over the first 40s period of a 15-minute data set; (b) Histogram of RSSI sampled over the 15-minute data set



(a) R sits on top of S

(b) R and S sit side-by-side

(c) R moves from S towards J

(d) J moves towards R

Figure 2. Experimental topologies for single receiver, $d = dist(R,S) - dist(R,J)$

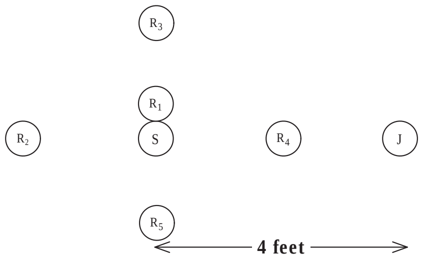


Figure 3. Experimental topology ***I*** **for multiple receivers:** R_1 **and** S **sit side-by-side,** R_2**,** R_3**,** R_4 **and** R_5 **sit around** S **with distance 2 feet, and the distance between** S **and** J **is 4 feet**

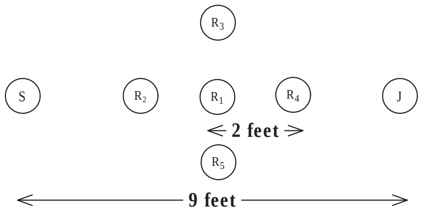


Figure 4. Experimental topology ***II*** **for multiple receivers:** R_2**,** R_3**,** R_4 **and** R_5 **sit around** R_1 **with distance 2 feet, and the distance between** S **and** J **is 9 feet**

on top of S and side-by-side. The reason for this choice is that intuitively it is the hardest scenario for jamming from the perspective of location. In *c*, we moved R from S towards J to see how distance impacts corruption. In *d*, we moved J towards R to see the effects of jamming on not only "inner-band" links (with high packet delivery ratio) but also for "middle-band" links (with modest packet delivery ratio).

For the multiple receiver case, we chose two topologies in our experiments, as shown in Figure 3 and 4. We added 4 more receivers in topologies *(b)* and *(c)* to study the variation in reception among different receivers.

We then carried out the following experiments:

1. **Fixed location, varying jamming power:** In this experiment, we used topology *a*. S sent a message containing all zeros (m^0) at power level 3 (-24 dBm) immediately after receiving R's control message; meanwhile, J sent a message containing all ones (m^1) at various power level from 3 to 31 (0 dBm).
2. **Fixed power, varying** R**'s location:** In this experiment, we used topology *c*. We moved R from S towards J, and collected data at R when $d = \{6, 3, 0, -3, -6\}$ feet. S sent m^0 and J jammed with m^1, both at power level 3 (-24dBm).
3. **Varying the bit value:** In this experiment, we used topology *a* and *b*. S used power level 3, and J used power level 31. We recorded data for different communication content as follows, (I) S: m^0; J: m^1; (II) S: m^1; J: m^0; (III) S: m^0; J: m^0; (IV) S: m^1; J: m^1.
4. **Multiple receivers:** In this experiment, we used topology *I* and *II*, as shown in Figure 3 and 4. In the experiment using topology *I*, S sent packets with all zeros at power level 3, while J jammed the network with all ones at power level 31. In the experiment using topology *II*, S sent packets with all zeros at power level 3, while J jammed the network with all ones at power level 3. We recorded data at R_i ($1 \leq i \leq 5$).
5. **Middle-band links:** All the above experiments are conducted on inner-band links, where the PRR between S and R is more than 99% in the absence of J. In this experiment, we set S and R such that the average PRR is below 80% in the absence of J. We used topology *d*, and put J on L_1, L_2, L_3, and L_4 to see the loss and corruption difference when J disrupts the communication at power level 3.

In each experiment, R triggered communication once every 300 milliseconds, for a total of 30 minutes. The packet size was 64 bytes in all experiments. Our results are presented in the following section.

4 Fine-grain Jamming

We begin by addressing the question of how to make jamming successful, in other words, *what is a procedure for choosing a location and power level for J so as to predictably jam the communication from S to R.*

4.1 Designing Predictable Jamming

In the traditional graph-based interference model, when two concurrent transmissions collide, none of the packets are received. However, researchers have found this is not what results in real scenarios [22]. A better theoretical model is that of *Signal to Interference plus Noise Ratio* (SINR). In the SINR model (also known as physical model [6]), whether a transmission is successful depends on the received signal strength, the interference caused by simultaneous transmissions from other nodes, and the ambient noise level. That is, a transmission can only be successful if and only if

$$\frac{P_R}{P_n + I_R} \geq \beta \qquad (1)$$

where P_R is the received power of a signal sent to the receiver, P_n is the noise power level, I_R is the interference power generated by other concurrent transmissions, and β is the minimum signal-to-interference-ratio that is required for a message to be delivered at the receiver.

In wireless networks, received power decays (roughly) exponentially with the distance of $dist(S,R)$. It is modeled theoretically as:

$$P_R = \frac{P_S}{dist(S,R)^{\alpha}} \qquad (2)$$

where P_S is the sending power of sender S, and α is the path-loss constant, a value typically between 2 and 6. Replacing P_R in Equation 1 with Equation 2, we have:

$$\frac{P_S}{dist(S,R)^{\alpha}(P_n + I_R)} \geq \beta \qquad (3)$$

Therefore, R is unable to correctly receive the message from S if and only if

$$I_R > \frac{P_S}{\beta \times dist(S,R)^{\alpha}} - P_n \qquad (4)$$

In the presence of jamming, the minimum interference power is the received power at R from the jammer J, hence successful jamming needs:

$$\frac{P_J}{dist(J,R)^{\alpha}} > \frac{P_S}{\beta \times dist(S,R)^{\alpha}} - P_n \qquad (5)$$

Jammer Design Procedure: The first step in the procedure is to estimate the unknown parameters in Equation 5. Assuming that the parameters are not known a priori, they are evaluated as follows:

