Detection of Reactive Jamming in DSSS-based Wireless Networks

Domenico Giustiniano[*], Vincent Lenders[†], Jens B. Schmitt[‡], Michael Spuhler[*], and Matthias Wilhelm[‡]
[*]ETH Zürich, Switzerland
[†]armasuisse, Switzerland
[‡]TU Kaiserslautern, Germany
domenico.giustiniano@tik.ee.ethz.ch, vincent.lenders@armasuisse.ch,
{jschmitt,wilhelm}@cs.uni-kl.de, spuhlemi@student.ethz.ch

ABSTRACT

We propose a novel approach to detect reactive jammers in direct sequence spread spectrum (DSSS) wireless networks. The key idea is to use the chip error rate of the first few jamming-free symbols at the DSSS demodulator during the signal synchronization phase of regular packet reception to estimate the probability of successful packet delivery. If the estimated probability is significantly higher than the actual packet delivery ratio, we declare jamming. As a proof of concept, we implement a prototype in a network of three USRP software-defined radios (transmitter, receiver, and jammer) and evaluate the feasibility, responsiveness, and accuracy of our approach in a controlled lab environment. Our experiments with IEEE 802.15.4 DSSS-based communication show that for links with a jamming-free packet delivery probability above 0.5, the false positive and negative detection rates remain below 5 %.

Categories and Subject Descriptors

C.2.0 [**Computer Communication Networks**]: General—*Security and protection (e.g., firewalls)*

Keywords

Jamming detection; reactive jamming; 802.15.4; DSSS

1. INTRODUCTION

Wireless networks are built upon a shared medium, which makes them vulnerable to jamming attacks. Jamming attacks are accomplished by emitting interfering RF signals that do not adhere to the rules of an underlying MAC protocol [17]. When such jamming signals interfere with the transmissions of legitimate transmitters at the receiver, the signals collide and render the originally transmitted data signals uninterpretable at the receiver.

In contrast to traditional security primitives such as authentication, confidentiality, or integrity that can be addressed with the application of cryptographic techniques, jamming attacks cannot be entirely fended off by conventional security mechanisms. While spread spectrum communication techniques are able to mitigate the effect of narrowband sources of interference, a jammer can always disturb the communication by emitting broadband signals that exceed the power of legitimate signals at the receiver.

Jammers may employ a wide range of strategies to disturb wireless communication [3, 8, 9, 16, 17]. Among these existing strategies, *reactive* jammers have been shown to be not only the hardest to detect, but also the most energy-efficient approach, making them a serious threat in wireless networks. In addition, [15] demonstrated that reactive jammers can be implemented on inexpensive COTS platforms such as the USRP2 from Ettus Research, and that reactive jamming can be triggered selectively on any field of the packet header, making them a *realistic threat* for wireless communication.

Since jamming cannot be prevented by design, it is important to understand how it works and, in turn, how to detect its presence. This paper proposes a novel method to detect reactive jamming in direct sequence spread spectrum (DSSS) systems. In DSSS systems, bits or symbols at the transmitter are spread to higher-order chip sequences. To detect the presence of jamming, our approach accounts for chip errors in the preamble at the output of the demodulator to model the probability of packet losses. If the experienced packet loss rate exceeds the one estimated from chip errors in the preamble, a reactive jammer is likely jamming parts of the packet, and we thus declare jamming. Since the preamble of a packet represents the very first chips being sent for synchronization purposes, it significantly reduces the probability that a reactive jammer will jam these chips because it requires very fast reactivity, low signal propagation delays, and prevents a jammer from making jamming decisions according to physical, MAC, or payload based rules [15].

At the core of our detection scheme is an accurate packet delivery estimation model based on chip errors in the preamble, which is independent of the received signal strength (RSS) that is being used by existing detection schemes [14, 17]. Our approach does not require any modification to the communication system or standard and works even when the reactive jammer targets the synchronization phase of packet transmissions. We implement a reactive jamming detector

WiSec '13, April 17–19, 2013, Budapest, Hungary.

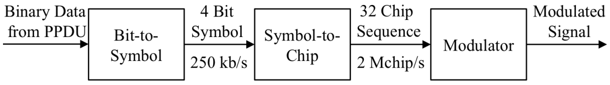

Figure 1: DSSS modulation in IEEE 802.15.4.

for IEEE 802.15.4 on the USRP software-defined radio platform from Ettus Research and we evaluate its performance in a controlled lab environment with the reactive jammer from [15]. Our results show that our detection scheme is able to accurately detect reactive jammers on fading wireless links with a jamming-free packet delivery probability above 0.5. The false positive and negative detection rates remain below 5 % for these links.

