# Detection of Reactive Jamming in DSSS-based Wireless Communications

Michael Spuhler, *Member, IEEE,* Domenico Giustiniano, *Member, IEEE,* Vincent Lenders, *Member, IEEE,*
Matthias Wilhelm, *Student Member, IEEE,* and Jens B. Schmitt

***Abstract*—Reactive jammers have been shown to be a serious threat for wireless communication. Despite this, it is difficult to detect their presence reliably. We propose a novel method to detect such sophisticated jammers in direct sequence spread spectrum (DSSS) wireless communication systems. The key idea is to extract statistics from the jamming-free symbols of the DSSS synchronizer to discern jammed packets from those lost due to bad channel conditions. Our contribution is twofold. First, we experimentally evaluate new empirical models utilizing the preamble symbols of IEEE 802.15.4 packets, thus enabling the accurate prediction of the packet delivery ratio (PDR). We show that the chip error rate-based metric is superior to metrics used in the literature, offering an accurate and reactive indicator of the true PDR. Our second contribution is the design and evaluation of a detection technique relying on this metric to detect reactive jammers. We build a software-defined radio testbed and show that our technique enables the error-free detection of reactive jammers that jam all packets on links with a PDR above 0.3. To the best of our knowledge, our detector is the first to detect reactive jamming attacks targeting the physical layer header of DSSS packets, and does not require any modifications of the wireless communication system.**

***Index Terms*—Jamming detection, reactive jamming, direct sequence spread spectrum, 802.15.4, chip errors.**

## I. Introduction

WIRELESS networks are built upon a shared medium, making them vulnerable to jamming attacks. Such attacks are accomplished by generating intentional RF interference that does not adhere to the conventions of an underlying MAC protocol [1]. Jamming signals interfere with the transmissions of legitimate transmitters at the receiver in the sense that the signals collide and render the originally transmitted data signals uninterpretable. In contrast to traditional security primitives such as authentication, confidentiality, or integrity that can be addressed with cryptographic techniques, jamming attacks targeting the availability cannot be fended off entirely by conventional security mechanisms. While spread spectrum communication techniques are able to mitigate the effect of narrowband interference, a jammer can always disturb the communication by emitting broadband signals that exceed the power of legitimate signals.

Jammers may employ a wide range of strategies to disturb wireless communications [1]–[5]. Among these existing strategies, *reactive* jammers that become active upon detection of transmissions over the channel have been shown not only to be the hardest to detect, but also the most energy-efficient approach, making them a serious threat in wireless networks. In addition, recent work [6] has demonstrated that reactive jammers can be implemented on inexpensive commercial off-the-shelf (COTS) platforms such as the USRP from Ettus Research, and that reactive jamming can be triggered selectively, for example, on any field of the packet header, making it a *realistic threat* for wireless communications.

Fundamentally, jamming cannot be prevented by design, hence 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 jammers in wireless communication systems. The key idea is to use information extracted from the first few jamming-free bits received during the signal synchronization phase of regular packet reception to discriminate jammed packets from packets that are lost due to natural causes such as bad channel conditions. This problem is known to be challenging in real-world environments [1], [7].

Our work targets direct sequence spread spectrum (DSSS) communication systems such as the one employed in the IEEE 802.15.4 standard. We take advantage of the fact that the first few jamming-free bits are known *a priori* because they constitute a fixed preamble intended for signal synchronization at the DSSS receiver. Since the packet preamble represents the start of the DSSS signal on the air, it is unlikely that a reactive jammer jams this part of the communication because it demands very high reactivity, low signal propagation delays, and it prevents a jammer from making smart jamming decisions according to physical, MAC, or payload based rules [6]. We therefore use these preamble symbols to estimate the link quality reliably and provide the following contributions:

- We compare empirical models that rely either on chip errors, symbol errors, or the received signal strength (RSS) in the preamble to predict the packet delivery ratio (PDR). We evaluate these models under different environmental link conditions using measurements in our software-defined radio testbed. Our results indicate

Manuscript received June 8, 2013; revised October 3, 2013; accepted December 20, 2013. The associate editor coordinating the review of this paper and approving it for publication was Z. Wang.

M. Spuhler is with the Department of Information Technology and Electrical Engineering, ETH Zurich, Gloriastrasse 35, 8092 Zurich, Switzerland (e-mail: michaelspuhler@gmail.com).

D. Gustiniano is with the Institute IMDEA Networks, Avda. del Mar Mediterraneo, 22, 28918 Leganes (Madrid), Spain (e-mail: domenico.giustiniano@imdea.org). The work was partially carried out at Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich, Switzerland.

V. Lenders is with armasuisse, Feuerwerkerstrasse 39, 3602 Thun, Switzerland (e-mail: vincent.lenders@armasuisse.ch).

M. Wilhelm and J. B. Schmitt are with the Computer Science Department, TU Kaiserslautern, 67663 Kaiserslautern, Germany (e-mail: wilhelm@cs.uni-kl.de, jschmitt@informatik.uni-kl.de).

Digital Object Identifier 10.1109/TWC.2013.013014.131037

that the chip error rate (CER) based metric outperforms the alternatives across all considered environments and estimates the PDR with a mean absolute error around 5 % for all considered conditions.

- Based upon these insights, we design a jamming detection technique that relies on the CER at the demodulator. Jamming is detected by comparing the estimated PDR of jamming-free preamble symbols with the actual PDR. If the experienced PDR exceeds the one estimated by a certain threshold, a reactive jammer is likely to be active and we thus declare jamming.

We have implemented our detection technique on the USRP1 software-defined radio platform and tested its performance in a controlled lab environment with three nodes: an IEEE 802.15.4 transmitter, an IEEE 802.15.4 receiver with our jamming detector, and the reactive jammer from [6]. Owing to the rich information that can be extracted from preamble chips (up to 256 jamming-free chips per packet), our results show that our detection scheme is able to accurately detect reactive jammers on fading wireless links with a PDR above 0.3. The false positive and negative detection rates for jammers that target all packets remain zero, outperforming models that rely on RSS [1], [7]. In addition, 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 a packet transmission, which prevents approaches in related work to derive accurate PDR estimates. To the best of our knowledge, our detector is the first to detect such sophisticated reactive jamming attacks targeting the physical layer header of packets and does not require any modifications or additional system requirements.

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. Sections III and IV explore the feasibility to model the packet delivery with limited information from chip errors in the preamble and compare it with existing approaches. In Section V, we introduce our jamming detection scheme based on the CER metric. Section VI covers the evaluation of the detection performance. Related work is discussed in Section VII and Section VIII concludes the paper.