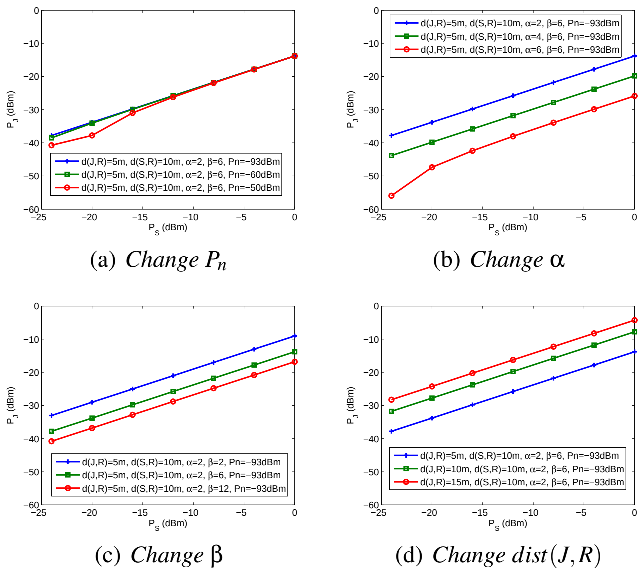
1. P_n: The background noise, P_n, is estimated at different nodes by sampling RSSI values at nodes available to the jammer in the absence of communication, as we did in Section 3.2.2. For example, the background noise (see Figure 1) in our experiments is estimated to be around -93 *dBm*.
2. α: The path loss constant is estimated given the transmission power, the distance between a sender and a receiver that are available to the jamming designer, and the RSSI of received packets at the receiver. We note that α typically ranges between 2 and 6, the ranges for different sorts of environments are well studied in the literature, and even significant errors in this step of estimation are usually acceptable.
3. β: The jamming designer can similarly estimate β based on the statistics of packet reception rate (PRR). As [22] pointed out, different nodes have different reception sensitivity, so it is better to estimate β based on data from multiple nodes if a priori information about the range of hardware variation is not available.
4. P_S: If J and S cooperate, P_S is known to J, otherwise, J may eavesdrop S's transmissions first, and then make an estimation based on sampled *RSSI*, distance between S and J, and the previously estimated α.
5. P_J: This is a control parameter for the jammer designer and depends on the location of the chosen J.
6. $dist(R,S)$ and $dist(J,S)$: These estimates are likely the most important ones for the procedure. The jammer designer might get these estimates from a localization service, cooperation with S, or application specific tools.

Figure 5 illustrates the minimum P_J required to predictably for different values of P_S for sample parameter values of P_n, α, β, $dist(R,S)$ and $dist(J,S)$. We see that the impact of P_n at -93 *dBm* is quite low, so that the relationship between P_S and P_J is approximately linear. In Figure 5(a), its impact is still low in these settings even when P_n increases to -60 *dBm*. The impact of α and β is relatively large when their values span a relatively large interval in Figure 5(b) and 5(c). However, in a fixed environment with fixed devices, α and β usually only vary in a small region, which makes conservative estimation of P_J easier. Among these parameters, $dist(R,S)$ and $dist(J,S)$ might be the most important factors to determine P_J given P_S in a general scenario where $dist(R,S)$ and $dist(J,S)$ may vary.

The second step in the jammer designer procedure is, given knowledge or estimates of these parameters, to choose the best mote location and power level combination for its specific jamming goals. Note that choosing substantially higher than necessary values of P_J will imply that m_J dominates m_S. In this case, S's signal will be ignored as random noise at R and m_J will be delivered. Choosing only slightly higher than necessary values of P_J will imply that neither m_S nor m_J will be delivered; that is, no message or only corrupted messages will be delivered at R. In other words, jammer designers may prefer choosing P_J values from an interval for which it is not the case where only m_S or only m_J are correctly delivered. Similar to Equation 5, for J's message to not be received at R, we have

$$\frac{P_S}{dist(S,R)^{\alpha}} > \frac{P_J}{\beta \times dist(J,R)^{\alpha}} - P_n \qquad (6)$$

When Equation 5 and 6 are both satisfied, $m_R \neq m_S$ and $m_R \neq m_J$. Figure 6 shows an example of three regions, where m_J is received if P_J is above the dash line, while m_S is re-



(a) *Change* P_n

(b) *Change* α

(c) *Change* β

(d) *Change* $dist(J,R)$

Figure 5. Examples of required P_J for successful jamming.

ceived if P_J is below the solid line. Since our model is sensitive to the energy expended by the internal jammer, the jammer designer should choose the lowest P_J such that it can meet its specific jamming goal. Therefore, understanding the outcomes of jamming in the jamming area —the middle region in Figure 6— is of importance to the jamming designer.

To the best of our knowledge, however, the structure of the jamming region has not been explored systematically. Researchers usually do not distinguish between loss and corruption (and sometimes even the receipt of packets from J) during jamming. We explore the distinctions between these various cases from the perspective of J, as well as the perspective of R, in the following subsections.

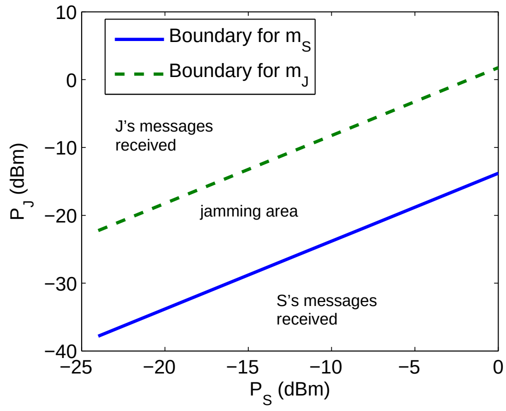


Figure 6. An example of jamming area where $dist(S,R) = 10m$, $dist(J,R) = 5m$, $\alpha = 2$, and $\beta = 2$

4.2 Predicting Loss versus Corruption

Logically, the outcome at R of J operating in the jamming area include: (i) communication loss; (ii) corrupted communications; or (iii) accidentally receiving uncorrupted communications of both m_S and m_J. Not surprisingly, our experiments confirm that the probability of case (iii) is very low.

What is surprising then is that *if the link between S and R is in the inner-band*, i.e., distance(S,R) is small enough that the SNR is significantly above the threshold for reliably receiving packets, *then in the presence of jamming the PDR does not decrease as a result of increased losses, but the number of corrupted packets delivered increases*. In other words, the probability of case (i) is also low and, as a result, the probability of case (ii) dominates. Specifically, in all our experiments on inner-band links, we observe rare (below 0.25%) packet loss; a likely explanation of this phenomenon is that packet preambles are not corrupted with the mote radio families we have considered in our experimental setup and so either a corrupted or an uncorrupted packet is delivered.

	loss	corruption
J is off	23.62%	0.39%
J on L_1	29.59%	0.48%
J on L_2	60.47%	14.46%
J on L_3	40.30%	56.39%
J on L_4	9.4%	81.23%

Table 2. Packet loss and corruption on a middle-band link when J moves towards R. L_1, L_2, L_3 and L_4 are the location spots in Figure 2 *(d)*

If the link between S and R is in the middle-band, i.e., SNR is in the transition region and the PDR is modest, *then the loss rate increases with the presence of moderate jamming. As the jamming level increase*, say because J moves closer to R, *the loss rate becomes progressively less and corruption rate increases*, likely as a result of J's preamble being more reliably received by R. Specifically, in Experiment 5, S and R are chosen to be 10 feet away, since our empirical studies showed that the reliable range of TelosB motes with an internal antenna at lowest power level 3 is around 8 feet in our experimentation environment. The results of moving J towards R, are shown in Table 2. We see that after J starts jamming, the loss rate increases from 24% to 30%. When J moves towards R, both loss and corruption increases. However, when J is much closer to R (starting at L_3), loss rate decreases and corruption rate increases. When J reaches L_4, its packets start dominating: loss keeps decreasing, and more of J's packets received at R. A likely explanation for the increased loss is that when the SNR for both S and J are close to the threshold for reliably receiving packets, even slight changes caused by jamming may result in a loss.