The rest of this paper is organized as follows. In the next section, we briefly review important aspects of the IEEE 802.15.4 standard, introduce the attacker model, and describe the experimental setup used in the evaluation. Section 3 explores the feasibility to model the packet delivery with limited information from chip errors in the preamble. In Section 4, we introduce our jamming detection scheme. Section 5 covers the evaluation of the detection accuracy. Related work is discussed in Section 6, and Section 7 concludes the paper.

2. BACKGROUND AND ATTACKER MODEL

In this section, we briefly review important aspects of the IEEE 802.15.4 standard, introduce the attacker model and describe the experimental setup that we use for evaluation.

2.1 Background on IEEE 802.15.4

Our work on jamming detection focuses on direct sequence spread spectrum (DSSS) communication systems, and is practically demonstrated for the IEEE 802.15.4 standard [1]. IEEE 802.15.4 defines a 16-ary quasi-orthogonal DSSS modulation technique. This modulation spreads a low rate sequence of bits to a higher rate sequence of so-called chips in the following way: binary source data is divided into groups of 4 bits (referred to as *symbols*) and mapped to a quasi-orthogonal 32-chip pseudo-noise sequence $(b_0, b_1, b_2, b_3) \mapsto (c_0, c_1, \dots, c_{31})$, resulting in a chip rate of 2 MChips/s as shown in Figure 1. The effect of this spreading is an increased robustness against fading and in-band interference: DSSS systems can tolerate a certain number of chip errors and still receive symbols correctly.

Our proposed detection scheme relies on the fact that the packet error probability can be predicted accurately using the number of chip errors in the first few symbols in a packet. An IEEE 802.15.4 packet consists (as shown in Figure 2) of a physical layer header with a preamble sequence for symbol synchronization (eight `0` symbols), a start of frame delimiter (SFD; symbols `7` and `10`) and a frame length field indicating the duration of the frame, followed by a MAC protocol data unit (MPDU). The MPDU contains a MAC header, data payload, and ends with a frame check sequence (FCS) that is used to detect transmission errors. IEEE 802.15.4 does not mandate the use of error correction mechanisms, and any received packet with an incorrect FCS is hence discarded.

To receive a packet, the receiver first synchronizes with the preamble sequence to detect the symbol boundaries, i.e., the time instants when chip sequences start. This timing information is subsequently used to detect the SFD and frame length field. The rest of the signal is decoded using a correlator to map the received 32 chips back to symbols. The received chip sequence R may contain errors caused by fading or interference. It is compared to the 16 predefined chip sequences $C_i, i = 0, 1, \dots, 15$. The receiver chooses the best match, i.e., the C_i for which $h(R, C_i)$ is minimal, where $h(\cdot,\cdot)$ is the Hamming distance (number of positions containing different chips) between the two arguments. However, if too many chips are flipped, the expression $h(R, C_i)$ may be minimal for the wrong chip sequence C_i such that the receiver interprets the chip sequence as a wrong symbol.

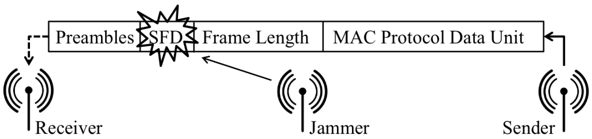

Figure 2: Reactive jamming: an attacker jams the start-of-frame delimiter (SFD) to disturb the synchronization of the packet at the receiver.

2.2 Attacker Model

We consider reactive jammers that aim to minimize their jamming duration to only a few symbols in order to remain undetected and to save energy. We assume a jammer that is able to sniff any symbol of the packet over the air in real-time and react with a jamming signal that flips selected bits at the receiver with high probability. An attacker may therefore pursue different reactive jamming strategies [15]. It may jam *(i)* the MPDU, *(ii)* the packet length field, *(iii)* the frame synchronization field (SFD), or *(iv)* the preamble of the packet. The first two strategies cause packet losses because of resulting FCS errors, while the last two strategies introduce synchronization failures, causing the entire packet to be missed by the receiver. Figure 2 illustrates jamming strategy *(iii)* that targets the SFD.