## II. 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 used in the evaluation.

### A. Background on IEEE 802.15.4

**Packet transmission.** Our work on jamming detection focuses on direct sequence spread spectrum (DSSS) communication systems, and is practically demonstrated for the 2.4 GHz physical layer (PHY) of the IEEE 802.15.4 standard [8, Section 6.5]. This PHY defines a 16-ary quasi-orthogonal DSSS modulation technique; the modulation spreads a low-rate bit sequence to a higher-rate sequence, consisting 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 $(b_0, b_1, b_2, b_3) \mapsto (c_0, c_1, \ldots, c_{31})$, 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.

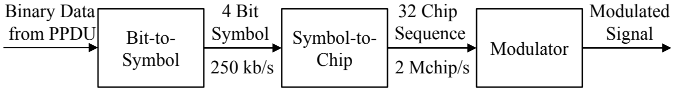


Fig. 1. DSSS modulation in the 2.4 GHz physical layer of IEEE 802.15.4.

Our proposed detection scheme relies on an estimation of the PDR based on the observation of the packet preamble. The preamble in IEEE 802.15.4 is a sequence of eight symbols `0` with the same modulation as the following data bits of the packet. After the preamble follows a start of frame delimiter (SFD; symbols `7` and `10`), a frame length field indicating the duration of the frame, and finally the MAC protocol data unit (MPDU). The MPDU contains a MAC header, data payload, and ends with a frame check sequence (FCS) 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. This implies that reactive jammers can drop packets very efficiently by destroying only one or two symbols in a packet [6].

**Packet reception.** 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, and the carrier and baseband phase offsets. 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 each received block of 32 chips back to symbols. It is compared to the 16 predefined chip sequences $C_i, i = 0, 1, \ldots, 15$. The received chip sequence $R$ may contain errors caused by fading or interference. 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 differing chips) between the two arguments. However, if too many chips are flipped (e.g., when a jammer is active), then the expression $h(R, C_i)$ may be minimal for the wrong chip sequence $C_i$ and the receiver interprets the chip sequence as a wrong symbol.

**Example.** Figure 2 illustrates the synchronization phase of two packets schematically. In the first case, the packet is lost due to bad channel conditions, while in the second case the packet is transmitted successfully despite chip errors. In (a), the sender starts to transmit the preamble sequence, the SFD, and the corresponding length field and MPDU (denoted here as *rest of packet*). During the transmission of the eight preamble symbols of the first packet, $P_{1,2}$, $P_{1,3}$, $P_{1,4}$ are not decoded correctly due to a high number of chip errors. In contrast, $P_{1,7}$ is transmitted successfully because, as shown in (d), only three chips are flipped during the transmission and the maximum error threshold to discriminate between a correct and wrong preamble symbol is not exceeded. Finally, due to a corrupted symbol in $SFD_1$, the synchronization of the first packet fails and the receiver is not able to decode this packet entirely. Specifically, this means that the packet is not counted as a

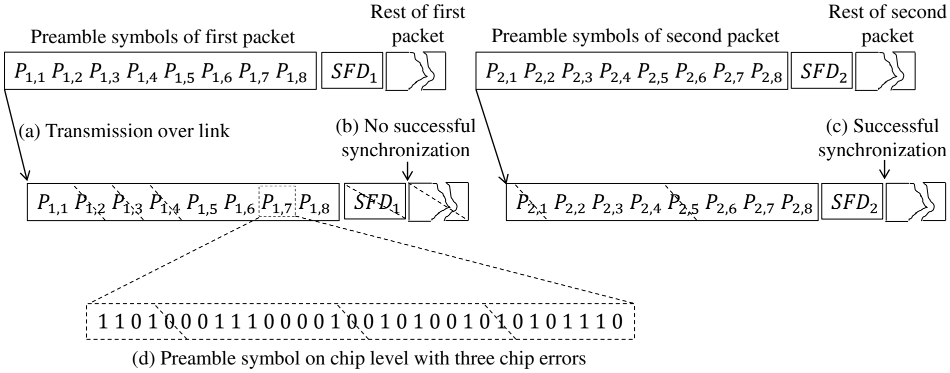

Preamble symbols of first packet
Rest of first packet
Preamble symbols of second packet
Rest of second packet
$P_{1,1}$ $P_{1,2}$ $P_{1,3}$ $P_{1,4}$ $P_{1,5}$ $P_{1,6}$ $P_{1,7}$ $P_{1,8}$
$SFD_1$
$P_{2,1}$ $P_{2,2}$ $P_{2,3}$ $P_{2,4}$ $P_{2,5}$ $P_{2,6}$ $P_{2,7}$ $P_{2,8}$
$SFD_2$
(a) Transmission over link
(b) No successful synchronization
(c) Successful synchronization
1 1 0 1 0 0 0 1 1 1 0 0 0 0 1 0 0 1 0 1 0 0 1 0 1 0 1 0 1 1 1 0
(d) Preamble symbol on chip level with three chip errors

Fig. 2. Examples of how bad channel conditions affect chip errors in the preamble, and its relation with packet losses.

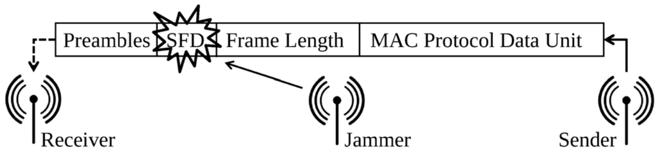


Fig. 3. Reactive jamming: an attacker jams the start-of-frame delimiter (SFD) to disturb the synchronization of the packet at the receiver. Without SFD, the receiver cannot synchronize with the packet and misses it entirely.

packet error because the receiver never enters its reception mode and its FCS is not checked, making it hard to derive statistics for jamming detection when synchronization fails.

Contrary to the first packet, the second packet is transmitted successfully (c) because only the preamble symbols $P_{2,1}$ and $P_{2,5}$ are not correctly decoded, which allows the receiver to synchronize to the packet and decode a valid SFD. Concluding, symbol level analysis can only distinguish between symbols above and below the chip-error threshold. Instead, chip errors provide a richer information about the status of the channel and its expected PDR. In Section III, we will show that the number of chip errors is highly correlated with the probability of successful packet reception, and that the use of the preamble enables us to accurately derive PDR statistics even if a receiver never enters reception mode.

### *B. Attacker Model*

We consider jammers that aim to block the entire communication over a link by emitting interference reactively when they detect packets over the air. The jammers minimize their jamming activity to only a few symbols per packet and use minimal but sufficient power to remain undetected. We assume that the jammer is able to sniff any symbol of the packet over the air in real-time and react with a jamming signal that flips selected symbols at the receiver with high probability. An attacker may therefore pursue different reactive jamming strategies [6]. It may jam *(i)* the MPDU, *(ii)* the frame length field, *(iii)* the SFD, or *(iv)* the preamble of the packet. Figure 3 illustrates jamming strategy *(iii)* that targets the SFD. 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. Such synchronization errors make it hard to devise jamming detectors because often the packet error count is used to distinguish jammed and non-jammed situations [1], [7], which cannot be derived in this situation. The experimental evaluation in Section VI shows that our CER-based approach does not suffer from this restriction, and we are able to detect all four jamming strategies.