The nontrivial likelihood of corruption brings us to the question of what sort of corruption is likely? Is there an abstract model to describe the output of jamming m_S with m_j, namely $m_S|\triangleleft m_J$? We consider these questions in the next subsection, looking first at the case of bit level corruption and then at the case of packet level corruption.

4.3 Corruption Outcomes

4.3.1 Bit Level Corruption Outcomes

Our probabilistic bit level jamming model for a single receiver, cf. Table 3, postulates that there is a “probability of corrupting a bit in the presence of jamming”, $Pr(b \neq v)$, and a “probability of detecting corruption”, $Pr(E|b \neq v)$.

Source bit	Received bit	corruption	detection
b	v	$Pr(b \neq v)$	$Pr(E\|b \neq v)$

Table 3. Bit level jamming model for corruption and detection. b is the bit sent from S, v is the bit received at R, E is the jamming event

$Pr(b \neq v)$ can be obtained using the finer-grain probabilistic jamming model in Table 4, if both the distribution of source value and jamming value are known. (We will shortly show that the probability of bit level corruption, $Pr(b \neq v)$, is fairly high in the presence of jamming. We will also show, in Section 4.4, that it is hard for a single receiver and for multiple receivers to detect if the bit corruptions are caused by jamming.) In this model, the probability of corruption also depends upon the jamming value. For example, $p^0_{1|\lhd 0}$ is the probability of corrupting 1 to 0 with jamming value 0, and $p^0_{1|\lhd 1}$ is the probability of corrupting 1 to 0 with jamming value 1.

Source bit	Jamming bit	Received bit 0
0	0	$p^0_{0\|\lhd 0}$
0	1	$p^0_{0\|\lhd 1}$
1	0	$p^0_{1\|\lhd 0}$
1	1	$p^0_{1\|\lhd 1}$

Table 4. Bit level jamming model with different jamming value

We list a few special cases of the finer-grain model:

- If $p^0_{0|\lhd 0} = p^0_{0|\lhd 1}$ and $p^0_{1|\lhd 0} = p^0_{1|\lhd 1}$, then the outcome of corruption does not depend upon jamming value.
- If $p^0_{1|\lhd 0} + p^0_{1|\lhd 1} = p^1_{0|\lhd 0} + p^1_{0|\lhd 1}$, this model is similar to the Binary Symmetric Channel (BSC), a common communications channel model used in coding theory and information theory. In BSC, a transmitter wishes to send a bit (a zero or a one), and the receiver receives a bit. It is assumed that the bit is usually transmitted correctly, but that it will be “flipped” with a small probability (the “crossover probability”). In our bit level model, they are typically not equal, and the crossover probability is non-trivial.
- If $p^0_{0|\lhd 0} = 0$, $p^0_{0|\lhd 1} = 0$, $p^0_{1|\lhd 0} = 1$, and $p^0_{1|\lhd 1} = 1$, this becomes a “flipping” model where every bit in the source message is flipped by jamming.
- If $p^0_{0|\lhd 0} = 1$, $p^0_{0|\lhd 1} = 1$, $p^0_{1|\lhd 0} = 1$, and $p^0_{1|\lhd 1} = 0$, this is “AND” channel, the outcome is the “AND” of source bit and jamming bit.
- If $p^0_{0|\lhd 0} = 1$, $p^0_{0|\lhd 1} = 0$, $p^0_{1|\lhd 0} = 0$, and $p^0_{1|\lhd 1} = 1$, this is “XOR” channel, the outcome is the “XOR” of source bit and jamming bit. As an example of one-time pad, XOR channel achieves perfect secrecy for source messages if the jamming bits are truly random [3].
- More generally, if $p^0_{0|\lhd 0} + p^0_{0|\lhd 1} = 1$ and $p^0_{1|\lhd 0} + p^0_{1|\lhd 1} = 1$, then this jamming channel again achieves perfect secrecy since the probability of bit 0 in the outcome is always 50% when m_J is truly random.

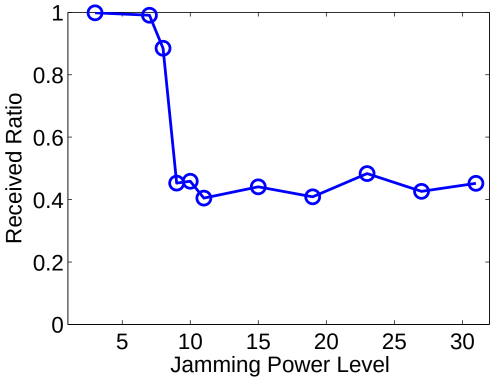


Figure 7. Bit level corruption when S sends 0s and J sends 1s, $p^0_{m^0|\lhd m^1}$, at different jamming power levels (Observation 2)

Bit Level Corruption Results. Figure 7 shows the bit level corruption statistics collected in Experiment 1. The probability of $p(b_0|(m^0, m^1)$ remains relatively stable in the presence of jamming. We note that henceforth in this section the message m in p^b_m has length one. If corruption is low (e.g., jamming power level is less than 9), $p(b_0|(m^0, m^1)$ is certainly dominated by uncorrupted m^0. However, when more messages are corrupted, $p(b_0|(m^0, m^1)$ increases and stays within a certain range, [0.4, 0.6] in this data set. This shows that there is a non-trivial probability of corrupting any bit via jamming.

	$p^0_{m^\dagger}$	$p^0_{m^0\|\lhd m^1}$
$d = 6$	37.64%	55.96%
$d = 3$	48.48%	51.86%
$d = 0$	49.47%	50.85%
$d = -3$	45.42%	37.69%
$d = -6$	55.79%	46.56%

Table 5. Bit level corruption when R moves from S to J, $d = dist(R,S) - dist(R,J)$, (Observation 2)

Table 5 shows the bit level corruption statistics collected in Experiment 2. Again, we see that the probability of corrupting a bit is non-trivial. Specifically, most of $p^0_{m^0|\lhd m^1}$ are around 50%.

Table 6 shows the bit level corruption statistics in Experiment 3. Again, $p^0_{m^0|\lhd m^0}$ is non-trivial regardless of changing bit values that S and J send.