The jamming reaction time τ denotes the time difference between the arrival of the original signal and the jammer signal at the receiver. The minimal reaction time $\tau_{\min}$ is bounded by the sum of the signal propagation delay between sender and jammer, the reaction delay of the jammer to process the incoming signal and to make a jamming decision, and the signal propagation delay between jammer and receiver. It is therefore safe to assume that the minimum reaction time $\tau_{\min}$ is greater than the duration of one symbol (e.g., 16 μs in IEEE 802.15.4). Otherwise it would not be possible to assess the channel state prior to jamming. In fact, [15] showed that the reaction time of a realistic jammer is significantly larger than this minimum reaction delay because of the inherent hard- and software delays to detect, demodulate, process, and trigger jamming signals according to particular jamming rules. While it might be technically feasible to implement reactive devices with lower reaction delays than the duration of one symbol duration (for example by using simple power detectors with analog parts [7,11]), reactive jammers of that kind are not able to use the semantics of the signals to perform smart jamming decisions like jamming only selected packets according to specific rules (e.g., matching packet modulation or header properties).

2.3 Experimental Setup

We rely on measurements to study the performance of packet delivery models and to evaluate the proposed jamming detection. Our experimental setting considers point-to-point data transmissions in a network consisting of three nodes: sender, receiver, and jammer. Our experiments are based on a software-based implementation of IEEE 802.15.4. As hardware platform, we use the USRP software-defined radio from Ettus Research. For the software, we use a slightly optimized version of the UCLA IEEE 802.15.4 implementation [12] that runs on the GNU Radio framework. We have performed multiple tests in indoor lab environments, which are referred to as *cable*, *static line-of-sight*, *static non-line-of-sight*, and *mobile*. In the *cable* experiments, sender and receiver are connected by a shielded 60 cm coaxial cable with a 30 dB attenuator. In the *static* experiments, a stationary sender and receiver communicate using omni-directional antennas. The *mobile* experiments are similar to the static scenario except that the sender is kept stationary while the receiver is moving. The receiver is placed on a cart and moved at a constant speed of maximum $v = 1$ cm/s away from, and back towards, the sender.

In each experiment run, 40,000 packets of 26 bytes length are sent during 40 seconds from the transmitter to the receiver at constant rate. Various link conditions in the cable and static experiment runs are obtained by adjusting the transmit power and by changing the position of nodes. The true packet delivery ratio (PDR) at time t is calculated by averaging the number of received packets over a window of 100 packets centered around t. A window size of 100 packets assures that the true PDR is calculated over a time window that is smaller than the channel coherence time when moving the receiver at maximum $v = 1$ cm/s and at a frequency of 2.4 GHz.[1] Note that the mobility experiments have a relatively low node speed of maximum 1 cm/s for the sake of determining the true PDR. We intentionally kept the node mobility low such that the channel coherence time is larger than the window size of 100 packets that are used to calculate the true PDR. Our results are thus relatively conservative with respect to mobility.

As a jammer, we use the reactive jammer from [15], which also runs on the USRP2 platform. It can be configured to jam according to strategies *(i)* to *(iv)*. The detection and decision logic are implemented on the FPGA of the USRP2, resulting in a minimal reaction delay of $\tau_{\min} = 19\,\mu$s.

3. CHIP ERROR BASED PDR MODEL

Our jamming detection technique relies on a statistical model of packet delivery from chip errors in the first few symbols of the preamble [13]. This section provides experimental results that show that the packet delivery ratio in DSSS-based wireless networks can be modeled accurately using such limited information. We further show that our model significantly outperforms RSS-based PDR estimators, which constitute the basis of current jamming detection schemes.

Our statistical model exploits the strong correlation between DSSS chip errors in the preamble, observed at the output of the demodulator of the receiver, and the experienced packet delivery ratio. Figure 3 shows this correlation for four experiments in different environments (cable, static line-of-sight, static non-line-of-sight, and mobile). As we can see, the average number of chip errors per preamble symbol is highly correlated in the entire range of PDRs as indicated by a Pearson correlation coefficient of -0.965.[2] Note that the average number of chip errors does not exceed 4 because the receiver we used makes hard decoding on preamble symbols with this threshold. We varied the hard decoding threshold for the preambles to values ranging from 1 to 6 in order to evaluate the effect on the distribution: while the distribution gets shifted when changing this threshold, the strong correlation still remains.

