STIR-ing the Wireless Medium with Self-Tuned, Inference-Based, Real-Time Jamming

Bruce DeBruhl, Yu Seung Kim, Zachary Weinberg, Patrick Tague
Carnegie Mellon University
Electrical and Computer Engineering
{debruhl, yuseungk, zackw, tague} @cmu.edu

***Abstract*—Jamming, broadcasting to intentionally interfere with wireless reception, has long been a problem for wireless systems. Recent research demonstrates numerous advances in jamming techniques that increase attack efficiency or reduce the probability an attack will be detected by choosing attack parameters based on a system's configuration. In this work, we extend the attacker's capabilities by modifying the attack parameters in response to the observed performance of the target system, effectively creating a feedback loop in our attack model. This framework allows for more intricate attack models that are tuned online allowing for closer to optimal attacks against legitimate systems. To show the feasibility of the listening and attacking framework we introduce an attack called Self-Tuned, Inference-based, Real-time jamming or *STIR-jamming*. This attack listens to legitimate communication traffic, infers the systems performance, and optimizes jamming parameters. We propose the two types of STIR-jamming, *mSTIR-jamming* and *tSTIR-jamming*, and implement these attacks against an IEEE 802.15.4 link as a case study. With the empirical results, we demonstrate the attack system adapting to various scenarios and finding stable solutions.**

I. Introduction

Wireless communications allow for the open and convenient exchange of data without the use of costly wired connections [1]. This provides great benefits to legitimate users but makes them susceptible to denial-of-service (DOS) attacks known as jamming attacks [2].

Traditionally, a communication system mitigates the effects of jamming by raising the cost of an equally effective attack [3] using spread spectrum techniques. However, spread spectrum is not effective against wide-band or high-power jamming attacks. Recently, to dissuade attackers from mounting high-power attacks, jamming detection techniques have been proposed [4], [5], [6], [7]. If a jamming attack can be detected, there are techniques to mitigate their effect, including spatial or frequency retreats [7], jamming aware routing [8], or the jammer can be decommissioned.

Recent literature [9] also motivates using low-power jamming from the vantage point of energy conservation. Mounting jamming attacks from mobile platforms is desirable for some attack scenarios and efficient low-power attacks can increase a mobile attackers lifespan. Understanding, designing, and optimizing these type of low-power attacks allows for insights into threats on critical devices and infrastructure (Medical, military, safety, etc.). There has been some interest in using reactive jamming to mount efficient low power attack [10]. These attacks are feasible but require special hardware that is expensive and infeasible to implement in commodity hardware.

Intelligent jamming attacks have been designed recently but generally just using a static strategy against a particular protocol [2]. A low-power attack that has been suggested is short form periodic jamming (SFPJ) [11]. The SFPJ attack uses very short loud bursts to interfere with DSSS communications. Such an attacker is very effective against IEEE 802.15.4, which is commonly used in sensor networking, but requires tuning against a system and node geometry to obtain good results. In this work we present Self-Tuning, Inference-based, Real-time jamming or *STIR-jamming* which explores an attacker which continually adapts its attack parameters with performance informations obtained from the system under attack. To do this we envision an attacker with both inference and jamming capabilities. The attacker's inference capability estimates the performance of the legitimate network in real time. Using the inferred information, the attacker then tunes its jamming attack to achieve better performance. This attack differs greatly from traditional jamming in that it looks to continually optimize the physical layer jamming attack in real time. This type of attack could use commodity sensor network hardware to make an efficient and long lasting attack, showing the need for more research in detecting and mitigating low-power jamming attacks against DSSS.

The major contributions of this work are as follows.

- We propose the STIR-jamming framework for continually modified jamming attacks using an observe-and-attack feedback loop between the attacker and the target system.
- We present two instances of STIR-jamming. The first algorithm performs repeated parameter optimization using a reference model to predict the effect of parameter selection. The second algorithm iteratively tunes the parameters to increase or decrease the attack impact.
- We show results from a proof-of-concept implementation of the two STIR-jamming algorithms against a link using the 802.15.4 protocol and empirically show these attacks achieve relatively stable performance.

The remainder of this paper is organized as follower. In Section II, we explore jamming attack and defense literature. In Section III, we introduce our system model and in Section IV we introduce the STIR-jamming framework and algorithms. We show how the two STIR-jamming algorithms can be

This research was supported by CyLab at Carnegie Mellon University under grant DAAD19-02-1-0389 from the Army Research Office and by the National Science Foundation under grant CNS-1149582. Bruce DeBruhl is supported by an NDSEG fellowship. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of ARO, CMU, NSF, or the U.S. Government or any of its agencies.

implemented against an 802.15.4 link in Section V and present empirical results in Section VI. Finally, Section VII concludes the paper.

II. Related Work

Before presenting our proposed attack model, we discuss related work that provides the basis for our investigation.

One of the simplest forms of jamming is the modulation of a single tone at the carrier frequency, known as tone jamming. Spread spectrum is an effective defense against this basic jamming attack [1], aiming to increase the cost for an attacker to mount an equally-effective jamming attack. One such technique is direct sequence spread spectrum (DSSS) which maps a narrow-band signal to a wider frequency band providing increased robustness of the transmission against a narrow-band attacker through redundancy. A second technique is frequency hopping spread spectrum (FHSS), in which a sender and receiver "hop" between channels using a pre-determined schedule. FHSS is very effective at defending against narrow-band attacks provided the two nodes are time-synchronized, the hopping schedule remains secret, and a sufficient number of orthogonal channels are used [12].

Detection of jamming attacks via system monitoring is another approach to jamming mitigation, allowing the system under attack to change its operation or impose a penalty on the attacker. One such detection technique is to monitor network performance metrics and verify consistency. For example, observing a low packet delivery ratio and consistently high received signal strength, a receiver may infer the presence of a jammer [13]. Such detection techniques can then be used to trigger anti-jamming mechanisms [14], [7], [15].

To counteract the anti-jamming capabilities of spread spectrum, attackers must either increase their resource usage or increase their attack efficiency [13]. An efficient alternative is through random or periodic jamming, in which the jammer alternates between an attacking mode and a sleeping mode to reduce energy usage [9]. Another alternative which combines efficiency and effectiveness is reactive jamming, in which the attacker listens to the channel and transmits a high-power jamming signal when it senses a packet transmission [10]. An additional benefit of random, periodic, and reactive jamming attacks is the reduced likelihood of being detected, a natural protection against the detect-and-respond approaches above.

Another way that attackers can increase efficiency and reduce detectability is by incorporating higher-layer information in the jamming attack formulation. Jamming attackers can incorporate MAC layer information to precisely time jamming emissions [2], [9], [16], for example by jamming the channel when acknowledgement (ACK), clear-to-send (CTS), or data packets are expected according to protocol schedules. Attackers can further incorporate network layer information by observing traffic flows and tuning jammers across the network to minimize network throughput [17].