In addition to experiments on *CC2420* motes, we also experimented on MICA2 motes (with *CC1000* radio). Since the results were consistent with those observed on *CC2420* motes, we present only one set of our results here. In these

	side-by-side	one on the other
$p^0_{m^0 \lvert\lhd m^1}$	45.05%	46.95%
$p^0_{m^1 \lvert\lhd m^0}$	42.81%	40.37%
$p^0_{m^0 \lvert\lhd m^0}$	15.63%	15.58%
$p^0_{m^1 \lvert\lhd m^1}$	52.55%	53.68%

Table 6. Bit level corruption where S and J vary bit values, showing non-trivial probability of bit corruption of Observation 2

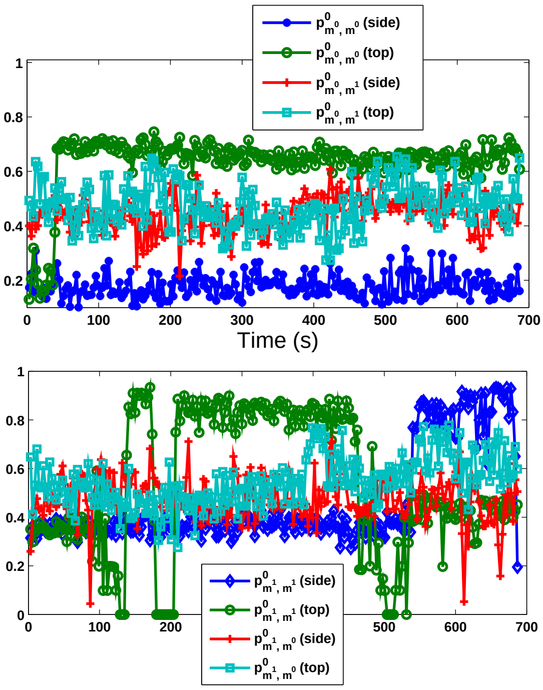


Figure 8. Bit level corruption changes over time (Observation 2)

experiments, both S and J are located around 4 feet away from R, and both use power level 3. The bit corruption results are shown in Table 7. Again, we see that the probability of corrupting a bit by jamming is fairly high, at least 37% in this data set. Not only is the probability of corrupting a bit non-trivial, we find that this probability exhibits significant temporal variation, as shown in Figure 8. Most cases have around 15% (absolute) variation, while some change dramatically over time. For example, $p(b_0|(m^0, m^0))_{top}$ is around 20% in the first 50-second period, but it jumps to 70% and stays around there after that. $p(b_0|(m^1, m^1))_{top}$ has even larger fluctuation. It varies between 0% and 90%. Such a dramatic fluctuation makes predicting $p^0_{m^1|\lhd m^1}$ even harder.

$p^0_{m^0 \lvert\lhd m^1}$	37.74%
$p^0_{m^1 \lvert\lhd m^0}$	65.68%
$p^0_{m^0 \lvert\lhd m^0}$	65.27%
$p^0_{m^1 \lvert\lhd m^1}$	37.77%

Table 7. Bit level corruption using *CC1000* motes (Observation 2)

In summary, we conclude that hat there is a non-trivial probability of corrupting a bit via jamming, which holds across different platforms. Not only the probability is non-trivial, it also changes over time, sometimes even dramatically.

4.3.2 Packet Level Corruption Outcomes

Our packet level jamming model, cf. Table 8, postulates that there is a "probability of corrupting a packet in the presence of jamming", $Pr(m_R \notin \{m_S, m_J\}|E)$, and a "probability of detecting corruption", $Pr(E|m_R \notin \{m_S, m_J\})$.

S	R	corruption	detection
m_S	m_R	$Pr(m_R \notin \{m_S, m_J\} \lvert E)$	$Pr(E \lvert m_R \notin \{m_S, m_J\})$

Table 8. Packet level jamming model. m_R is the message $m_S|\lhd m_J$ received at R, E is the jamming event, is the probability of a corrupted value received at R in the presence of jamming

Each packet received at R can be one of m^0, m^1 or $m^\dagger$. Therefore, the ratio of message m^w, denoted by $r(N_{m^w})$, is simply

$$r(N_{m^w}) = \frac{N_{m^w}}{N_{m^0} + N_{m^1} + N_{m^\dagger}}, w = 0, 1, \dagger \qquad (7)$$

The probability of bit i is then calculated as

$$p^{b_i}_{m^j|\lhd m^k} = r(N_{m^i}) + r(N_{m^\dagger}) \times p^{b_i}_{m^\dagger}, i, j, k = 0, 1 \qquad (8)$$

where $b_i \in \{0, 1\}$. Equation 7 quantifies packet level corruption, while Equation 8 quantifies bit level corruption.

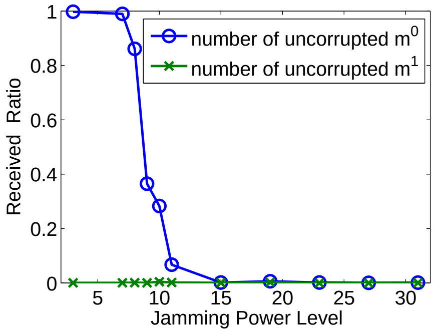


Figure 9. Packet level corruption at different jamming power level, (Observation 1)

Packet Level Corruption Results. Figure 9 lists the packet level corruption statistics in Experiment 1. We see that corruption is rare when jamming power level is less than 8. More than 98% of the messages (m^0) coming from S are received at R. The possibility of receiving m_0 decreases as jamming power increases. For example, 86.10% of the messages are received when jamming power is 8, while only 36.53% of the messages are remained uncorrupted when jamming power is 9. Almost all the packets are corrupted when jamming power level is equal to or greater than 15. These results are consistent with the standard SINR model [22]. If we exchange the role of S and J, we see that almost all m^1 are jammed successfully by m^0. The probability of receiving m^1 at R is extremely low (close to 0) in all cases. These results indicate that there is a high probability that packet corruption occurs if J selects sufficient jamming power.

	uncorrupted m^0	uncorrupted m^1
$d = 6$	29.41%	0.04%
$d = 3$	6.56%	0.00%
$d = 0$	2.91%	0.21%
$d = -3$	0.09%	17.12%
$d = -6$	0.11%	16.63%

Table 9. Packet corruption when R moves from S to J, $d = dist(R,S) - dist(R,J)$, (Observation 1)

Table 9 lists the packet level corruption statistics in Experiment 2. Basically we see that R favors messages from S (respectively J) only when it is close to S (respectively J). For example, 29.41% of the messages were m^0, while only 0.04% were m^1, when $d = 6$. However, m^1 did not increase by much when d changes from -3 to -6. This might be due to hardware variance and multi-path effect. In any case, corruption is achieved with high probability when J selects a reasonable location.

	side-by-side	R on S
$p^{m^0}_{m^0 \| \lhd m^1}$	0.13%	0.85%
$p^{m^1}_{m^0 \| \lhd m^1}$	0.04%	0.13%
$p^{m^0}_{m^1 \| \lhd m^0}$	4.28%	15.19%
$p^{m^1}_{m^1 \| \lhd m^0}$	3.4%	0.47%
$p^{m^0}_{m^0 \| \lhd m^0}$	2.44%	47.47%
$p^{m^1}_{m^0 \| \lhd m^0}$	0%	0%
$p^{m^0}_{m^1 \| \lhd m^1}$	0%	0%
$p^{m^1}_{m^1 \| \lhd m^1}$	40.57%	37.45%

Table 10. Packet corruption when S and J varied values, showing high probability of packet corruption of Observation 1

Table 10 lists the packet level corruption statistics in Experiment 3 where distance(S,R)=0. Again, we see that most packets are corrupted regardless of packet values that S and J sent. For example, only less than 1% of the messages remain uncorrupted in the scenario where m^0 was jammed by m^1. (Interestingly, when a packet is jammed with an identical valued packet, the probability of receiving that packet is relatively high, except $p^{m^0}_{m^0|\lhd m^0}$ when R and S sit side-by-side.)