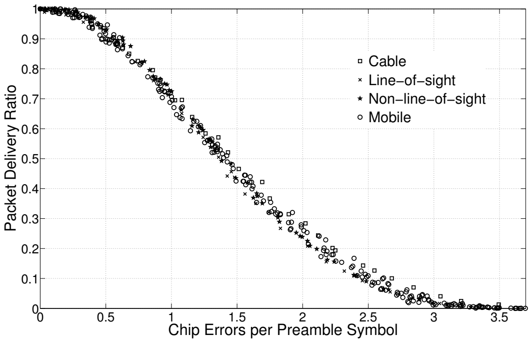

Figure 3: Correlation between average chip errors in the preamble and packet delivery ratio.

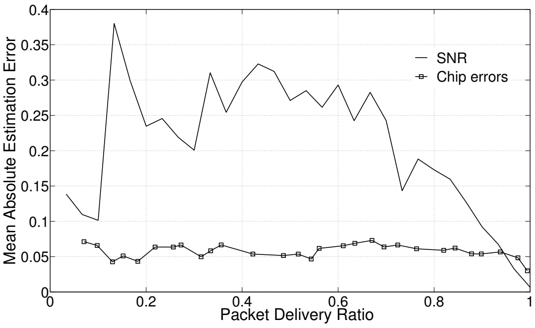

Figure 4: Comparison of mean absolute estimation error of the packet delivery ratio for a model that relies on the average number of chip errors in the preamble versus a model that relies on the RSS.

This correlation is well suited to predict the PDR, as shown in Figure 4 for the case of *mobile* scenarios. The figure compares the mean absolute packet delivery estimation error of our model that relies on the chip errors in the preamble to a model based on the signal-to-noise ratio (SNR) [5]. Our chip error based model estimates the PDR using a regression with a polynomial function $g_{\text{CER}}(p)$ that has a root mean square (RMS) error below 3 % across all considered environments. This regression function $g_{\text{CER}}(p)$ maps the average number of chip errors per preamble symbol p to the respective PDR. The polynomial function with the smallest degree is of the form

$$g_{\text{CER}}(p) = a_0\,p^5 + a_1\,p^4 + a_2\,p^3 + a_3\,p^2 + a_4\,p + a_5,$$

where $a_0 = 0.016$, $a_1 = -0.33$, $a_2 = 2.41$, $a_3 = -7.26$, $a_4 = 8.83$, $a_5 = -3.24$. Similarly, the SNR-based model estimates the PDR also using a polynomial regression function, but fitted to the empirical SNR–PDR distribution. Selecting

[1] The coherence time is the time duration for which the channel impulse response is considered to be stationary and is approximately $\frac{1}{4D}$, where D is the Doppler spread.

[2] Values close to 0 indicate a low correlation and values close to ±1 represent a high linear dependence of two variables.

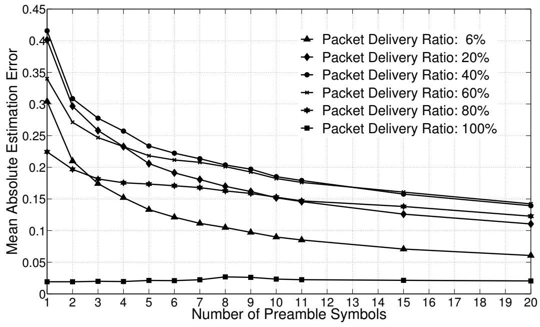

Figure 5: Mean absolute estimation error of the packet delivery for various PDRs and different number of preamble symbols in the model.

the polynomial function $g_{\mathrm{SNR}}(\mathrm{SNR})$ as the one with RMS error below 3 % across all considered environments, we have:

$$g_{\mathrm{SNR}}(\mathrm{SNR}) = b_0\ \mathrm{SNR} - b_1,$$

where SNR is the signal-to-noise-ratio expressed in dB and the coefficients are $b_0 = 0.12$ and $b_1 = -1.7$.

Using the above models, Figure 4 shows that the mean absolute PDR estimation error is significantly lower for the chip error based model across almost the entire range of delivery ratios (for PDR > 0.95, the absolute error is slightly lower for the SNR-based model). Existing jamming detection schemes that rely on the RSS thus suffer inherently from this estimation error. The fact that RSS-based models of packet delivery are generally not very accurate in real-world wireless networks has also been reported previously in the literature [2, 13].