Attackers can also adapt jamming behaviors based on system performance. Maintaining a network history allows an attacker to decide whether or not it will jam at a particular time using a game theoretic approach [16] or choosing which of a group of jammers should be used at a particular time [17]. Our work in this paper is similar in spirit to these adaptive attacks, but we investigate parameter adaptation at the physical layer and with a much finer granularity of control, noting that our approach can be combined with the above-mentioned attacks to further increase efficiency and reduce detectability.

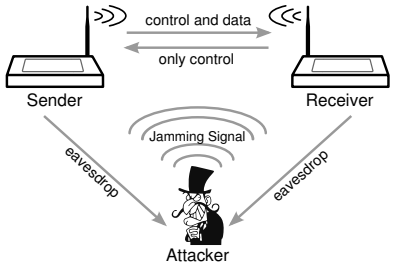


Fig. 1: Our attack model gives the attacker both observation and jamming capabilities allowing for continual modification of attack parameters from observed performance characteristics.

TABLE I: We provide a summary of the notation used through the remainder of this paper.

Definitions	
k	Discrete time variable
$u_k(t)$	Signal broadcast by the jammer in time step k
$w_k(t)$	Signal observed by the jammer in time step k
$\mathcal{H}$	System transfer function
$\boldsymbol{\phi}_k$	Performance parameters for time step k
$\mathcal{S}$	Jamming strategy
$\boldsymbol{p}_k$	Jammer parameters for k^{th} time step
$\boldsymbol{M}(\mathcal{S}, \boldsymbol{p})$	Jamming metrics for strategy $\mathcal{S}$
$\iota(\mathcal{S}, \boldsymbol{p})$	Jamming metric for impact
$\varsigma(\mathcal{S}, \boldsymbol{p})$	Jamming metric for stealth
$\eta(\mathcal{S}, \boldsymbol{p})$	Jamming metric for expenditure
Π_k	Packet delivery ratio for time step k
P_{det}	Probability of attack detection
$\mathcal{G}$	Discrete mapping of $\boldsymbol{p} \mapsto \boldsymbol{\phi}$
ϵ	Error in estimate of $\boldsymbol{\phi}$
μ	Normalized combination of jamming metrics
T	Target value for tuning based STIR-jamming

III. System Model

In this section, we introduce the system model and notation used for the remainder of this work. We consider a communication system containing a sender, a receiver, and an attacker as shown in Figure 1. The sender transmits both data and control packets, and the receiver responds only with corresponding control packets. We assume that all parties remain in communication range of each other and that the sender and receiver are able to communicate with low error under benign conditions.

The attacker in our model performs two functions: observing and jamming. In the jamming role, the attacker can choose from a variety of jamming strategies and corresponding parameters. In the observing role, the attacker infers performance characteristics of the sender and receiver. In order to formulate the jamming attack model of interest, we first present a mathematical model for this system from the perspective of the attacker. The notation used for the remainder of this paper is given in Table I.

Although the signals transmitted and received by the attacker are continuous, the packet observations and attack decisions are discrete-time events. We thus define the system model,

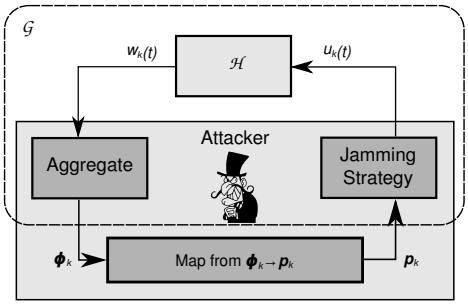


Fig. 2: In our system model, the attacker generates a jamming signal $u_{k+1}(t)$ in response to previously observed signals $w_k(t)$ from the target system, represented by a transfer function $\mathcal{H}$. We further abstract this model to relate the jammers parameter selection $\boldsymbol{p}_k$ to the observed signal $\boldsymbol{\phi}_k$.

as viewed by the attacker, in terms of a discrete time step k, where the length of k can be periodic or event driven.

We define the continuous signal that the attacker broadcasts during time step k as $u_k(t)$ and the continuous signal observed by the attacker during time step k as $w_k(t)$. To capture the relationship between the jamming signal $u_k(t)$ and the jammer-influenced observation $w_k(t)$, we define the transfer function $\mathcal{H}$. In terms of the model in Figure 1, $\mathcal{H}$ replaces the sender, receiver, and the channels between the three parties, yielding the jammer-centric mathematical model shown in Figure 2.

The continuous signal $w_k(t)$ is composed of both control and data packet communication between the sender and receiver. The observed communications are under the influence of both noise and fading over the wireless sender-to-attacker and receiver-to-attacker channels respectively. In order to facilitate the discrete decision process of the attacker, we suppose that the attacker aggregates and summarizes the time domain signals into a vector $\boldsymbol{\phi}_k$, which represents an observation of a set of sender-to-receiver performance metrics of interest. We further discuss the mapping between the observed signal $w_k(t)$ and the summary $\boldsymbol{\phi}_k$ in Section IV-A.

We define $u_k(t)$ as the continuous signal that the jammer transmits using a mapping from the discrete parameter space of the jammer to the actual jamming signal. We define a jamming strategy function $\mathcal{S}$ and a jamming parameter vector $\boldsymbol{p}_k$, such that $u_k(t) = \mathcal{S}(\boldsymbol{p}_k)$. The strategy $\mathcal{S}$ defines the type of jamming attack being mounted, and the parameter vector $\boldsymbol{p}_k$ specifies a number of parameters that are left free by the general strategy.

Similar to the continuous transfer function $\mathcal{H}$ that maps $u_k(t) \mapsto w_k(t)$, we define a discrete representation $\mathcal{G}$ of the transfer function that virtually maps the jamming strategy $\mathcal{S}$ and parameter vector $\boldsymbol{p}_k$ to the observation summary $\boldsymbol{\phi}_k$. The ability for the attacker to dynamically modify its attack is thus determined by choosing $\boldsymbol{p}_k$ according to estimates of $\mathcal{G}$ and observations $\boldsymbol{\phi}_k$.

IV. STIR-Jamming

Using the system model given in Section III, we next introduce *STIR-jamming* or self-tuned, inference-based, real-time jamming, in which the attacker observes the sender's and receiver's communications to continually modify attack parameters with the aim of mounting high-efficiency jamming attacks. We describe the actions of STIR-jamming as follows.

1) **Observe**: From the observed signal $w_k(t)$, create a summary of the performance metrics $\boldsymbol{\phi}_k$.
2) **Estimate**: Given the previously chosen parameters $\boldsymbol{p}_k$ and the observation $\boldsymbol{\phi}_k$, characterize the effect of the attack efforts.
3) **Optimize**: Use the estimate above to select parameters $\boldsymbol{p}_{k+1}$ according to a given collection of jamming metrics $\boldsymbol{M}(\mathcal{S}, \boldsymbol{p})$.