In summary, we conclude that packet level corruption is achieved with high probability if a jammer chooses a reasonable location and uses sufficient jamming power, per our jammer design procedure.

4.4 Corruption and Jamming Detection

Packet corruption is efficiently detected using error coding, a fact which is corroborated by our experimental results where the packet level CRC error code was different from the CRC computed for the corrupted packet in more than 99% of the corrupted packets produced.

Estimating jamming induced corruption has the error, however, that some corruption may be due to reasons other than jamming, namely unintentional interference and noise. According to Bayes rule,

$$\begin{aligned} Pr(m_R \notin \{m_S, m_J\}) &= Pr(m_R \notin \{m_S, m_J\}|E)Pr(E) \\ &+ Pr(m_R \notin \{m_S, m_J\}|I)Pr(I) \quad (9) \end{aligned}$$

where I is the occurrence of all other unintentional interference and noise).

THEOREM 4.1. *The error of detecting jamming induced corruption is upper bounded by* $\frac{Pr(I)}{Pr(m_R \notin \{m_S, m_J\})}$.

PROOF. From Equation 9, we have $Pr(m_R \notin \{m_S, m_J\}|E)Pr(E) = Pr(m_R \notin \{m_S, m_J\}) - Pr(m_R \notin \{m_S, m_J\}|I)Pr(I) > Pr(m_R \notin \{m_S, m_J\}) - Pr(I)$. By the conditional probability rule, $Pr(E|m_R \notin \{m_S, m_J\})Pr(m_R \notin \{m_S, m_J\}) = Pr(m_R \notin \{m_S, m_J\}|E)Pr(E)$, therefore, $Pr(E|m_R \notin \{m_S, m_J\})Pr(m_R \notin \{m_S, m_J\}) > Pr(m_R \notin \{m_S, m_J\}) - Pr(I)$, and then $Pr(E|m_R \notin \{m_S, m_J\}) > 1 - \frac{Pr(I)}{Pr(m_R \notin \{m_S, m_J\})} > 1 - \frac{Pr(I)}{Pr(m_R \notin \{m_S, m_J\})}$. Since the error of detecting jamming is $1 - Pr(E|m_R \notin \{m_S, m_J\})$, it is upper bounded by $\frac{Pr(I)}{Pr(m_R \notin \{m_S, m_J\})}$. □

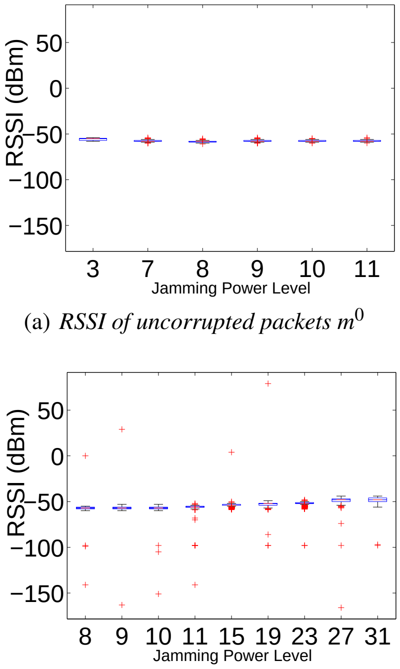
From Theorem 4.1 we may conclude that the error of detecting jamming induced corruption by simply detecting corrupted packets is low if we use a MAC protocol that prevents interference successfully and thus has low $Pr(I)$ –i.e., CSMA MACs with relatively light traffic— and if $Pr(m_R \notin \{m_S, m_J\})$ is non-trivial in any period with jamming events.

4.4.1 Jamming Detection Using Physical Signatures

Corruption detection omits the jamming cases where packets of S and J are delivered. In these cases, we also care about (a) not falsely detecting a valid packet of S as being a jammed packet and (b) accurately detecting jamming if only packets of J are delivered. We therefore consider detection based on physical characteristics. There are several other motivations for exploring jamming detection using physical characteristics: (1) error codes are often excluded for short packets, (2) intelligent external jammers may be able to corrupt so that that error code does not detect the corruption, and (3) there are advantages to doing bit level (or byte level) corruption detection, where again no error code can be used. In this subsection we present a method that uses physical signatures of radios to robustly address these two goals.

We see in the experiments above that the probability of bit level corruption, $Pr(b \neq v)$, is fairly high in the presence of jamming. However, it is not only hard for a single receiver to detect bit level corruption, but also hard for a network to detect bit level corruption in the presence of jamming. As a consequence, it is hard to detect the presence of jamming based on bit corruption. We therefore explore physical signature to see whether it is feasible to detect jamming solely based on physical signature without the knowledge of error code.

We measured RSSI for both corrupted and uncorrupted packets to see whether it is possible to detect collision via simple mechanisms such as RSSI comparison.



(a) *RSSI of uncorrupted packets m^0*

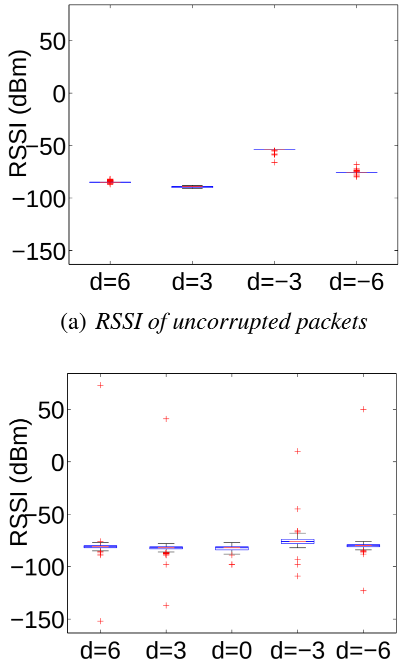
(b) *RSSI of corrupted packets*

Figure 10. RSSI at R when S sends 0s and J sends 1s in Experiment 1, (Observation 3)

Figure 10 shows the boxplot of measured RSSI for uncorrupted packets m^0 and corrupted packets $m^\dagger$ collected in Experiment 1. Surprisingly, *RSSI for messages m^0 that were successfully delivered is relatively stable*, varying within a rather tight range [-3,3] dBm, even when the jamming power is varied. By way of contrast, *RSSI for corrupted messages has substantial more variation and outliers*. This is further verified by results in Experiment 2.

Figure 11 shows the boxplot of measured RSSI in Experiment 2. Note that we used RSSI of m^0 when $d > 0$, and RSSI of m^1 when $d < 0$ in Figure 11(a) for more samples. We see that although RSSI for uncorrupted messages varies a little across different location, which is reasonable since both $dist(R,S)$ and $dist(R,J)$ changed, the RSSI range for each fixed link is very narrow (modulo rare outliers), indicating that RSSI of uncorrupted packets is surprisingly stable. In contrast, RSSI for corrupted messages has wider range of variation, and has more outliers. Some may be still within the range of RSSI of uncorrupted packets. Note that we do expect a quite different RSSI observation when J jams the channel using more powerful devices.