As we cannot control the reaction time τ of the adversary, it is crucial that the proposed model of packet delivery manages to estimate with as few preamble symbols as possible. Figure 5 evaluates the mean absolute estimation error of the packet delivery versus a varying number of preamble symbols used in the estimation. Preamble symbols can be accumulated over multiple transmissions, i.e., they do no have to be from the same packet, hence enabling a number of preamble symbols larger than 8. As we can see, the error quickly converges, hence providing a useful estimator even for a model that accounts for just a few symbols.

4. JAMMING DETECTION

In this section, we describe our jamming detection scheme using the packet delivery model of the previous section. The basic idea is to continuously monitor the traffic over a link and determine two metrics. The first metric is the *observed* packet delivery ratio $\mathrm{PDR_o}(t)$ at time t, which is calculated by counting the ratio of correctly received packets over the total number of transmitted packets in a sliding observation window:

$$\mathrm{PDR_o}(t) = \frac{\#\text{ of correct packets in } [t-W, t]}{\#\text{ of transmitted packets in } [t-W, t]}$$

To determine the number of correctly received packets the receiver checks the FCS of all received packets and, if correct, increments a counter. Determining the total number of transmitted packets at the receiver must take into account that a reactive jammer might successfully jam all SFDs of the transmitted packets, thus preventing any successful packet synchronization at the receiver. The only reliable information source is therefore the preamble when the reactive jammer has not yet started. The receiver counts the received preamble symbols and increments its counter of transmitted packets when at least one symbol 0 is detected within a sliding time window of the size of the preamble. The observed $\mathrm{PDR_o}$ should be calculated over a time window that is shorter than the channel coherence time but sufficiently long to capture enough packets to derive a statistically relevant average. In this work, we fix this window size to $W = 100$ ms, corresponding to roughly 100 data packets at the actual transmit rate of the sender.

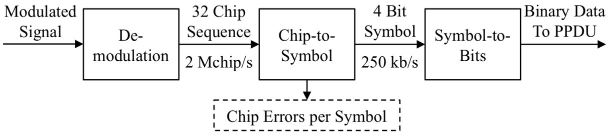

Figure 6: Chip errors in the preamble symbols are determined during the chip-to-symbol mapping of the receiver.

The second metric is an *estimated* PDR based on the preamble chip errors. As shown in Figure 6, the IEEE 802.15.4 receiver demodulates the incoming signal and attempts to map the demodulated 32-chip sequence to a known symbol. When the receiver is not synchronized yet, it attempts to map the incoming sequences to symbol 0. This is done with hard-decision decoding, that is, the receiver checks if the Hamming distance of the received chip sequence is smaller than a threshold value. This threshold value (4 for our receiver) is usually significantly below the mean Hamming distance of the symbols to prevent the receiver to synchronize on noise. To calculate a statistically relevant chip error rate, the receiver averages the Hamming distances of multiple preamble symbols. We point out again that the calculated average is not constrained to include only preamble symbols from a single packet. For example, when a jammer is reacting very quickly and jams symbols at positions 2 to 8 in the preamble, the received chip sequences 2 to 8 are not accounted for the statistics because, due to chip flipping, their Hamming distance becomes greater than the hard decoding threshold and these symbols are hence not interpreted as 0. Similarly, when the link conditions are poor, a receiver might miss multiple symbols per preamble. After receiving enough 0 symbols, the estimated PDR is calculated as

$$\mathrm{PDR_e} = g_{\mathrm{CER}}\left(\frac{\sum_{j=1}^{|\mathcal{S}|} h(R_j, C_0)}{|\mathcal{S}|}\right),$$

where R_j is the jth received 32-bit chip sequence that has been interpreted as a 0 with hard decoding, C_0 is the chip sequence of symbol 0, $h(\cdot,\cdot)$ is the Hamming distance, $\mathcal{S}$ is the set of received preamble symbols within a sliding window, and $g_{\mathrm{CER}}(\cdot)$ is a function that models the empirical distribution of the PDR versus chip errors per preamble symbol as defined in Section 3. To assure that the set $\mathcal{S}$ is large enough irrespectively of the channel quality and the jammer reaction time, we do not determine $\mathrm{PDR_e}$ based on a fixed sliding time window but rather on a fixed set size. We have set this size to $|\mathcal{S}| = 10$ (i.e., 10 symbols 0) in our work as it has proven to provide a reasonable tradeoff between accuracy and reactivity of jamming detection.