Before presenting the algorithms in Section IV-C, we describe the estimation of performance characteristics $\boldsymbol{\phi}_k$ in Section IV-A and selection of jamming metrics $\boldsymbol{M}(\mathcal{S}, \boldsymbol{p})$ in Section IV-B.

A. Observation

In the observations phase of STIR-jamming the attacker converts an observed signal $w_k(t)$ into performance parameters $\boldsymbol{\phi}_k$. There are two major considerations for observation, what information is desired and how to obtain that information. What information is desired depends dramatically on the goal of the attacker. In this work, we focus on attacking a single link so we consider packet delivery ratio (PDR) as a metric.

The second question about the observation of a legitimate system is how to obtain this information. To do this an attacker can observe packet transmissions only when it is not jamming, and use statistical analysis to estimate the PDR metric. Consideration of appropriate statistical techniques and the resulting estimation accuracy is left as future work.

B. Jamming Metrics

As previously defined, the jamming metrics in the set $\boldsymbol{M}(\mathcal{S}, \boldsymbol{p})$ are used to gauge the effectiveness of a jamming strategy $\mathcal{S}$ with parameters $\boldsymbol{p}$. We consider three jamming metrics that are important for an effective jamming attack. These metrics are *impact*, *stealth*, and *expenditure*, respectively denoted as $\iota(\mathcal{S}, \boldsymbol{p})$, $\varsigma(\mathcal{S}, \boldsymbol{p})$, and $\eta(\mathcal{S}, \boldsymbol{p})$.

1) Impact: We define impact $\iota(\mathcal{S}, \boldsymbol{p})$ as the amount of degradation a jammer causes in the sender-receiver system. One measure of impact is reduction of the PDR, also used as a performance metric in Section IV-A. In other words, a lower PDR indicates a higher impact.

2) Stealth: We define stealth $\varsigma(\mathcal{S}, \boldsymbol{p})$ as the ability of a jammer to evade detection by the sender-receiver communication system. Stealth is important because if an attack is detected, the legitimate nodes can take appropriate actions to compensate for the attack impacts or directly penalize the attacker. We propose using a stealth metric that is inversely proportional to the probability of detection P_{det}.

3) Expenditure: We define expenditure $\eta(\mathcal{S}, \boldsymbol{p})$ as the amount of resources the attacker uses to mount an attack. Many metrics can be considered for the resource usage in expenditure (e.g., bandwidth, energy, broadcast time, etc) but we focus specifically on average power in this work, supposing that the attacker could be a battery-powered mobile device or sensor node. Average power is a straightforward metric defined in terms of the amount of time spend jamming and the jamming signal power and can be used as an indication of energy consumption.

C. Attack Algorithm

In this section, we present our attack algorithm which uses the observation methods presented in Section IV-A to optimize the jamming metrics in Section IV-B. We consider estimation and optimization with two approaches. The first approach is *model-based* or *mSTIR-jamming*, which uses a non-linear reference model and rigorous optimization methods. The second approach is *tuning-based* or *tSTIR-jamming*, which uses a technique of adjusting jamming parameters to increase or decrease the attack's effect without the need for designing reference model. In what follows, we present jamming algorithms for these two approaches and a comparison of the resulting attacks.

1) mSTIR-jamming: The mSTIR-jamming algorithm involves the three steps of observing the performance parameters $\boldsymbol{\phi}_k$, estimating the discrete transfer function $\mathcal{G}_{k+1}$ that is expected in the next time step, and using the estimate of $\mathcal{G}_{k+1}$ to choose the subsequent attack parameters $\boldsymbol{p}_{k+1}$. The proposed algorithm uses the observation technique described in Section IV-A to compute the performance parameters $\boldsymbol{\phi}_k$ at each time step.

The algorithm relies on the attacker's ability to compute an estimate $\widehat{\mathcal{G}}_k$ of the discrete transfer function $\mathcal{G}_k$ which maps $(\mathcal{S}, \boldsymbol{p}_k) \mapsto \boldsymbol{\phi}_k$, i.e. mapping the jammer's effort to its observed effect. As part of the mSTIR-jamming algorithm, the attacker updates its estimate of the discrete transfer function, using $\boldsymbol{\phi}_k$ and $\widehat{\mathcal{G}}_k$ to estimate $\widehat{\mathcal{G}}_{k+1}$. To facilitate this process, we introduce a scalar error value ϵ_k computed using the function $\text{error}(\boldsymbol{\phi}_k, \widehat{\mathcal{G}}_k, \boldsymbol{p}_k)$ as

$$\epsilon_k = \text{error}(\boldsymbol{\phi}_k, \widehat{\mathcal{G}}_k, \boldsymbol{p}_k). \tag{1}$$

Using (1), we define the function for updating the transfer function estimate as

$$\widehat{\mathcal{G}}_{k+1} = \text{update}(\widehat{\mathcal{G}}_k, \epsilon_k). \tag{2}$$

In Section V-C, we show how the update function can be constructed by adding a tuning parameter into the transfer function model that can be updated to better match the expected and observed performance metrics.

The third step in the mSTIR-jamming attack is to choose the attack parameters to optimize the desired jamming attack metrics. This requires a decision of how to jointly optimize the jamming metrics in the set $\boldsymbol{M}(\mathcal{S}, \boldsymbol{p})$. Toward this end, we define the optimization objective function $\mu(\mathcal{S}, \boldsymbol{p}, \widehat{\mathcal{G}}_{k+1})$ as the combination of metrics in $\boldsymbol{M}(\mathcal{S}, \boldsymbol{p})$ the attacker will aim to maximize. Assuming the existence of lower and upper bounds on the jamming parameters $\boldsymbol{p}$, denoted by $\boldsymbol{p}_{min}$ and $\boldsymbol{p}_{max}$, respectively, we define the optimization problem as

$$\begin{aligned} &\underset{\boldsymbol{p}}{\text{maximize}} && \mu(\mathcal{S}, \boldsymbol{p}, \widehat{\mathcal{G}}_{k+1}) \\ &\text{subject to} && \boldsymbol{p}_{min} \leq \boldsymbol{p} \leq \boldsymbol{p}_{max}. \end{aligned} \tag{3}$$

One of the benefits of mSTIR-jamming is that it does not require a perfect model of the system and parameters to be efficient. In this paper we use a simple reference model based off of Friis equation to optimize and, as shown in section VI, obtain good results. We anticipate that in future work it is possible to use universal approximators [18] and obtain good results or include context awareness to make the system adaptation even better.