These experiments address goal (a): every packet of S that is accepted by R must have an RSSI estimate that falls within the tight band that is characteristic of S. For goal (b), we refer the reader to another work of ours (reference omitted) where, by virtue of hardware variability and location, richer physical signatures of nodes can be defined that are unique, stable, and easily learned; in other words, it is hard for another node (and even and intelligent internal jammer) to fool R into accepting an uncorrupted message as being from S by somehow emulating the richer physical signature of S (the signature validation procedure in this case includes neighboring receiver nodes other than R to help corroborate the physical signature of S).



(a) *RSSI of uncorrupted packets*

(b) *RSSI of corrupted packets*

Figure 11. RSSI of uncorrupted/corrupted packets in Experiment 2. RSSI of uncorrupted packets at $d > 0$ came from m^0 (S) , and RSSI of uncorrupted packets at $d < 0$ came from m^1 (J). $d = (dist(R,J) - dist(R,S))$, (Observation 3)

In summary, we conclude that techniques based on the combination of CRC based corruption detection, RSSI-variation based corruption detection, and richer physical signature based source authentication suffice to detect many cases of jamming. This conclusion is consistent with that of Xu et al [31], albeit our RSSI variation based corruption detection technique is different from their approach based on higher order count statistics. Note that the RSSI technique can be used for statistical detection of corruption at the byte level and the bit level as well, assuming that RSSI can be sampled at byte level and bit level (which is possible in some but not all radios).

4.5 Corruption Prediction and Recovery

Given the bit level model in Section 4.3.1 and our experimental results, we now prove several negative results regarding the existence of (deterministic) functions that predict the outcome of corruption or that recover uncorrupted values from corrupted packets; these results hold even if S and J cooperate on predicting jamming outcomes, and J and R cooperate on the recovering from corrupted outcomes. As the critical reader might well argue that probabilistic methods would suffice in lieu of deterministic methods in practice, we also prove that there is no probabilistic method that effectively predicts or recovers.

In what follows, let f be an arbitrary function that maps from packets to packets.

THEOREM 4.2. *There is no function f such that $(\forall m_S\ \exists m_J : m_S|\lhd m_J = f(m_S))$, if $(\exists i,k : i,k \in \{0,1\} : 0 < p^0_{0|\lhd i} < 1 \ \wedge\ 0 < p^0_{1|\lhd k} < 1)$.*

PROOF. The probability of bit (wlog let's say) 0 in $m_S|\lhd m_J$ is calculated by $p^0_{m_S|\lhd m_J} = \sum_{i=0}^{1}\sum_{k=0}^{1}(p^i_{m_S} \times p^k_{m_J} \times p^0_{i|\lhd k})$, where $p^0_{i|\lhd k}$ is the probability of bit 0 in the outcome in the jamming model. Let m'_S be a particular instance of m_S where on every bit b_S in m'_S, either $p^0_{b_S} = 1, p^1_{b_S} = 0$ or $p^0_{b_S} = 0, p^1_{b_S} = 1$. Because $0 < p^0_{0|\lhd i} < 1$ and $0 < p^0_{1|\lhd k} < 1$, no matter what m_J is, it is easy to see that $0 < p^0_{m'_S|\lhd m_J} < 1$. This implies that there is no function f that determines $m_S|\lhd m_J$, even if m_S is known to J. □

A weaker result follows trivially if J does not know m_S and randomly chooses the value m_J to jam with.

COROLLARY 4.3. *There is no function f such that $(\forall m_S, m_J :\ m_S|\lhd m_J = f(m_S))$, if $(\exists i,k : i,k \in \{0,1\} : 0 < p^0_{0|\lhd i} < 1 \ \wedge\ 0 < p^0_{1|\lhd k} < 1)$.*

The results imply that there is no deterministic way of predicting the outcomes for all m_S even if S and J cooperate in the sense that J knows m_S a priori. One can also prove that there is no deterministic way of predicting the outcome for any particular value of m_S which works regardless of how J chooses m_J. Our various experiments confirm that corruption outcome can be arbitrary even when the setup, including motes locations, transmission powers and message contents, is the same.

THEOREM 4.4. *There is no function f such that $(\forall m_S\ \exists m_J$: $m_S = f(m_S|\lhd m_J)$ if $(\exists i,k : i,k \in \{0,1\} : 0 < p^0_{0|\lhd i} < 1 \ \wedge\ 0 < p^0_{1|\lhd k} < 1)$.*

PROOF. Assume f s.t. $\forall\, m_S$, $m_S = f(m_S|\lhd m_J)$, then given an instance of $m'_S|\lhd m_J$, f should be able to determine that $m_S = m'_S$, not something else. Since $0 < p^0_{0|\lhd i} < 1$ and $0 < p^0_{1|\lhd k} < 1$ for $0 \le i,k \le 1$, for any particular m'_J, we have $0 < p^0_{m'_S|\lhd m'_J} < 1$. Similarly, there is another m''_S $(m''_S \neq m'_S)$ which satisfies $0 < p^0_{m''_S|\lhd m'_J} < 1$. This implies that given an instance of $C = m'_S|\lhd m'_J$, there is a non-zero probability for m''_S such that $m''_S|\lhd m'_J = C$. Therefore, for the same instance C received at R, the message from S could be m' or m'', hence there is no function f such that $f(C) = m'$. □

A weaker result follows trivially if J does not share m_J with R and hence R assumes the choice of m_J is random.

COROLLARY 4.5. *There is no function f such that $(\forall m_S, m_J$: $m_S = f(m_S|\lhd m_J)$ if $(\exists i,k : i,k \in \{0,1\} : 0 < p^0_{0|\lhd i} < 1 \ \wedge\ 0 < p^0_{1|\lhd k} < 1)$.*

The results imply that there is no deterministic way of recovering all m_S from $m_S|\lhd m_J$ even if R and J cooperate in the sense that R knows m_J a priori. One can also prove that there is no deterministic way of recovering any particular value of m_S which works regardless of how J chooses m_J.