We also assume that the attacker cannot destroy all preamble symbols, i.e., at least a few symbols across several packets are available as input to our detector. We denote the time difference between the arrival of the original signal and the jammer signal at the receiver as the jamming reaction time $\tau$. 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., $16\,\mu$s in IEEE 802.15.4). Otherwise it would not be possible to assess the channel state prior to jamming, i.e., not be reactive. In fact, [6] showed that the reaction time of a realistic jamming system 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 (for example, by using simple power detectors with analog parts [9], [10]), reactive jammers of that kind are unable to use the semantics of the signals to perform smart jamming decisions such as jamming selected packets according to specific rules (e.g., matching packet modulation or header properties).

### *C. Experimental Setup*

We rely on measurements to study the performance of packet delivery models and to evaluate the proposed jamming detection technique. 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 [11] that runs on the GNU Radio framework.

We have performed several tests in indoor lab environments, which are referred to as *cable*, *static*, and *mobile*. In the *cable* experiments, sender and receiver are connected by a shielded $60\,\text{cm}$ coaxial cable with a $30\,\text{dB}$ attenuator. This provides very stable link conditions and lets us evaluate the best case performance. In the *static* experiments, a stationary sender and receiver communicate using omni-directional antennas. While providing insights into the detection performance under the fading characteristics of indoor environments, it offers the same stable link conditions as, for example, typical in sensor network installations [12]. 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 $v = 1\,\text{cm/s}$ away from, and back towards, the sender. In this setting the PDR is dynamic, allowing us to evaluate the convergence speed and the stability of our detector.

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. Varying link conditions in the cable and static experiments are obtained by adjusting the transmit power and by changing the nodes' positions. The true PDR at time $t$ is calculated by averaging the number of correctly received packets in a window of $100$ packets centered around $t$. This window size ensures that the true PDR is calculated over a time window smaller than the channel coherence time when moving the receiver at maximum $v = 1\,\text{cm/s}$ and at a frequency of $2.4\,\text{GHz}$.[1] Note that the mobility experiments have a relatively low node speed 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 is 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 Wilhelm et al. [6], which runs on the USRP2 software radio platform from Ettus Research. It can be configured to jam according to strategies *(i)* to *(iv)* introduced in Section II-B. The detection and decision logic are implemented on the FPGA of the USRP2, resulting in a minimal reaction delay of $\tau_{\min} = 19\,\mu\text{s}$.

## III. Experimental Analysis of Preamble-based Performance Metrics

Our jamming detection technique is based on estimating the PDR from the first few preamble symbols. This section provides an experimental study of different performance metrics underlying this estimation. We consider four preamble-based performance metrics to estimate the PDR at the receiver of a link, which are calculated in average:

- number of decoded preamble symbols per successfully delivered packet,
- number of consecutively decoded preamble symbols per transmitted packet,
- number of chip errors per preamble symbol (CER),
- signal-to-noise ratio (SNR) during the preamble.

The key question we strive to answer is how well these metrics are able to predict the actual PDR. An important remark for the computation of the CER is the following. If too many chips are flipped, the expression $h(R, C_i)$ is minimal for the wrong chip sequence $C_i$, such that the receiver interprets the chip sequence as a wrong symbol. The result is that the symbol is discarded and ignored in the computation of the average CER. This means that only a (potentially small) subset of preamble symbols is used in the estimation.

We measure the correlation of these four metrics with the PDR in various settings (cable, static, and mobile) and determine the Pearson correlation coefficient. This coefficient is an indicator of the linear correlation of two variables, where values close to zero indicate a low correlation and absolute values close to one represent a high linear dependence of two variables.

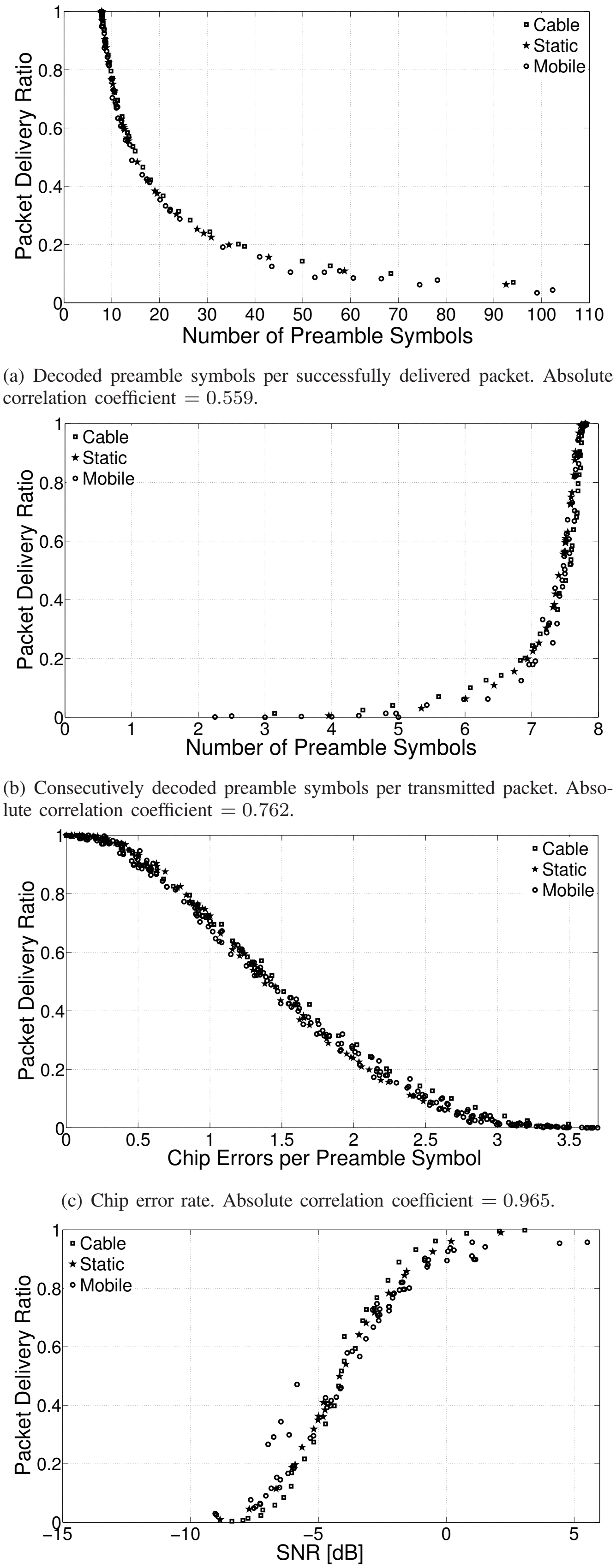
The correlations are plotted individually in Figure 4 for cable, static and mobile experiments. Since the environment has apparently only little impact on the distribution of the metrics, we compute a single correlation coefficient over all three environments for each metric in the further analysis. The best correlation is achieved for the CER metric (Figure 4(c)) with an absolute correlation coefficient of $0.965$, followed by the SNR (Figure 4(d)) with an absolute correlation coefficient of $0.92$. The other two metrics perform significantly worse. The number of decoded preamble symbols per successfully delivery packet (Figure 4(a)) achieves an absolute coefficient of only $0.559$, while the number of consecutively decoded preamble symbols per transmitted packet (Figure 4(b)) exhibits an absolute correlation coefficient of $0.762$.