2) tSTIR-jamming: The mSTIR-jamming attack is effective in optimizing the jamming parameters, but relies on availability of a usable approximation of the system $\mathcal{G}$. In many cases, estimating $\mathcal{G}$ with sufficient accuracy may be prohibitively costly or even infeasible. Hence, we present tSTIR-jamming to eliminate the need for this expensive estimation step.

tSTIR-jamming uses the same observation method proposed in Section IV-A. We define a target value $\boldsymbol{T}$ which serves as the desired value for $\boldsymbol{\phi}$. We then measure the error ϵ_k between the observation and the target as

$$\epsilon_k = \text{error}(\boldsymbol{\phi}_k, \boldsymbol{T}), \tag{4}$$

where a positive ϵ_k indicates that the attack was too aggressive and a negative value indicates the attack was not aggressive enough. We define the decision variable $\delta_k \in \{-1, 0, 1\}$ at time step k to indicate whether the attack effort should increase, remain unchanged, or decrease for the subsequent time step. We let ρ indicate the threshold at which the attack should be changed, yielding

$$\delta_k = \begin{cases} -1, & \text{if } \epsilon_k \leq -\rho \\ 1, & \text{if } \epsilon_k \geq \rho \\ 0, & \text{else.} \end{cases} \tag{5}$$

We use an intuitive update algorithm for the tSTIR-jamming attack. If $\delta_k = 0$, then no change to the attack parameters is required, so $\boldsymbol{p}_{k+1} = \boldsymbol{p}_k$. If δ_k is non-zero, the attack effort is decreased or increased accordingly. To modify the attack parameters, we use a one-step transition in any one of the parameters in $\boldsymbol{p}_k$, effectively taking a one-dimensional step in the parameter space. Letting $\boldsymbol{p}_k^+$ and $\boldsymbol{p}_k^-$ be the sets of all possible one-step parameter vectors with increased and decreased attack impact $\iota(\mathcal{S}, \boldsymbol{p}_k)$, respectively, the next parameters $\boldsymbol{p}_{k+1}$ are chosen from the corresponding one-step parameter space using the expenditure metric $\eta(\mathcal{S}, \boldsymbol{p}_k)$.

In general, the resulting conditional optimization problem is thus given by

$$\begin{aligned} &\text{if } \delta_k = 0 \\ &\quad \boldsymbol{p}_{k+1} = \boldsymbol{p}_k \\ &\text{else if } \delta_k = 1 \\ &\quad \boldsymbol{p}_{k+1} = \underset{\boldsymbol{p} \in \boldsymbol{p}_k^-}{\arg\min}\ \eta(\mathcal{S}, \boldsymbol{p}) \\ &\text{else if } \delta_k = -1 \\ &\quad \boldsymbol{p}_{k+1} = \underset{\boldsymbol{p} \in \boldsymbol{p}_k^+}{\arg\min}\ \eta(\mathcal{S}, \boldsymbol{p}) \end{aligned} \tag{6}$$

A two-parameter example of the one-step transitions used in the tSTIR-jamming attack is illustrated in Figure 3 for two parameters p_1 and p_2. If $\boldsymbol{p}_k = (p_1, p_2)$ and $\delta_k = -1$, the attacker can choose $\boldsymbol{p}_{k+1} = (p_1 + \Delta_1, p_2)$ or $\boldsymbol{p}_{k+1} = (p_1, p_2 + \Delta_2)$, where Δ_1 and Δ_2 are pre-determined step sizes for each parameter, depending on which results in lower energy expenditure. A similar state-transition diagram can be envisioned in a higher-dimensional space for arbitrary parameters $\boldsymbol{p}$.

3) Comparison: These two algorithms both have advantages and disadvantages. mSTIR-jamming is able to estimate information on stealth which could be a huge advantage in a stealth-critical situation. The cost for using the model-based attack, however, is a large increase in computation because

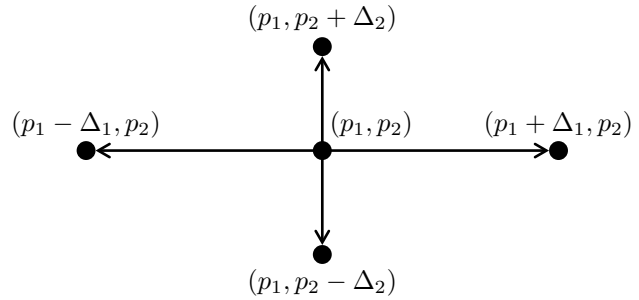


Fig. 3: State transition rule. At the beginning of each time step, the attacker is operating at a point (p_1, p_2) in the state space; this is the center dot in the diagram. For the next time step, it can remain where it is (holding both p_1 and p_2 constant) or move to one of the other four points (incrementing or decrementing p_1 or p_2, but not both).

the optimization problems in mSTIR-jamming are non-trivial optimization problems for attackers with little computational power. tSTIR-jamming is not able to cope with stealth due to relaxation of the model, but it is more computationally efficient. Ultimately the choice of algorithm depends on the availability of a reasonable reference model, the need to predict stealth, and the associated computation overhead.

V. Case Study: IEEE 802.15.4

In this section, we use the STIR-jamming framework presented in Section IV to design a proof of concept for both model-based and tuning-based STIR-jamming attacks targeting a sender-receiver system using the IEEE 802.15.4 protocol [19]. 802.15.4 is a popular low-power personal area network protocol used in many sensor network platforms. The implementations suggested in this section are just proof-of-concepts and not optimally designed attacks, we leave further research into optimal controller design for STIR-jamming as future work.

A. IEEE 802.15.4 overview

The 802.15.4 protocol is a low-power personal area network protocol commonly used in wireless sensor network platforms. In this work we are primarily concerned with the 2.4 GHz physical layer protocol defined in the 802.15.4 specification [19]. This protocol uses direct sequence spread spectrum to provide robust transmission against noise and interference. The use of DSSS maps 4-bit symbols to 32-chip quasi-orthogonal patterns that are then sent at a data rate of 2 Mcps (mega-chips per second), yielding an effective data rate of 250 kbps.

We note that while the use of DSSS provides strong chip-error correction capability at the symbol level, there is no packet level error correction in the 802.15.4 standard. Instead, a two-byte checksum is included in each packet to allow the receiver to detect, with high probability, if the packet is received in error but does not allow for error correction.

B. Short Form Periodic Jamming

Short Form Periodic Jamming (SFPJ) is an efficient class of jamming attacks against the 802.15.4 protocol described in Section V-A which can be tuned to have high impact with low cost [20].

In a SFPJ attack, the jamming signal is cycled on and off with a period on the order of the symbol duration, as compared to the order of the packet duration as proposed in previous work [20]. A periodic jamming attack can be defined in terms of three jamming parameters: period, defined as the number of symbols per on-off cycle; duty cycle, defined as the percent of time the jammer is on; and amplitude, defined as the jamming signal power. It can be shown that it is possible to effectively jam an 802.15.4 link with a small duty cycle. It has been shown that SFPJ can interfere with 95% of 802.15.4 communications with a duty cycle under 10% [20].

1) SFPJ Metrics against 802.15.4: As discussed in Section IV-B, STIR-jamming is largely dependent on jamming metric definitions. Here we formulate specific instances of the *impact*, *expenditure*, and *stealth* metrics for the 802.15.4 network scenario that are used for both mSTIR-jamming and tSTIR-jamming attacks. We define our metrics in terms of the jamming signal amplitude a and duty cycle d, holding the signal period constant.