Effective Probabilistic Methods. An effective probabilistic method would be one that predicts outcomes or recovers messages with a probability much better than random guessing. If we assume that J does not collaborate with S or R, we have

LEMMA 4.6. *There is no effective probabilistic method that $\forall m_S, m_J$ predicts $m_S|\lhd m_J$ or recovers m_S from $m_S|\lhd m_J$, if $p^0_{0|\lhd 0} + p^0_{0|\lhd 1} = 1 \ \wedge\ p^0_{1|\lhd 0} + p^0_{1|\lhd 1} = 1$.*

PROOF. The probability of bit (wlog, let's say) 0 in $m_S|\lhd m_J$ is calculated by $p^0_{m_S|\lhd m_J} = \sum_{i=0}^{1}\sum_{k=0}^{1}(p^i_{m_S} \times p^k_{m_J} \times p^0_{i|\lhd k})$, where $p^0_{i|\lhd k}$ is the probability of bit 0 in the jamming outcome. Since J is non-cooperative the method has to work for a random choice of m_J, therefore $p^0_{m_S|\lhd m_J} = \frac{1}{2}\sum_{i=0}^{1}\sum_{k=0}^{1}(p^i_{m_S} \times p^0_{i|\lhd k})$. If $p^0_{0|\lhd 0} + p^0_{0|\lhd 1} = 1$ and $p^0_{1|\lhd 0} + p^0_{1|\lhd 1} = 1$, then $\forall\, m_S, m_J$, $p^0_{m_S|\lhd m_J} = \frac{1}{2}$, and $p^1_{m_S|\lhd m_J} = \frac{1}{2}$. This means that $m_S|\lhd m_J$ is independent of m_S, hence there is no effective probabilistic method to predict $m_S|\lhd m_J$ given m_S, or to infer m_S given $m_S|\lhd m_J$. □

But does the sufficient condition for this negative result hold in practice? As we see from our experiments, although the value of $p^0_{0|\lhd 0} + p^0_{0|\lhd 1}$ and $p^0_{1|\lhd 0} + p^0_{1|\lhd 1}$ is not exactly 1, it is however very close to 1 on both radios (see Table 6 and 7). This implies that the probability of $m_S|\lhd m_J$ being independent of m_S is quite high. We conclude that it is hard to effectively predict $m_S|\lhd m_J$ given m_S and also hard to effectively recover m_S given $m_S|\lhd m_J$ with a probabilistic approach.

4.6 Network Recovery

So far, we have only considered a single receiver. We now address whether multiple receivers can do better at recovering from a corrupted value, in particular, using the simple logic of majority counting.

We propose a bit level jamming model for multiple receivers in Table 11. The model has two metrics, receiver difference probability and the probability that the majority vote is in fact the same as the source bit. More specifically, the first metric, $Pr(v_{R_i} = b, v_{R_j} = \neg b)$, is the probability that receiver R_i and R_j $(i \neq j)$ receive different bit values, and the second is $Pr(v_m = b)$ where v_m is the majority vote value in the network and b is the source bit value.

Source bit	Reception Difference	Majority Success
b	$Pr(v_{R_i} = b, v_{R_j} = \neg b)$	$Pr(v_m = b)$

Table 11. Bit level jamming model for multiple receivers. b is source bit transmitted from S.

Note that if the reception difference probability is very low (close to 0) for many receiver pairs, then most receivers would get the same value. In this case, it is conceivable that majority counting yields some added confidence in the recovery process. Note also that this metric cannot be too high (close to 1) for all pairs: by way of illustration, consider an example of three receivers, R_A, R_B and R_C, where if $Pr(v_{R_A} = b, v_{R_B} = \neg b) \approx 1$ and $Pr(v_{R_A} = b, v_{R_C} = \neg b) \approx 1$, then $Pr(v_{R_B} = b, v_{R_C} = \neg b) \approx 0$.

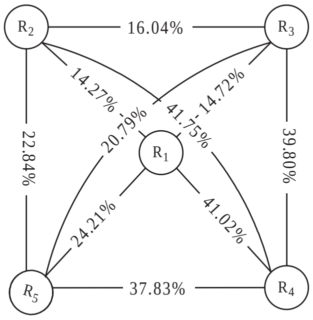


Figure 12. Bit level reception difference using experimental topology *I*

We present the results for multiple receivers in experiments using topology *I* and *II* . Figure 12 and 13 show bit level reception difference among five receivers in topology *I* and *II* respectively. Basically, we see that the value of $Pr(v_{R_i} = b, v_{R_j} = \neg b)$ in our bit level model is non-trivial. Some are relatively low, around 20%, while some are around 50%, close to the probability of random guess. In addition, it is not the case that symmetric locations experience similar corruption results. For example, the locations of R_3 and R_5 are symmetric relative to S and J, however, more than 20% bits are received differently. Further, their relation among other nodes are also different. For example, $Pr(v_{R_1} = b, v_{R_3} = \neg b) = 14.72\%$ and $Pr(v_{R_1} = b, v_{R_5} = \neg b) = 24.21\%$ in Figure 12. The reason for such a difference involves the randomness of corruption and hardware variability.

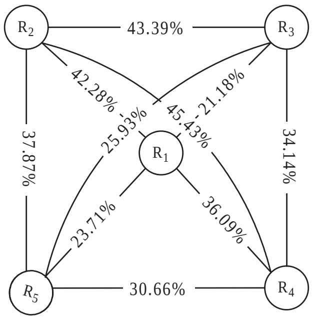


Figure 13. Bit level reception difference using experimental topology *II*

In Table 12 and 13, we list the value of $Pr(v_m = b)$ based on majority counting. We see that the counting logic does not work well enough to infer a packet of multiple source bits. For example, when the number of nodes in the majority set is 3, the probability of success is around 60% and 70% respectively using topology *I* and *II*. When the number of the nodes in the majority set increases, the probability decreases. For example, when the number of the nodes is 5, the probability of success is around 50%, which is not better than random guessing. None of these results are good enough to recover typical-sized packets which have hundreds of bits.

Z_m=3	61.16%
Z_m=4	57.61%
Z_m=5	47.85%

Table 12. The probability of a successful majority vote single-bit guess in experiment 4 using topology *I*. Z_m is the number of nodes in the majority count (Observation 3)

Z_m=3	72.30%
Z_m=4	63.14%
Z_m=5	57.51%

Table 13. The probability of successful majority vote single-bit guess in experiment 4 using topology *II*. Z_m is the number of nodes in the majority count (Observation 3)

5 Discussion

As wireless networks evolve to support applications that are launched remotely and also to allow multiple applications from different entities to run concurrently in the same fabric, in-network attacks including internal jamming are becoming increasingly feasible. Our observations in Section 4 indicate that intentional jamming can be readily designed to disrupt the communications between the sender and potential receivers with high probability, however it is hard for the jammer to be undetected, or to fool the receiver if the received message is indeed valid.

From the jammer's point of view, it is important to design her attack so that her identity is not revealed even if jamming is detected. From the receivers' point of view, it is important to efficiently detect the jammer without involving many innocent nodes. We have addressed both these issues, and commented that it is hard for the jammer to fool a receiver into rejecting an uncorrupted packet from the sender.

We have also observed that not only can a single receiver not recover source packets correctly in the presence of jamming, but also a network (a set of receivers) can not recover the source packets based on simple counting logic, not to mention the processing and communication overhead in potential cooperative recovery procedures. However, it is not clear whether it is possible to do network based recovery using more complicated methods, such as selective recovery set where a node is only eligible for recovery if it believes there is only a very small portion of the message that is corrupted.

Perhaps most interesting is that low power wireless networks can profitably exploit jamming capabilities. If a node intentionally jams the wireless channels between a sender and potential eavesdroppers, it can successfully prevent messages from being heard by eavesdroppers. A central underlying question is how to deliver a message to some legitimate nodes secretly without being overheard by adversaries. One work in this direction is from an information theory perspective [12]; in other work, we have realized a protocol level solution to this problem as well (reference omitted). Another interesting question is how to use jamming capabilities to hide communications from malicious jammers in general, especially when the jammer only relies on certain patterns (e.g., preambles) being sensed over the channel.