Given the lower correlation of the two symbol error-based metrics, we do not consider these any further and focus in the following on the most promising two: the CER and SNR based metrics. As a next step, we analyze the correlation coefficient over different time intervals, i.e., when the metrics are averaged over varying window sizes. Small window sizes are considered particularly important when the jamming detection algorithm is expected to perform fast. Figure 5 shows how the absolute value of the correlation coefficient of the CER and SNR-based metrics varies with the number of packets used for computing these metrics. As we can see the correlation is dependent on the window size. However, for any fixed window size, the CER-based metric outperforms the SNR-based one. We therefore conclude that the number of chip errors in the preamble is the best metric among those considered.

## IV. Chip Error Based Model of Packet Delivery

We have seen in the previous section that the CER correlates well with the actual PDR. In this section, we develop an estimator of the PDR based on this metric. To meet the requirements of accuracy and stability, our estimator operates on two time scales. At the preamble level, chip errors of received symbols are first averaged and fitted to a polynomial model to obtain an estimation of the instantaneous PDR. At the packet level, chip error statistics from multiple transmitted packets are filtered according to a weighted moving average function to smooth out short-term fluctuations of the estimation method.

[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.



(a) Decoded preamble symbols per successfully delivered packet. Absolute correlation coefficient $= 0.559$.

(b) Consecutively decoded preamble symbols per transmitted packet. Absolute correlation coefficient $= 0.762$.

(c) Chip error rate. Absolute correlation coefficient $= 0.965$.

(d) Signal-to-noise ratio. Absolute correlation coefficient $= 0.92$.

Fig. 4. Correlation between different preamble-based performance metrics and PDR (average numbers).

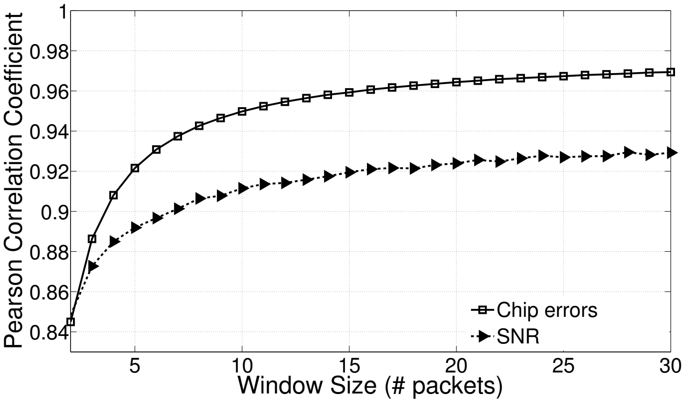


Fig. 5. Comparison of the Pearson correlation coefficient (absolute value) of the packet delivery ratio versus the chip error and SNR distributions for different window sizes.

### *A. Instantaneous PDR*

In a first step, we estimate the instantaneous (per-packet) PDR after the reception of the preamble of packet $k$ as

$$PDR_{\text{inst}}(k) = g\left(\frac{\sum_{i=1}^{32}\sum_{j=1}^{|\mathcal{S}_k|}\left(P_{k,j}[i] \oplus P[i]\right)}{|\mathcal{S}_k|}\right),$$

where $P_{k,j}[i]$ is a vector containing the 32 chips of the $j$-th received preamble symbol of packet $k$ for $i = 1, 2, \ldots, 32$, $P[i]$ denotes a vector with the expected chips of the known preamble symbol, $\oplus$ is the exclusive OR operator, and $|\mathcal{S}_k|$ is the number of received preamble symbols for packet $k$. The function $g(\cdot)$ models the empirical distribution of the PDR versus CER as shown in Figure 4(c). For best results, we use a polynomial regression function. We have experimented with polynomials of different degrees. The root mean square error of the fit could significantly be decreased up to a fifth degree polynomial. Higher degrees only resulted in minimal improvements. The fifth degree polynomial we used in this paper is of the form

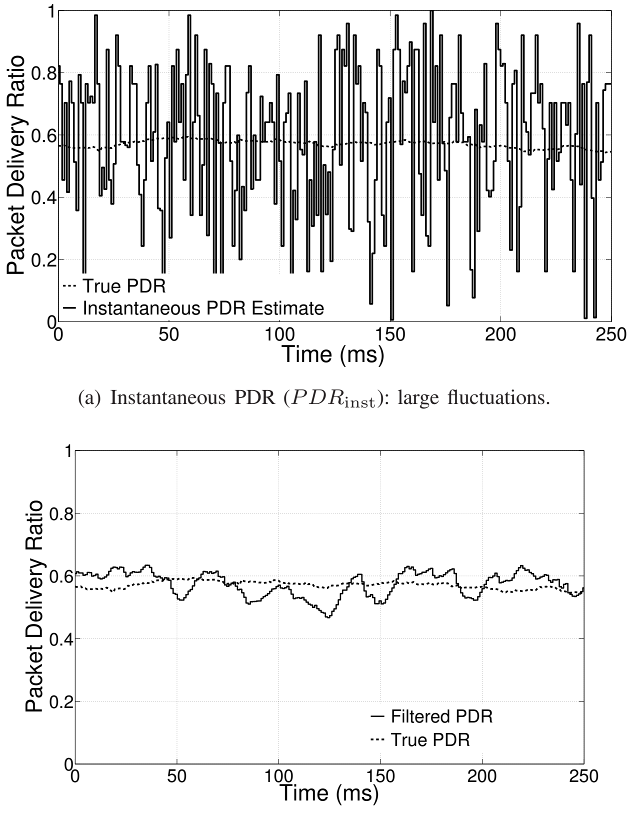
$$g(p) = \mathrm{a}_5\ p^5 + \mathrm{a}_4\ p^4 + \mathrm{a}_3\ p^3 + \mathrm{a}_2\ p^2 + \mathrm{a}_1\ p + \mathrm{a}_0$$

with the parameters of the fit being $\mathrm{a}_5 = 0.016$, $\mathrm{a}_4 = -0.33$, $\mathrm{a}_3 = 2.41$, $\mathrm{a}_2 = -7.26$, $\mathrm{a}_1 = 8.83$, $\mathrm{a}_0 = -3.24$. The root mean square error for this polynomial regression function is below $3\,\%$ across the entire range.