We define a hypothesis test based on the relative difference

Δ between the expected and observed PDR:

$$\Delta = \frac{\mathrm{PDR_e} - \mathrm{PDR_o}}{\mathrm{PDR_e}}.$$

Let us define the null hypothesis H_0 and the alternative hypothesis H_1 as

$$H_0 : \text{"Normal transmission,"}$$
$$H_1 : \text{"Jammed transmission."}$$

Then the test is as follows:

$$\text{accept } H_1, \text{ if } \Delta > \epsilon,$$
$$\text{stay with } H_0, \text{ if } \Delta \leq \epsilon,$$

where ϵ represents a tolerance level which directly affects the false positive and false negative detection rates. For small tolerance level values ϵ, the jamming detection is more sensitive, but at the price of higher false negative rates. For higher values of ϵ, the false negative rates may be reduced, but, in turn, at the price of higher false positive rates. To determine a good value for ϵ, we perform a maximum likelihood estimation using our measurements as follows. Let $\Lambda(\epsilon)$ be the sum of the false positive and false negative detection rates for a given PDR:

$$\Lambda(\epsilon) = P(H_0 \mid \text{jammer on}) + P(H_1 \mid \text{jammer off}).$$

Through exhaustive search using our measurements, we perform a maximum likelihood estimation that minimizes $\Lambda(\epsilon)$ for any value of $\epsilon > 0$ and PDR $\in [0, 1]$. The result is that $\Lambda(\epsilon)$ is minimized when $\epsilon = 0.5$ for all PDR $\in [0, 1]$. This agrees with the theoretical expectation that the error threshold lies in the geometric center of the decision region.

5. EVALUATION

Our evaluation focuses on quantifying the detection performance in terms of false positives and false negatives under realistic wireless fading channel conditions. For this purpose, we test our detection algorithm on software-defined radios with real traffic over the air.

5.1 Evaluated Jammer

For the performance evaluation, we consider a reactive jammer that jams all packets. We further study the robustness of our approach under the condition that the jammer does not succeed to jam all packets, but is still able to destroy 90 % of the packets. Figure 7 shows the impact of these two forms of reactive jamming on the correlation between the PDR and the chip errors in preamble symbols for $|\mathcal{S}| = 10$. The dark curve in the middle of the figure is the regression curve $\mathrm{g_{CER}}(\cdot)$ derived previously. As expected, if the transmission is not affected by the jammer, the points are spread around this curve. If the jammer is active, the position of these points changes and the strong correlation between the observed PDR and chip error distribution fades away. The points then coincide with the horizontal axis (for the 100% reactive jammer) or are spread around this axis (for the 90% reactive jammer). Another finding is that the detection of reactive jammers that successfully jam 90 % of the packets is more challenging as the PDR gets poorer, because the Euclidean distance between the PDR in presence and absence of jamming is reduced. In the region with higher number of chip errors per preamble symbol, this may be erroneously interpreted as links with poor quality (e.g., where losses are caused by a low SNR).

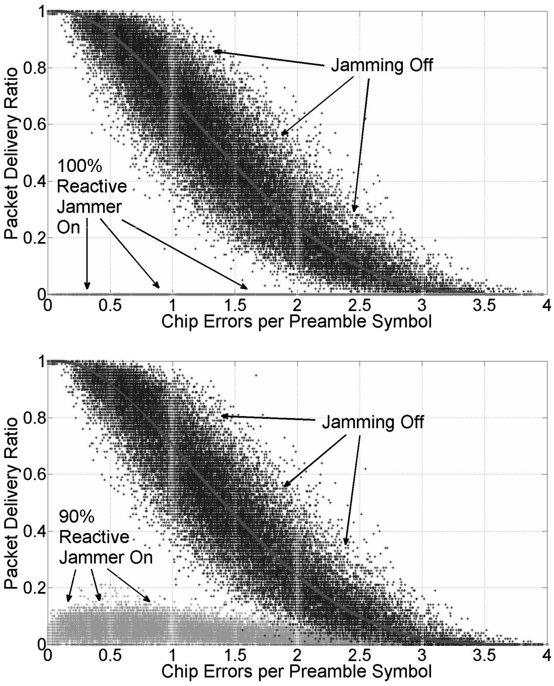

Figure 7: Impact of jamming on the correlation between the PDR and the preamble chip errors. Above we have the case of jamming all packets, below the one of jamming 90 % of packets.