Impact: We consider packet delivery ratio (PDR) as the measure of impact on the IEEE 802.15.4 system. To do this we estimate the PDR as a function of the amplitude a and duty cycle d. Because no error correction is used in the 802.15.4 protocol, the PDR $\Pi(a, d)$ can be estimated as

$$\Pi(a, d) = (1 - P_s(a))^{nd}(1 - P_s(0))^{n(1-d)}, \tag{7}$$

where $P_s(x)$ is the probability of symbol error with jamming signal average power x and n is the number of symbols per packet. For the 802.15.4 protocol, the symbol error $P_s(x)$ can be further decomposed as

$$P_s(x) = \sum_{i=17}^{32} \binom{32}{i} P_c(x)^i (1 - P_c(x))^{32-i}, \tag{8}$$

where $P_c(x)$ is the probability of chip error for 802.15.4 under attack with a jamming signal amplitude x. This probability of chip error can be estimated as

$$P_c(x) = \frac{1}{2} Q\left(\sqrt{\frac{T_c F(d_{tr}) S_{tx} - T_c F(d_{jr}) x}{N_0}}\right), \tag{9}$$

where T_c is the chip duration, d_{tr} and d_{jr} are the respective transmitter-to-receiver and jammer-to-receiver distances, S_{tx} is the average signal power from the transmitter, N_0 is the ambient noise power, and $F(d)$ is a model for path loss. As long as the relative geometry (i.e., d_{tr} and d_{jr}) and the transmit power S_{tx} are known to the attacker, the PDR estimate in (7) can be used to define the impact metric as

$$\iota(\mathcal{S}, \boldsymbol{p}) = 1 - \Pi(a, d), \tag{10}$$

where $\boldsymbol{p} = (a, d)$ as previously described.

Stealth: We consider the use of a combination of PDR and estimated signal strength S_{rx} at the receiver to measure stealth $\varsigma(\mathcal{S}, \boldsymbol{p})$. For $\boldsymbol{p} = (a, d)$ as above, the jammer can estimate the received signal strength $S_{rx}(a, d)$ as

$$S_{rx}(a, d) = S_{tx} F(d_{tr}) + adF(d_{jr}) \tag{11}$$

on average. The jammer can then estimate the probability P_{det} of being detected as

$$P_{det}(a,d) = \left(1 + e^{-(S_{rx}(a,d) - \kappa\Pi(a,d))}\right)^{-1}, \quad (12)$$

where κ is a scaling parameter that determines an acceptable threshold to relate the acceptable PDR for a given received signal strength. We provide our derivation for this equation in Appendix **??**. The estimation of the probability of detection at another node is still an open problem so when better methods are discovered they can be used in place of (12). The detection probability can then be used to define the stealth metric as

$$\varsigma(\mathcal{S}, \boldsymbol{p}) = 1 - P_{det}(a,d), \quad (13)$$

where $\boldsymbol{p} = (a,d)$.

Expenditure: As previously discussed, we measure expenditure in terms of the energy of the jamming signal. This energy expenditure can be measured directly by the jamming device as the combination of signal amplitude and duty cycle as

$$\eta(\mathcal{S}, \boldsymbol{p}) = ad \quad (14)$$

where $\boldsymbol{p} = (a,d)$.

C. mSTIR-Jamming Design

In this section, we present an instance of the mSTIR-jamming attack presented in Section IV-C1 using the metrics given in Section V-B1 for a periodic jamming attack on an 802.15.4 system. As in Section IV, we break our attack description into the steps of observing, estimating, and optimizing.

1) Observation: As previously described in Section IV-A, we use the PDR as the observation metric to gauge the actual effect of the jamming attack on the sender-receiver system. At every time step k, the jammer thus observes a PDR value Π_k.

2) Estimation: The estimation step for the mSTIR-jamming attack relies on the ability for the attacker to estimate the transfer function $\mathcal{G}_k$ in each time step. We present an estimation method based on the same metrics used in Section V-B1, with a slight modifications to allow for adaptation. Accepting the fact that the physical layer model assumed in the computation of the chip error probability in (9) is not a perfect model, we introduce an additional parameter α_k to assist in fitting the model to the observations made by the attacker. We thus define the modified chip error, symbol error, and packet delivery probabilities respectively as

$$P_c^{\alpha}(x) = \frac{1}{2} Q\left(\sqrt{\frac{\alpha T_c F(d_{tr}) S_{tx} - T_c F(d_{jr}) x}{N_0}}\right), \quad (15)$$

$$P_s^{\alpha}(x) = \sum_{i=17}^{32} \binom{32}{i} P_c^{\alpha}(x)^i (1 - P_c^{\alpha}(x))^{32-i}, \quad (16)$$

$$\Pi^{\alpha}(a,d) = (1 - P_s^{\alpha}(a))^{nd} (1 - P_s^{\alpha}(0))^{n(1-d)}. \quad (17)$$

The estimated transfer function $\hat{\mathcal{G}}_k$ mapping $\boldsymbol{p}_k = (a_k, d_k)$ to $\boldsymbol{\phi}_k = \Pi^{\alpha_k}(a_k, d_k)$ is thus provided by (15), (16), and (17), parameterized by the tuning variable α_k which is updated at each time step.

Using this estimation model in the mSTIR-jamming attack algorithm to compute an estimate $\hat{\mathcal{G}}_k$ via α_k then allows for the error between the predicted PDR $\Pi^{\alpha_k}(a_k, d_k)$ and the observed PDR Π_k at time step k to be defined as

$$\text{error}\left(\Pi_k, \Pi^{\alpha_k}(a_k,d_k), (a_k,d_k)\right) = \Pi_k - \Pi^{\alpha_k}(a_k,d_k). \quad (18)$$

Using the above model, we also define the update function via α_{k+1} as a function of the previous α_k and ϵ_k as

$$\alpha_{k+1} = \begin{cases} \alpha_k(1+\epsilon_k), & \text{if } \epsilon_k > 0 \\ \alpha_k(1-\epsilon_k)^{-1}, & \text{otherwise.} \end{cases} \quad (19)$$

3) Optimization: Lastly, we present the optimization formulation used for the mSTIR-jamming. In choosing $\boldsymbol{p}_{k+1} = (a_{k+1}, d_{k+1})$, the algorithm imposes lower bound $\boldsymbol{p}_{min} = (a_{min}, d_{min})$ and upper bound $\boldsymbol{p}_{max} = (a_{max}, d_{max})$, where $a_{min} = 0$, $a_{max} > 0$ is the maximum jamming power, and d_{min} and d_{max} are specified bounds on the duty cycle (satisfying $0 \leq d_{min} < d_{max} \leq 1$). The objective function $\mu(\mathcal{S}, \boldsymbol{p}, \hat{\mathcal{G}}_{k+1}) = \mu(a,d)$ can be any combination of the metrics in $\boldsymbol{M}(\mathcal{S})$. We choose a linear combination of the impact, stealth, and expenditure metrics defined in Section IV-B, modified for use with the α_k parameterization, as

$$\mu(a,d) = \beta_\iota \iota^{\alpha_{k+1}}(a,d) - \beta_\eta \eta^{\alpha_{k+1}}(a,d) + \beta_\varsigma \varsigma^{\alpha_{k+1}}(a,d), \quad (20)$$

where β_ι, β_η, and β_ς are scalar weights to indicate the attacker's preference and to scale the metrics into comparable ranges. We further discuss the β parameters in Section VI.