While $PDR_{\text{inst}}(k)$ provides a very fast estimate of the link quality, it is subject to large fluctuations as shown in Figure 6(a). The figure compares the fluctuation of the instantaneous PDR on a static link to the true PDR defined as the ratio of correctly received packets to the total number of sent packets for a fixed time window of 100 packets (see Section II-C). To provide a more stable link quality metric, we need to further average and filter consecutive instantaneous PDR estimates as described next.

### *B. Averaged and Filtered PDR*

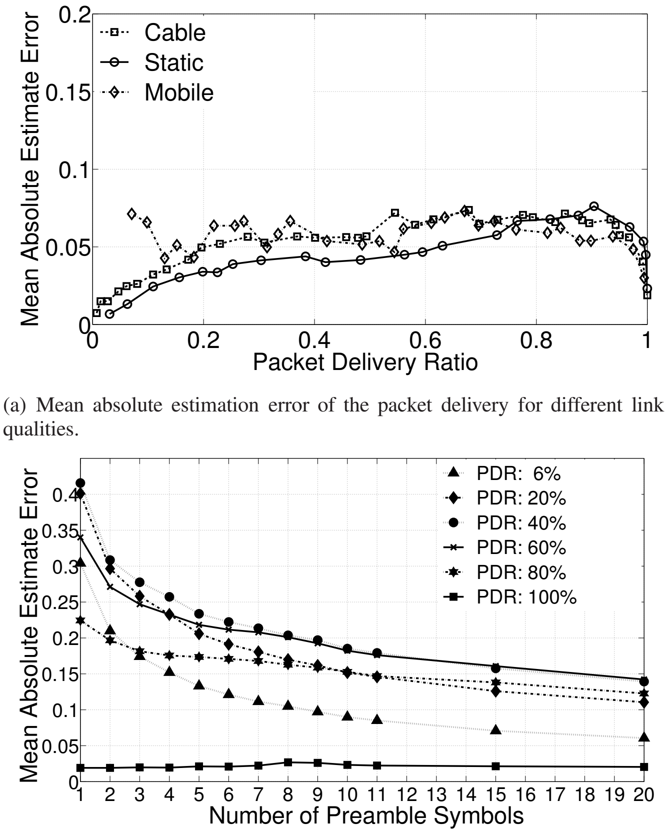
A classical approach to increase the stability of an estimator is to weight sequential estimates in form of a weighted moving average. For example, Woo et al. [13] use this technique to increase the stability of estimators using packet count



(a) Instantaneous PDR ($PDR_{\text{inst}}$): large fluctuations.

(b) Filtered PDR ($PDR_{wa}$): small fluctuations.

Fig. 6. Fluctuation of the instantaneous PDR estimation, $\mathcal{PDR}_{\text{inst}}$, and filtered PDR estimation, $PDR_{wa}$, on a wireless static link with a true PDR of approximately 55 %.



(a) Mean absolute estimation error of the packet delivery for different link qualities.

(b) Mean absolute estimation error of the packet delivery for various PDRs and different numbers of preamble symbols in the model.

Fig. 7. Estimation error of chip error-based model of PDR.

statistics. We apply a similar approach to smooth out consecutive samples of the instantaneous PDR: we perform a low-pass filtering of the weighted average in a window of $w$ consecutive samples. Suppose $\mathcal{PDR}_{\text{inst}}(k)$ is the set of the past $\ell+1$ samples of the instantaneous PDR at position $k$. Let further $PDR_{\text{inst}}(k-\ell), PDR_{\text{inst}}(k-\ell+1), \ldots, PDR_{\text{inst}}(k-1), PDR_{\text{inst}}(k) \in \mathcal{PDR}_{\text{inst}}(k)$ be the past $\ell+1$ samples. Then, the weighted average $\text{wa}(k)$ over these recent $\ell+1$ samples at the position $k$ is calculated as

$$\text{wa}(k) = \sum_{m=0}^{\ell} \beta_m PDR_{\text{inst}}(k-m),$$

with the weighting factors $\beta_m$ such that $\sum_{m=0}^{\ell} \beta_m = 1$. Using this weighted average $\text{wa}(k)$, we compute the output of the low pass filter $f_{\text{wa}}(k)$ (filtered weighted average) as

$$f_{\text{wa}}(k) = \alpha\ f_{\text{wa}}(k-1) + (1-\alpha)\left(\frac{1}{\text{wa}(k)} - 1\right),$$

where the parameter $\alpha \in [0,1]$ controls the smoothness. For example, a small factor $\alpha$ gives more importance to the recent link behavior. Finally the $k$-th estimate is obtained as

$$PDR_{wa}(k) = \frac{1}{1 + f_{\text{wa}}(k)}.$$

The benefits of the averaging and filtering are illustrated in Figure 6 (b), showing the resulting estimation error after filtering and weighting the samples. With the parameters $\ell$, $\beta_m$, and $\alpha$, the estimation window can be changed to tune the reactivity of the estimator. In this work, we set $\ell = 6$, $\beta_0 = 0.3, \beta_1 = 0.2, \beta_{2,\ldots,6} = 0.1$, and $\alpha = 0.9$, as we observed that this provides a good configuration in our experiments.

### *C. Performance Evaluation of the Jamming-free Packet Delivery Estimator*

**Detection accuracy.** We evaluate the performance of our estimator for different link qualities and symbol estimation windows. The results are presented in Figure 7. In Figure 7(a), we see the mean absolute estimation error versus the PDR for a fixed estimation window size of ten packets. The mean absolute error remains around a remarkable 5 % on average across all link qualities and environments (cable, static, and mobile) compared to typical values of 3–60 % for existing link quality estimators [14], although these exploit the entire packet to estimate the PDR.

**Detection speed.** As we cannot control the reaction time $\tau$ of the adversary and we are not aware of the reactive jamming strategy employed, it is in addition crucial that the proposed model of packet delivery manages to estimate with as few preamble symbols per packet as possible. Figure 7(b) evaluates the mean absolute estimate error of the PDR versus a varying number of preamble symbols used in the estimation for the static environment. Preamble symbols can be accumulated over multiple packet 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 decreases with the number of preamble symbols, providing a useful estimator even for a model that needs to cope with only a few symbols.