5.2 Detection Performance

The false positive and false negative rates are evaluated in Figure 8. The jammer is configured in these experiments to react and hit the SFD of transmitted packets. This jamming strategy is of particular interest because packet synchronization fails and existing detection mechanisms are not able to cope with this type of reactive jamming. Both the false negative and positive error rates have probabilities below 5 % for links ranging from perfect to a $\mathrm{PDR_e}$ of 0.5. Below a $\mathrm{PDR_e}$ of 0.35, the reactive jammer causes false negatives over 10 %, constantly increasing for worse links. The false positives rate stays very small as well for good links and exceeds the error threshold of 10 % for $\mathrm{PDR_e}$ below 0.35 and then increases similarly for worse link qualities.

This general observation of increasing false positive and false negative rates in poor link environments for the jamming scenario is because $\mathrm{PDR_o}$ and $\mathrm{PDR_e}$ tend to overlap. A $\mathrm{PDR_o}$ obtained in poor link environments is more difficult to assign to either a jammed poor link quality situation or an ordinary poor link quality state. However, it has to be considered that the benefit in detecting jammers in poor link qualities conditions is not that crucial because low quality links are generally not used by higher layer network and application protocols. For good links with $\mathrm{PDR_e} > 0.5$, an accurate jamming detection is more valuable. In this region, we measure that the reactive jammer has a false negative error rate below 5 %.

6. RELATED WORK

To the best of our knowledge, this work is the first to provide a jamming detection scheme that can cope with so-

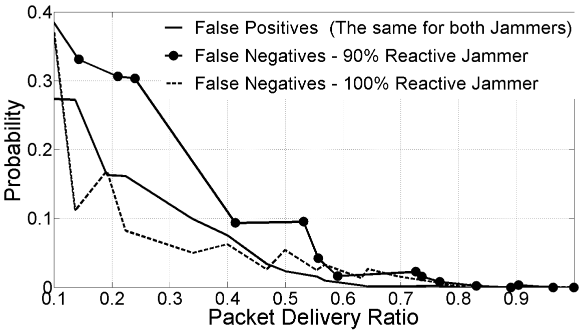

Figure 8: Performance evaluation of reactive jammer detection with respect to the false positive and false negative rates.

phisticated reactive jamming attacks targeting packet synchronization. Strasser et al. [14] propose a jamming detection scheme for sensor networks that enables a per-packet detection of reactive (single-bit) jamming. The main idea is to identify the cause of individual bit errors within a packet by analyzing the RSS of each received bit in the packet. A limitation of this approach is that it relies on a successful packet synchronization. Thus it is not able to detect SFD jamming attacks because decoded MPDU symbols are unavailable at the receiver due to the synchronization prevention. A further challenge is to localize bit errors in a packet. The authors propose to either use *a priori* knowledge of the bit stream sent, the use of error detecting/correcting codes, with drawbacks such as additional overhead and transmission costs, or to acquire the error position based on limited, short-range sensor node wiring in the form of wired node chains. Because our approach is not relying on error positions in a packet, it does not suffer from these restrictions.

Xu et al. [17] propose the usage of the PDR along with either RSS or device location information as a consistency check for proactive and reactive jamming detection. In the first case, jamming is detected if the PDR is low although the RSS is high. In the second case, the PDR is low although the sender–receiver distance is small. Unlike our work, these techniques are not able to detect reactive jamming that targets the physical layer header, or jammers that affect only a few bits per packet.

Xuan et al. [18] describe a method to identify so-called *trigger nodes* that are in the vicinity of reactive jammers and thus trigger jamming. This information is subsequently used to exclude such nodes and route around jammed areas. The authors assume that the detection of jamming on a per-packet level is feasible without error, such that the challenges treated in this work are avoided.

Chiang and Hu [4] leverage the properties of orthogonal spreading codes to achieve jamming detection and mitigation. In contrast to our work, their mode of operation is CDMA and the codes are long and confidential such that the attacker cannot interfere with all transmissions. We assume DSSS systems with public (or compromised) codes.

Finally, Qin et al. [10] suggest that the chip error rate might be a better channel quality indicator than signal power based metrics, particularly in the presence of interference. However they do not propose any estimator nor do they evaluate the feasibility to estimate the PDR from chip error measurements as we do in this work. CEPS [6] models the PDR from chip errors in the payload of successfully received packets. In contrast, we model the PDR from chip error measurements in the synchronization phase at the preamble and show that this information is already sufficient for detecting reactive jamming.