D. tSTIR-Jamming Design

We next present an instance of the tSTIR-jamming attack presented in Section IV-C2 using the metrics in Section V-B1 for a similar periodic jamming attack on an 802.15.4 system.

Similar to the case of the model-based attack, we use the PDR as the observation metric of interest to measure the impact of the jamming attack on the sender-receiver system, again through the observation of a PDR value Π_k at each time step. In the tuning-based attack, the estimation and optimization steps are computationally simpler than in the model-based attack. Given the target value T as the PDR desired by the attacker, the error between the observed PDR Π_k and the target T is given by

$$\text{error}(\Pi_k, T) = \Pi_k - T. \quad (21)$$

For a given value of ρ, the δ_k decision parameter then allows for the one-step transition to be made from the previous parameters (a_k, d_k) to the subsequent parameters (a_{k+1}, d_{k+1}). In this case, the one-step transitions follow the logic given in the example in Figure 3.

VI. Implementation Results

In this section, we discuss our proof-of-concept implementation of mSTIR-jamming and tSTIR-jamming attacks described in Section V and present our performance results. The implementation of both algorithms uses the USRP2 software-defined radio platform [21] with the GNU Radio software package [22]. We use a previously developed implementation of the 802.15.4 protocol from UCLA [23], and we develop our customized jamming attack mechanisms to implement the attacks. We present the results individually and then provide a brief comparison of the two sets of results.

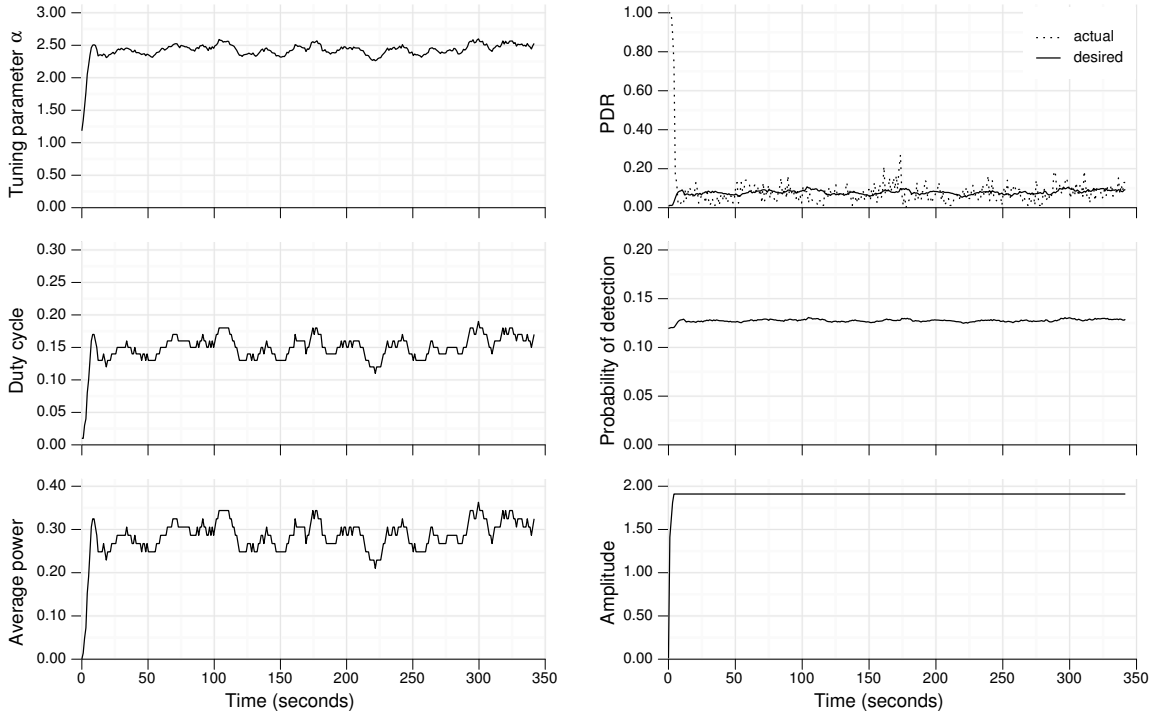


Fig. 4: The results for the mSTIR-jamming attack are shown, with $\beta_\iota = \beta_\varsigma = \beta_\eta = 50\%$ and induced measurement error $\xi = 10\%$.

A. mSTIR-Jamming Results

We implemented the mSTIR-jamming attack as described in Section V-C. As previously mentioned, the estimation process involved in observing the PDR Π_k in each time step, so we instead provide this statistic to the attacker via a direct line from the receiver. However, to test the performance of our STIR-jamming formulation with various levels of error, we induce errors into the observed PDR Π_k. In particular, the observation Π_k given to the attacker is equal to the true PDR perturbed by a uniform random variable in the interval $[-\xi, \xi]$, and we test several values of ξ in our implementation.

The results of one trial run of the mSTIR-jamming attack are shown in Figure 4. From the right-hand column, it can be seen that the jamming amplitude, probability of detection, and PDR goal rapidly stabilize; the actual achieved PDR fluctuates on a moment-to-moment basis but rarely exceeds 0.2, marking a successful attack. Probability of detection is also consistently low. The left-hand column shows the time evolution of the tuning parameter α, which does jitter a bit but is stable overall, and the duty cycle and average power, which track α precisely (the apparent additional variation is an artifact of the scale).

Figure 5a and Figure 5b demonstrate an STIR-jamming scenario where the jamming metrics are not equally weighted in (20). In both of these figures, the weights for the impact and stealth metrics are fixed at $\beta_\iota = \beta_\varsigma = 50\%$, while the weight for the expenditure metric is varied among $\beta_\eta = 25\%, 50\%, 75\%, 100\%$ during the four tests. When the expenditure is weighted heavily ($\beta_\eta = 75\%$ and $\beta_\eta = 100\%$), the resulting signal power and the corresponding jamming impact are both reduced. When the expenditure is lightly weighted ($\beta_\eta = 50\%$), the attack requires significantly higher resources but also yields significantly higher impact (i.e., lower PDR). We note that the attacker could also adapt these weights depending on energy availability (i.e., system health), change in jamming goals, or other dynamics, though such weight modification is beyond the scope of this work.