## V. CER-BASED JAMMING DETECTION

In this section, we describe our jamming detection scheme that relies on the packet delivery model introduced in the previous section. The basic idea is that the receiver computes two metrics based on the incoming traffic, the observed and an estimated PDR.

**Observed PDR.** The observed packet delivery ratio $PDR_o(t)$ at time $t$ is calculated by counting the ratio of correctly received packets over the total number of transmitted packets in a sliding observation window:

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

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 within the preamble since the reactive jammer is not capable to jam all the preamble symbols. Therefore, 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. Note that when facing an extremely fast reactive jammer, i.e., one that jams close to the sender on any power elevation over the channel without attempting to decode the preamble signals, our method might still detect `0` symbols in the payload of packets. We do not attempt to discriminate those symbols from the preamble symbols as they are still useful to estimate the PDR. In this case, the attacker would be forced to fully destroy a packet to erase all `0` symbols to mitigate our jamming detection mechanism, which greatly sacrifices the energy and stealth benefits of reactive jamming.

The observed $PDR_o$ should be calculated over a time window shorter than the channel coherence time, but sufficiently long to capture enough packets to derive a statistically relevant average. We have experimented with different values in the cable, static and mobile environments. A window size of around 100 data packets has proved to be a good choice across all environments, while not being highly sensitive to variations of this parameter. Hence, in this paper, we use a fixed window size of $W = 100$ ms, corresponding to roughly 100 data packets at the actual transmission rate of the sender.

**Estimated PDR.** The second metric is an estimated PDR based on the CER metric. As shown in Figure 8, the IEEE 802.15.4 receiver demodulates an incoming signal and attempts to map each 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 CER, the receiver averages the Hamming distances of multiple preamble symbols. We stress 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 larger 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 in a preamble. However, we do not require to detect any other field of the packet like the SFD or FCS, enabling our approach to detect a broader range of jamming attacks.

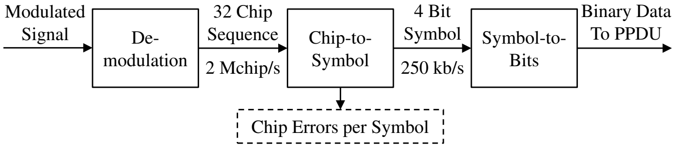


Fig. 8. Chip errors in the preamble symbols are determined during the chip-to-symbol mapping of the receiver.

After receiving enough `0` symbols, the estimated PDR is calculated as

$$PDR_e = PDR_{wa},$$

using the weighted average PDR defined in Section IV-C.

**Jamming detection.** We define a heuristic hypothesis test based on the relative difference $\Delta$ between the estimated and observed PDR

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

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

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

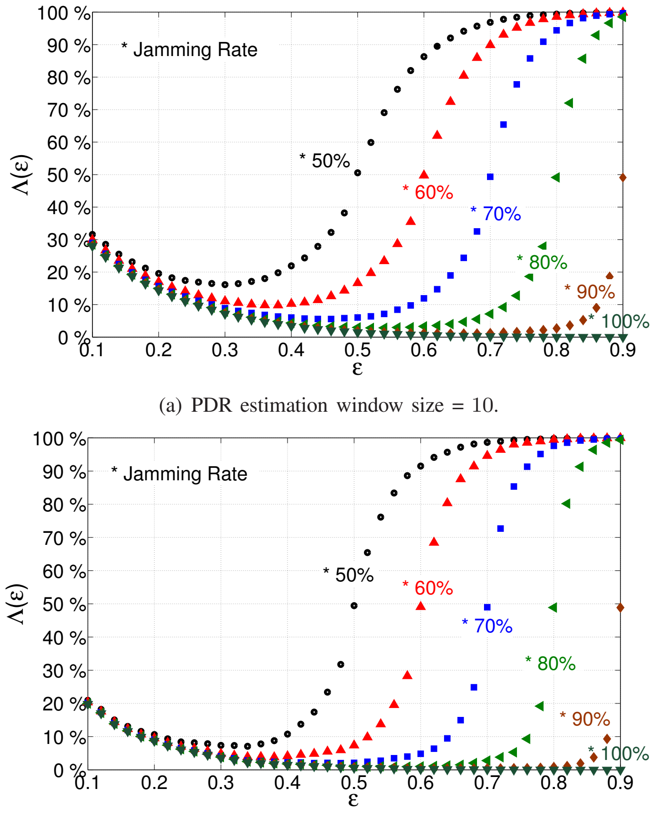
Then the test is as follows:

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

where $\epsilon$ represents a tolerance level that directly affects the false positive and false negative detection rates. Let $\Lambda(\epsilon)$ be the sum of the false positive and false negative detection rates for a given PDR:

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

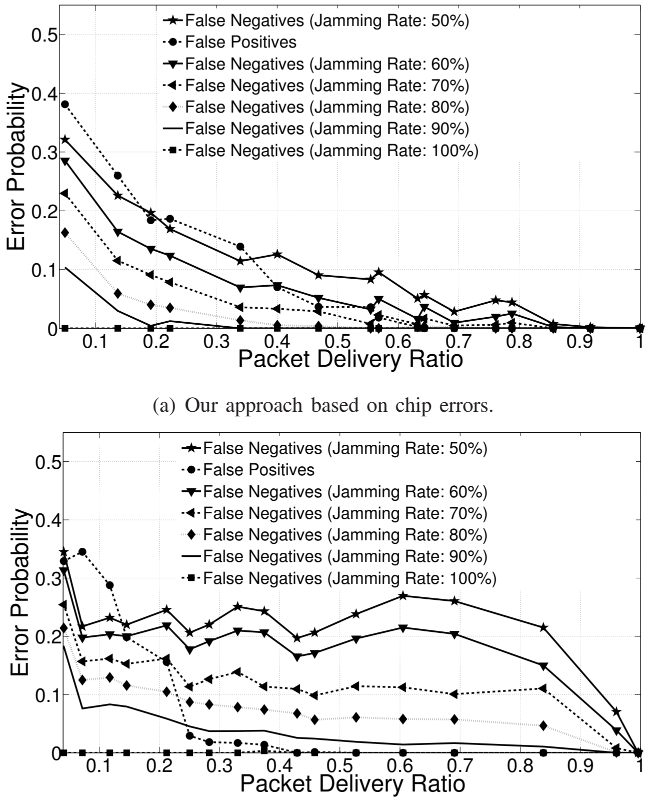
For small tolerance level values $\epsilon$, the jamming detection is more sensitive at the price of a higher false negative rate $P(H_1 \mid \text{jammer off})$. For higher values of $\epsilon$, the false negative rate may be reduced, but, in turn, at the price of a higher false positive rate $P(H_0 \mid \text{jammer on})$. We evaluate the impact of $\epsilon$ on the jamming detection performance in more detail in the next section.



(a) PDR estimation window size = 10.