6 Concluding Remarks

Jamming is an important topic for low-power wireless network adoption. In this paper, we performed a systematic study of the jamming capabilities and limitations achievable using only low power devices.

We investigated packet level and bit level corruption via jamming using multiple platforms, multiple transmission power levels, and choice of location and of communication content. We observed that it is easy for J to choose a location and a power level so as to corrupt a bit or a packet from a given S to a given R. While it is hard for R to detect corruption at the level of individual bits, it is easy at the level of packets. The probability of corrupting a bit via jamming is nontrivial and fluctuates dramatically over time, and this makes predicting the jamming output value hard. Conversely, even if J uses a known protocol and known values for jamming, it is hard to recover from the corrupted value received at R the original value sent by S. In fact, the probability of successful recovery at R is close to the probability of random guessing, even when multiple R cooperate.

In order to accurately evaluate the corruption induced by jamming, we performed our experiments in a electromagnetically silent office environment. We expect more loss and corruptions if the background is more noisy, but it would be worthwhile to further investigate jamming in a noisy environment.

We have deliberately limited our attention in this paper to internal jamming, which is well suited to studying considerations like controlling devices with artificial noise to achieve communication secrecy. That said, another avenue for further study will be to see which of the negative results here can be inherited for the case of external jamming.

7 References

[1] A. Cerpa, J. L. Wong, M. Potkonjak, and D. Estrin. Temporal properties of low power wireless links: Modeling and implications on multihop routing. *CENS Technical Report 0044*, 2005.

[2] E. Cota-Robles and J. P. Held. A comparison of windows driver model latency performance on windows nt and windows 98. *In Proceedings of the Third Symposium on Operating System Design and Implementation (OSDI).*

[3] H. Delfs and H. Knebl. Introduction to cryptography: Principles and applications. *Springer*, 2nd Edition, 2007.

[4] D. Ganesan, D. Estrin, A. Woo, D. Culler, B. Krishnamachari, and S. Wicker. Complex behavior at scale: An experimental study of low-power wrireless sensor networks. *Technical Report CS TR 02-0013, UCLA*, 2002.

[5] E. Geraniotis. Effect of worst case multiple partial-band noise and tone jammers on coded fh/ssma systems. *IEEE Journal on Selected Areas in Communications*, 8(4):613–627, May 1990.

[6] P. Gupta and P. Kumar. The capacity of wireless networks. *IEEE Transaction on information theory*, 46(2), Mar 2000.

[7] I. W. McKeague and C. R. Baker The coding capacity of mismatched gaussian channels. *IEEE Transactions on Information Theory*, 32(3):431–436, May 1986.

[8] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu. Impact of interferece on multi-hop wireless network performance. *Proceedings of the 9th annual ACM international conference on Mobile computing and networking*, pages 66–80, Sep 2003.

[9] K.-H. Kim and K. G.Shin. On accurate measurement of link quality in mulit-hop wireless mesh networks. *ACM Mobicom*, 2006.

[10] C. Kuo, M. Luk, R. Negi, and A. Perrig. Message-in-a-bottle: User-friendly and secure key deployment for sensor nodes. *ACM SenSys*, 2007.

[11] L. Pap. A general jamming model of spread spectrum systems. *IEEE Internatiomal Conference on Communications, ICC'93*, 1/3(2):473–477, May 1993.

[12] L. Lai, H. E. Gamal, and H. V. Poor. The wiretap channel with feedback: Encryption over the channel. *submitted to the IEEE Transactions on Information Theory*, April 2007.

[13] Y. W. Law, L. van Hoesel, J. Doumen, P. Hartel, and P. Havinga. Energy-efficient link-layer jamming attacks against wireless sensor network mac protocols. *ACM Security Sensor Ad-hoc Networks (SASN)*, 2005.

[14] D. J. C. MacKay. Information theory, inference, and learning algorithms. *Cambridge: Cambridge University Press*, 2003.

[15] Mallik, R.K., R. Scholtz, and G. Papavassilopoulos. Analysis of an on-off jamming situation as a dynamic game. *IEEE Transactions on Communications*, 48(8):1360–1373, Aug 2000.

[16] J. M. McCune, E. Shi, A. Perrig, and M. K. Reiter. Detection of denial of message attacks on sensor network broadcasts. *Proc. IEEE Symposium on Security and Privacy*, 2005.

[17] MoteIV. http://www.moteiv.com/.

[18] V. Navda, A. Bohra, S. Ganguly, and D. Rubenstein. Using channel hopping to increase 802.11 resilience to jamming attacks. *IEEE Infocom Mini-Symposium*, 2007.

[19] L. Qiu, Y. Zhang, F. Wang, M. K. Han, and R. Mahajan. A general model of wireless interference. *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, Sep. 2007.

[20] R. Negi and S. Goel Secret communication using artificial noise. *Vehicular Technology Conference*, 3:1906–1910, 2005.

[21] CC2420 Data Sheet. http://www.chipcon.com/files/cc2420 _data _sheet _1_3.pdf.

[22] D. Son, B. Krishnamachari, and J. Heidemann. Experimental analysis of concurrent packet transmissions in low-power wireless networks. *ACM SenSys*, 2006.

[23] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis. Understanding the causes of packet delivery success and failure in dense wireless sensor networks. *ACM SenSys*, 2006.

[24] K. Srinivasan and P. Levis. RSSI is under appreciated. In *Proc. of the Third Workshop on Embedded Networked Sensors, EmNets 2006, Boston, MA*, May 2006.

[25] B. T. The gaussian test channel with an intelligent jammer. *IEEE Transactions on Information Theory*, 29(1):152–157, Jan 1983.

[26] D. Thuente and M. Acharya. Intelligent jamming in wireless networks with applications to 802.11b and other networks. In *Proceedings of the 25th IEEE Communications Society Military Communications Conference (MILCOM)*, October 2006.

[27] K. Whitehouse, A. Woo, F. Jiang, J. Polastre, and D. Culler. Exploiting the capture effect of collision detection and recovery. *IEEE Workshop on Embedded Networked Sensors,EmNetS-II*, 2005.

[28] A. Woo, T. Tong, and D. Culler. Taming the underlying challenges of

reliable multihop routing in sensor networks. *ACM SenSys*, 2003.

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

[30] W. Xu, K. Ma, W. Trappe, and Y. Zhang. Jamming sensor networks: Attack and defense strategies. *IEEE Networks Special Issue on Sensor Networks*, 20(3):41–47, May 2006.

[31] W. Xu, W. Trappe, Y. Zhang, and T. Wood. The feasibility of launching and detecting jamming attacks in wireless networks. *ACM MobiHoc*, pages 46–57, 2005.

[32] J. Zhao and R. Govindan. Understanding packet delivery performance in dense wireless sensor networks. *ACM SenSys*, pages 1–13, 2003.