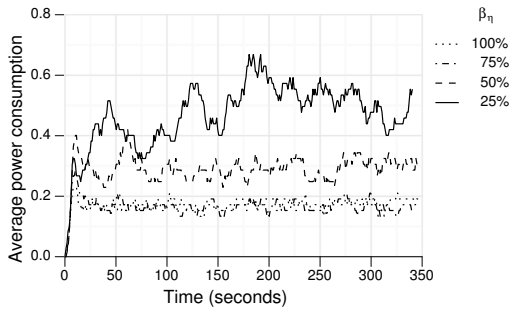
B. tSTIR-Jamming Results

We also demonstrate the effect of the induced error in the PDR measurements taken by the attacker. Figure 6 shows the error in PDR estimation as a function of the level of error ξ induced in the measurement. The worst case ($\xi = 50\%$) parameter tested demonstrates that the measurement error does affect the result but does not defeat the STIR-jamming attack.

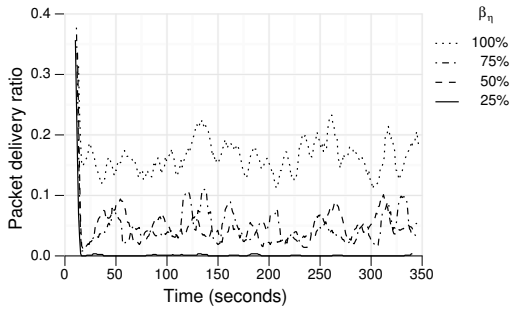
We implemented the tSTIR-jamming attack as described in Section V-D. Figure 7 illustrates the parameter trajectory through the state space for two target values, $T = 30\%$ PDR and $T = 70\%$ PDR. In both cases, we see that the attack parameters stabilize relatively quickly to a small region of the state space, although in the case of $T = 30\%$ PDR, the parameters jump to a different region after a short time.

The results of the tSTIR-jamming attack are shown in detail in Figure 8. This attacker can hold the system under attack to within about 15% of the PDR goal, although with a great deal of variance. Though this is generally undesirable for control systems, it may actually benefit the attacker in terms of reducing the chance of detection.

The plot of signal power in Figure 8 also sheds some light on the jump in the state space that was observed in Figure 7. After about 500 seconds during the $T = 30\%$ trial, something changed in the experimental environment (possibly a reduction of external network use) that allowed the sender-receiver system



(a)



(b)

Fig. 5: In (a) we show the average power consumed by the jammer, for four possible values of the expenditure parameter β_η, with β_ι and β_ς fixed at 50%. In this figure, higher values of β_η cause the jammer to reduce power consumption. In (b) We show the PDR achieved by the jammer, with the same settings as (a).

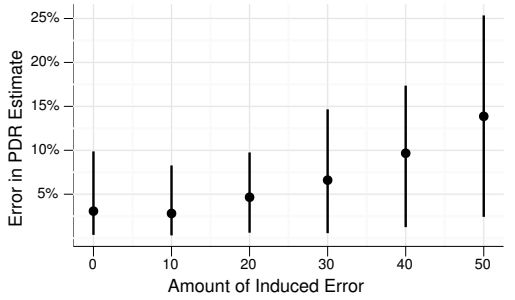


Fig. 6: The percent error in the attacker's estimated PDR compared to the actual PDR at the receiver is plotted as a function of the level of error ξ induced in the PDR measurements. The error bars show one standard deviation around the mean (with a minimum of zero).

to improve packet delivery. The attacker reacted to this external event by first increasing its duty cycle and then increasing the signal power. After only a few seconds, the attacker adapted to the changes and drove the PDR back to the desired target. This ability to adapt to dynamic environmental conditions is one of the major advantages of STIR-jamming attacks compared to using static parameters.

Comparing the mSTIR-jamming and tSTIR-jamming attacks is interesting, but since the trials were taken during different times of day with slightly different hardware configuration a direct comparison is not possible at this time. However, one interesting point of comparison is that the tuning-based attack uses considerably less power. Since the model-based attack re-optimizes the power level and duty cycle at every time step, it often leads to very loud pulses with power near the maximum, whereas the tuning-based attack slowly changes the power level only when the benefit of doing so is sufficiently large. Another trend we observed is that the model-based attack tends to be more stable than the tuning-based attack, as the estimate of the system transfer function converges relatively quickly with considerably less variance. Further comparison of the different attack types and investigation of the stability of the attacks are beyond the scope of this work and are left for future consideration.

VII. Conclusions

This paper introduces a framework for adaptation in wireless network attackers. This basic framework allows an attacker to listen to and infer information about a legitimate system using commodity hardware and adapt its attack to find more robust attack models. To prove the value of this concept we introduce self-tuned, inference-based, real-time jamming or *STIR-jamming*. This attack allows for jamming attack parameters to be tuned real-time to optimize an attacker's *impact*, *stealth*, or *expenditure*. This is accomplished by continually observing the system under attack, estimating the impact of jamming, and using this information to optimize its attacks. We show two proof of concept implementations for this attack, mSTIR-jamming which uses a rough channel model to optimize over possible attack parameters and tSTIR-jamming which searches for optimal attacks by taking small search steps in the parameter space. We implement proof of concept versions of these attacks which are able to find stable attack parameter locations to degrade an 802.15.4 links performance with high efficiency and low detectability.

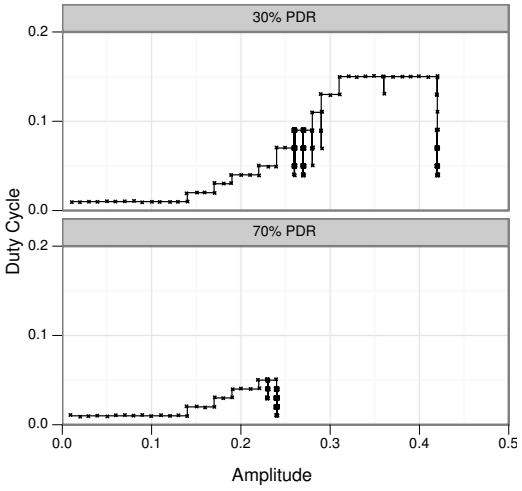


Fig. 7: The time evolution of control parameters for the tSTIR-jamming attack is shown, with two different PDR targets. The boxes are jittered slightly to reveal where the search stabilized: denser blobs indicate longer dwell times.