(b) PDR estimation window size = 50.

Fig. 9. Impact of $\epsilon$ on jamming detection performance.



(a) Our approach based on chip errors.

(b) Alternative approach based on SNR.

Fig. 10. Comparison of jamming detection error performance versus true PDR (PDR estimation window size of 10 packets).

## VI. Evaluation

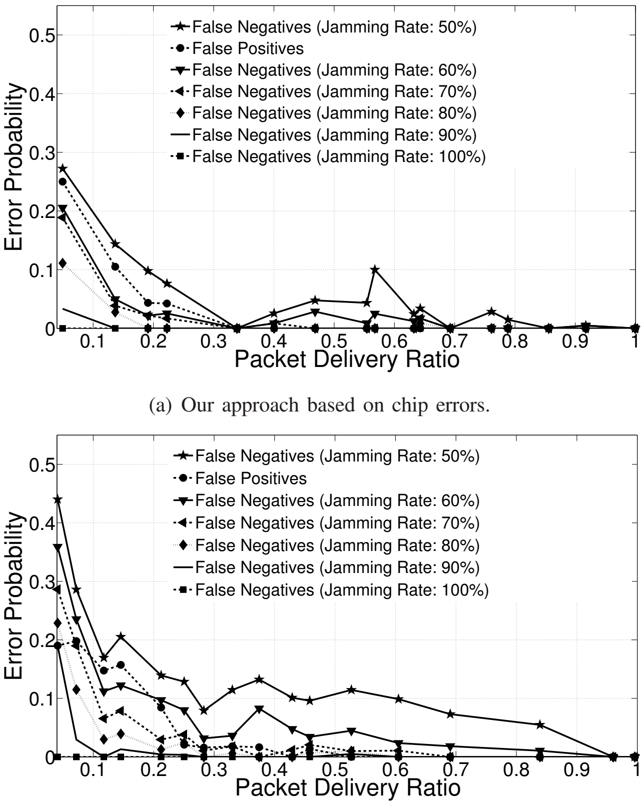
Our evaluation focuses on quantifying the detection performance in terms of false positives and false negatives under realistic wireless fading channel conditions. For this purpose, we benchmark our detection algorithm on software-defined radios with real traffic over the air. The jammer in our experiments reacts and hits the SFD of any transmitted packet. This jamming strategy is of particular interest because packet synchronization fails and thus existing detection mechanisms are unable to detect this type of reactive jammers.

### A. Impact of Tolerance Level $\epsilon$

We first consider the impact of the tolerance level $\epsilon$ of the hypothesis test on the detection performance. Figure 9 highlights the sum of the false positives and negatives $\Lambda(\epsilon)$ for varying values of $\epsilon$ and different jamming rates. Here, we define the jamming rate as the percentage of packets that the reactive jammer hits with interference and causes a packet loss. A jamming rate below $100\,\%$ may for example occur when the jammer fails to detect some packets because it experiences a momentary deep fade during a transmitted signal or when the jamming signal at the receiver is too weak to reliably destroy all packets. Figure 9(a) and Figure 9(b) are obtained with a PDR estimation window of 10 and 50 packets, respectively. After only 10 packets, the hypothesis test is capable of determining the correct hypothesis with high probability, with $\Lambda(\epsilon)$ below $10\,\%$ given that the jamming rate is above $60\,\%$. We see that the optimum $\epsilon$, defined as the level at which $\Lambda(\epsilon)$ is minimized, is dependent on the jamming rate but fairly independent of the estimation window size. We also note that the minimum of $\Lambda(\epsilon)$ for a given jamming rate is reduced for larger window sizes. The robustness of the optimum $\epsilon$ when changing to other $\epsilon$ values increases as jamming rates become higher and is very high for jamming rates close to $100\,\%$. For a window size of 50, choosing a value of $\epsilon$ around $0.3$, the jamming detector achieves false detection rates $\Lambda(\epsilon)$ below $8\,\%$, and regardless of the jamming rates employed. Therefore, jamming detection is not sensitive to the jamming rate when the parameter $\epsilon$ is selected according to this range. This result is remarkable because *the receiver is able to detect the jammer even if it is unaware of both the reactive jamming strategy and the jamming rate*.

### B. Jamming Detection Performance

Finally, we evaluate the false positive rate for an optimal detector that tunes the tolerance level $\epsilon$ to minimize the sum of errors. We compare our results with an SNR-based model; we include this model here because it is the second best metric in our evaluation in Section III, and previous works [1], [7] have proposed to use the SNR to detect reactive jamming. For a fair comparison, we also apply a fifth degree polynomial regression fit to the empirical data in Figure 4(d). We note that we are evaluating the SNR-based detection under best conditions when the (reactive jamming) attacker does not add power to the symbols in the preamble, as required by the receiver to perform jamming-free measurements of the SNR.



(a) Our approach based on chip errors.

(b) Alternative approach based on SNR.

Fig. 11. Comparison of jamming detection error performance versus true PDR (PDR estimation window size of 50 packets).

Figures 10 and 11 show the results of these experiments for packet delivery estimation window sizes of 10 and 50 packets, respectively. Figures 10(a) and 11(a) display the error rate for our approach using the chip error rate to estimate the PDR while Fig. 10(b) and 11(b) show the performance observed when relying on the SNR-based model. The horizontal axis represents the true PDR. Our approach significantly outperforms the SNR-based model in terms of detection accuracy for all the range of PDR.

In addition, we observe two major performance trends. First, while links with a high PDR tend to provide good detection performance, low quality links exhibit higher detection errors. The reason is the following: the mean absolute estimation error (of around $5\%$, see Figure 7(a)) across all PDRs has a relatively higher impact on links with a lower PDR. Therefore, the relative difference $\Delta$ is reduced and estimation errors in the PDR for low quality links are more likely to be mis-interpreted as jamming. Furthermore, it is inherently more challenging to differentiate between losses caused by bad channel conditions and those by jamming since the gap between the true PDR and the estimated PDR under jamming is greatly reduced.

The second trend relates to the jamming rate. While reactive jammers that manage to hit all packets are detected with low or even no error at all, jammers with a jamming rate below $100\%$ are harder to distinguish. Again, this is due to the problem of distinguishing jammed packets from those lost because of bad channel conditions. This is why the false negatives may increase up to $5$–$10\,\%$ depending on the link channel conditions. We also note that the false positive rate is identical for all jamming rates as it represents the errors when no jammer is present.

We believe that these two trends are not problematic issues in real-world applications. Low quality links tend not to be used by higher-layer application protocols because they provide poor system performance. Therefore, detecting jamming on these links is generally not required. Furthermore, attackers that miss to react to a high percentage of the data traffic have limited negative impact on the communication. For example, if an attacker jams only $50\,\%$ of the packets, it means that the remaining $50\,\%$ are still delivered successfully. With a retransmission mechanism in place, a jamming rate of merely $50\,\%$ is thus not sufficient to effectively block the communication over a link.