7. CONCLUSION

We have proposed a novel approach to detect sophisticated reactive jamming attacks that may target any part of a packet transmission. Our approach is based on chip errors of a few initial symbols during the synchronization phase of a packet transmission in order to predict the link packet delivery, which makes it suitable to even detect jammers that target the physical layer header of packets. Our experiments under real-world channel conditions showed that it is possible to predict the packet delivery accurately using the chip error rate derived from just a few preamble symbols. We further showed that we can detect reactive jammers with a false negative rate below 5 % for PDRs over 0.5.

8. REFERENCES

[1] IEEE Standard 802 Part 15.4: Wireless medium access control and physical layer specifications for low-rate WPANs.

[2] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level measurements from an 802.11b mesh network. In *Proc. of ACM SIGCOMM '04*, pages 121–132, Aug. 2004.

[3] M. Çakiroglu and A. T. Özcerit. Jamming detection mechanisms for wireless sensor networks. In *Proc. of ICST InfoScale '08*, pages 1–8, June 2008.

[4] J. T. Chiang and Y.-C. Hu. Cross-layer jamming detection and mitigation in wireless broadcast networks. *IEEE/ACM Trans. Netw.*, 19(1):286–298, Jan. 2011.

[5] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Predictable 802.11 packet delivery from wireless channel measurements. *Proc. of ACM SIGCOMM '10*, pages 159–170, Aug. 2010.

[6] P. Heinzer, V. Lenders, and F. Legendre. Fast and accurate packet delivery estimation based on DSSS chip errors. In *Proc. of IEEE INFOCOM '12*, pages 2916–2920, Mar. 2012.

[7] M. Kuhn, H. Luecken, and N. O. Tippenhauer. UWB impulse radio based distance bounding. In *Proc. of WPNC '10*, pages 28–37, Mar. 2010.

[8] Y. W. Law, M. Palaniswami, L. V. Hoesel, J. Doumen, P. Hartel, and P. Havinga. Energy-efficient link-layer jamming attacks against wireless sensor network MAC protocols. *ACM Trans. Sensor Netw.*, 5(1):6:1–6:38, Feb. 2009.

[9] A. Proaño and L. Lazos. Packet-hiding methods for preventing selective jamming attacks. *IEEE Trans. Dependable Secure Comput.*, 9(1):101–114, Jan. 2012.

[10] Y. Qin, Z. He, and T. Voigt. Towards accurate and agile link quality estimation in wireless sensor networks. In *Proc. of IFIP Med-Hoc-Net '11*, pages 179–185, June 2011.

[11] K. B. Rasmussen and S. Čapkun. Realization of RF distance bounding. In *Proc. of USENIX Security '10*, pages 389–402, Aug. 2010.

[12] T. Schmid. GNU Radio 802.15.4 en- and decoding. Technical Report TR-UCLA-NESL-200609-06, UCLA NESL, Sept. 2006.

[13] M. Spuhler, V. Lenders, and D. Giustiniano. BLITZ: Wireless link quality estimation in the dark. In *Proc. of EWSN '13*, pages 99–114, Feb. 2013.

[14] M. Strasser, B. Danev, and S. Čapkun. Detection of reactive jamming in sensor networks. *ACM Trans. Sensor Netw.*, 7(2):16:1–16:29, Aug. 2010.

[15] M. Wilhelm, I. Martinovic, J. B. Schmitt, and V. Lenders. Reactive jamming in wireless networks: How realistic is the threat? In *Proc. of ACM WiSec '11*, pages 47–52, June 2011.

[16] A. D. Wood, J. A. Stankovic, and G. Zhou. DEEJAM: Defeating energy-efficient jamming in IEEE 802.15.4-based wireless networks. In *Proc. of IEEE SECON '07*, pages 60–69, June 2007.

[17] W. Xu, W. Trappe, Y. Zhang, and T. Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In *Proc. of ACM MobiHoc '05*, pages 46–57, May 2005.

[18] Y. Xuan, Y. Shen, N. P. Nguyen, and M. T. Thai. A trigger identification service for defending reactive jammers in WSN. *IEEE Trans. Mob. Comput.*, 11(5):793–806, May 2012.