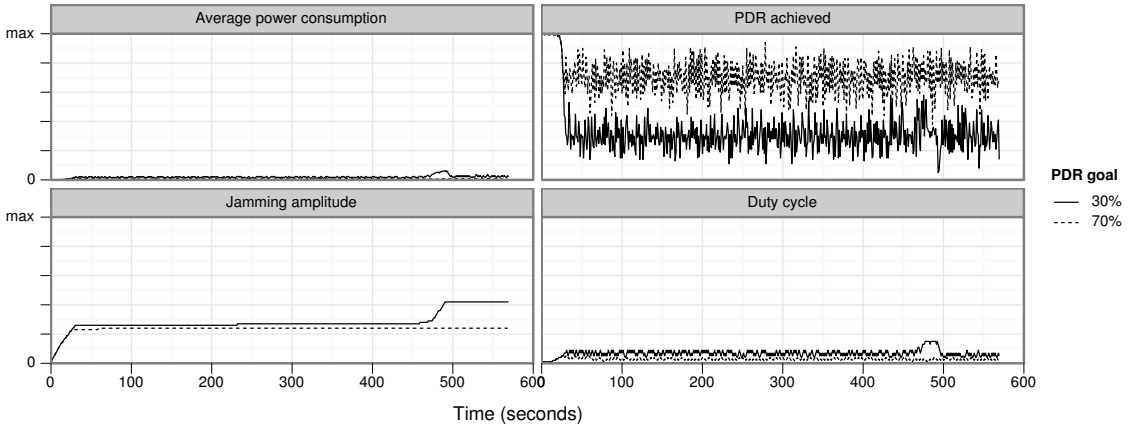


Fig. 8: The performance of a tSTIR-jammer is shown for a goal of 30% and 70%.

REFERENCES

[1] D. J. Torrieri, *Principles of Secure Communication Systems*, 2nd ed. Boston: Artech House, 1992.

[2] K. Pelechrinis, M. Iliofotou, and S. Krishnamurthy, "Denial of service attacks in wireless networks: The case of jammers," *IEEE/ACM IEEE Communication Surveys and Tutorials*, vol. 13, no. 2, pp. 245–257, May 2011.

[3] A. Molisch, *Wireless Communications*. John Wiley & Sons, Inc., 2005.

[4] M. Çakıroğlu and A. T. Özcerit, "Jamming detection mechanisms for wireless sensor networks," in *Proc. 3rd International Conference on Scalable Information Systems (InfoScale'08)*, Vico Equense, Italy, 2008, pp. 1–8.

[5] J. T. Chiang and Y.-C. Hu, "Cross-layer jamming detection and mitigation in wireless broadcast networks," in *Proc. 13th Annual ACM International Conference on Mobile Computing and Networking (MobiCom'07)*, Montréal, Québec, Canada, 2007, pp. 346–349.

[6] M. Raya, I. Aad, J.-P. Hubaux, and A. El Fawal, "DOMINO: Detecting MAC layer greedy behavior in IEEE 802.11 hotspots," *IEEE Transactions on Mobile Computing*, vol. 5, no. 12, pp. 1691–1705, Dec. 2006.

[7] W. Xu, T. Wood, W. Trappe, and Y. Zhang, "Channel surfing and spatial retreats: Defenses against wireless denial of service," in *Proc. of the ACM Workshop on Wireless Security*, Philadelphia, PA, Oct. 2004.

[8] P. Tague, S. Nabar, J. A. Ritcey, and R. Poovendran, "Jamming-aware traffic allocation for multiple-path routing using portfolio selection," *IEEE/ACM Transactions on Networking*, vol. 19, no. 1, pp. 184–194, Feb. 2011.

[9] D. J. Thuente and M. Acharya, "Intelligent jamming in wireless networks with applications to 802.11b and other networks," in *Proc. 25th IEEE Communications Society Military Communications Conference (MILCOM'06)*, Washington, DC, Oct. 2006, pp. 1–7.

[10] M. Wilhelm, I. Martinovic, J. Schmitt, and V. Lenders, "Reactive jamming in wireless networks: How realistic is the threat?" in *Proc. 4th ACM Conference on Wireless Network Security*, Hamburg, Germany, Jun. 2011.

[11] B. DeBruhl and P. Tague, "Mitigation of periodic jamming in a spread spectrum system by adaptive filter selection," in *2nd International Conf. on Pervasive and Embedded Computing and Communication Systems (PECCS)*, Rome, Italy, Feb. 2012.

[12] K. Pelechrinis, C. Koufogiannakis, and S. V. Krishnamurthy, "Gaming the jammer: Is frequency hopping effective?" in *Proc. 7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt'09)*, Seoul, Korea, Jun. 2009.

[13] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proc. ACM 6th International Symposium on Mobile Ad Hoc Networking and Computing*, Urbana-Champaign, IL, USA, May 2005, pp. 46–57.

[14] H. Liu, Z. Liu, Y. Chen, and W. Xu, "Localizing multiple jamming attackers in wireless networks," in *Proc. of Int'l Conf. on Distributed Computing Systems*, 2011.

[15] P. Tague, "Improving anti-jamming capability and increasing jamming impact with mobility control," in *6th IEEE International Workshop on Wireless and Sensor Networks Security (WSNS)*, Nov. 2010.

[16] B. Awerbuch, A. Richa, and C. Scheideler, "A jamming-resistant mac protocol for single-hop wireless networks," in *Proc. of the 27th ACM symposium on Principles of distributed computing*, Toronto, Canada, 2008.

[17] P. Tague, D. Slater, G. Noubir, and R. Poovendran, "Linear programming models for jamming attacks on network traffic flows," in *Proc. 6th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt'08)*, Berlin, Germany, Apr. 2008, pp. 207–216.

[18] F. Lewis, S. Jagannathan, and A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. Taylor and Francis, 1999.

[19] "IEEE 802.15.4-2006," http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf.

[20] B. DeBruhl and P. Tague, "Digital filter design for jamming mitigation in 802.15.4 communication," in *Proc. International Conference on Computer communications and Networks*, Maui, Hawaii, Aug. 2011.

[21] "Ettus research LLC," http://www.ettus.com/.

[22] "GNU radio," http://gnuradio.org/.

[23] T. Schmid, O. Sekkat, and M. Srivastava, "An experimental study of network performance impact of increased latency in software defined radios," in *Proc. 2nd ACM workshop on Wireless network testbeds, experimental evaluation and characterization*, Montreal, Quebec, Canada, 2007.

APPENDIX

To compute the probability of detection $\mathbf{P_{det}}$, we assume that the defending system will implement a detection strategy which checks for consistency in received signal strength (RSS) and packet delivery ratio (PDR) [13]. Originally, this work divided the RSS v. PDR plane into a jammed region and non-jammed region. We propose doing this with the linear boundary defined by

$$b = S_{rx}(a, d) - \kappa \Pi(a, d), \tag{22}$$

where κ is a scaling parameter, $\Pi(a, d)$ is the estimated packet delivery ratio with jamming power a and duty cycle d, and $S_{rx}(a, d)$ is the estimated received signal strength defined as

$$S_{rx}(a, d) = S_{tx}F(d_{tr}) + adF(d_{tr}), \tag{23}$$

where $F(\cdot)$ computes the fading coefficient at the given transmitter-to-receiver and jammer-to-receiver distances d_{tr} and d_{jr}, respectively.

Once we define a boundary b splitting the RSS v. PDR plane into a jammed and not jammed region, we propose using a sigmoid function [18] with the boundary defined by b to determine the probability of being in jammed region which triggers detection. This function is defined as

$$P_{det}(a, d) = \left(1 + e^{-b}\right)^{-1}, \tag{24}$$

which takes values on the interval $[0, 1]$ with higher values indicating greater probability of detection.