## VII. Related Work

To the best of our knowledge, this work is the first to provide a jamming detection scheme that can cope with sophisticated reactive jamming attacks targeting packet synchronization. Strasser et al. [7] 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. Therefore it is not able to detect SFD jamming attacks because decoded MPDU symbols are unavailable at the receiver due to the lack of synchronization. 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. Since our approach is not relying on error positions in a packet, it does not suffer from these restrictions.

Xu et al. [1] 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. The reason is that the techniques require the measurement of the RSS *during* a packet, which in turn requires to detect a packet first using the SFD. If an attacker destroys the PHY header, the RSS value cannot be measured reliably.

Xuan et al. [15] 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 [16] 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.

There are also approaches orthogonal to our work that mitigate jamming attacks on higher layers. For example, Richa et al. [17] devise and simulate a MAC protocol, ANTIJAM, to resolve unintentional and malicious interference originated by an adaptive jammer that can determine whether the channel is currently idle or used. Our work differs from this approach since it operates at the physical layer rather than the MAC. However, works as [17] can complement our contribution, since they would benefit from the simplicity and robustness of our approach that can detect with high probability reactive jammers that target the preamble symbols.

Finally, Qin et al. [18] suggest that the CER 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 our work. In our previous work [14], [19], we have shown how chip error based models of packet delivery compare to traditional models of packet delivery. This manuscript complements our previous works by showing how these novel estimators can be used to detect reactive jamming considering only limited information from the preamble. Preliminary results of this work have appeared in a short paper [20]. This article is an extension of our previous work providing more key insights on the performance of different metrics that rely on the information from the preamble to estimate the PDR as well as a much more thorough analysis with additional experiments to understand the detection performance under various parameter configurations and environmental conditions. Additionally, we compare the performance of our proposed detection scheme with a scheme that relies on the RSS.

## VIII. Conclusions

We have proposed a novel approach to detect sophisticated reactive jamming attacks that target any part of a packet transmission. Our approach is based on an estimation of the packet delivery probability during the signal synchronization phase of a packet transmission, which makes it suitable to detect even jammers that target the physical layer header of packets. We have analyzed the accuracy of different preamble-based metrics to predict the packet delivery probability and have shown that the chip error rate (CER) in the received preamble symbols is the most accurate estimator among the ones considered. Our experiments under real-world channel conditions have shown that it is possible to predict the PDR using the CER derived from just a few symbols in the preamble with a mean absolute estimation error of approximately 5 % across all channel conditions.

Based on this, we have developed a jamming detection algorithm that compares the estimated delivery probability with the observed delivery ratio to distinguish between packet losses caused by jamming and losses due to bad channel conditions. Our technique is able to detect reactive jammers that jam all packets on links with a PDR above 0.3 without any false positive or negative detection errors.

## References

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

[2] M. Çakiroglu and A. T. Özcerit, “Jamming detection mechanisms for wireless sensor networks,” in *Proc. 2008 ICST InfoScale*, pp. 1–8.

[3] 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.*, vol. 5, no. 1, pp. 6:1–6:38, Feb. 2009.

[4] A. Proaño and L. Lazos, “Packet-hiding methods for preventing selective jamming attacks,” *IEEE Trans. Dependable Secure Comput.*, vol. 9, no. 1, pp. 101–114, Jan. 2012.

[5] A. D. Wood, J. A. Stankovic, and G. Zhou, “DEEJAM: defeating energy-efficient jamming in IEEE 802.15.4-based wireless networks,” in *Proc. 2007 IEEE SECON*, pp. 60–69.

[6] M. Wilhelm, I. Martinovic, J. B. Schmitt, and V. Lenders, “Reactive jamming in wireless networks: how realistic is the threat?” in *Proc. 2011 ACM WiSec*, pp. 47–52.

[7] M. Strasser, B. Danev, and S. Čapkun, “Detection of reactive jamming in sensor networks,” *ACM Trans. Sensor Netw.*, vol. 7, no. 2, pp. 16:1–16:29, Aug. 2010.

[8] IEEE Standard 802 Part 15.4: Wireless Medium Access Control and Physical Layer Specifications for Low-Rate WPANs, IEEE Computer Society Std., Sept. 2006.

[9] K. B. Rasmussen and S. Čapkun, “Realization of RF distance bounding,” in *Proc. 2010 USENIX Security*, pp. 389–402.

[10] M. Kuhn, H. Luecken, and N. O. Tippenhauer, “UWB impulse radio based distance bounding,” in *Proc. 2010 WPNC*, pp. 28–37.

[11] T. Schmid, “GNU Radio 802.15.4 en- and decoding,” UCLA NESL, Tech. Rep. TR-UCLA-NESL-200609-06, Sept. 2006.

[12] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, and M. Alves, “Radio link quality estimation in wireless sensor networks: a survey,” *ACM Trans. Sensor Netw.*, vol. 8, no. 4, pp. 34:1–34:33, Sept. 2012.

[13] A. Woo and D. Culler, “Evaluation of efficient link reliability estimators for low-power wireless networks,” University of California, Berkeley, Tech. Rep. CSD-03-1270, Mar. 2003. Available: http://techreports.lib.berkeley.edu/accessPages/CSD-03-1270.html

[14] M. Spuhler, V. Lenders, and D. Giustiniano, “BLITZ: wireless link quality estimation in the dark,” in *Proc. 2013 EWSN*, pp. 99–114.

[15] Y. Xuan, Y. Shen, N. P. Nguyen, and M. T. Thai, “A trigger identification service for defending reactive jammers in WSN,” *IEEE Trans. Mobile Comput.*, vol. 11, no. 5, pp. 793–806, May 2012.

[16] J. T. Chiang and Y.-C. Hu, “Cross-layer jamming detection and mitigation in wireless broadcast networks,” *IEEE/ACM Trans. Netw.*, vol. 19, no. 1, pp. 286–298, Jan. 2011.

[17] A. Richa, C. Scheideler, S. Schmid, and J. Zhang, “An efficient and fair MAC protocol robust to reactive interference,” *IEEE/ACM Trans. Netw.*, vol. 21, no. 3, pp. 760–771, June 2013.

[18] Y. Qin, Z. He, and T. Voigt, “Towards accurate and agile link quality estimation in wireless sensor networks,” in *Proc. 2011 IFIP Med-Hoc-Net*, pp. 179–185.

[19] P. Heinzer, V. Lenders, and F. Legendre, “Fast and accurate packet delivery estimation based on DSSS chip errors,” in *Proc. 2012 IEEE INFOCOM*, pp. 2916–2920.

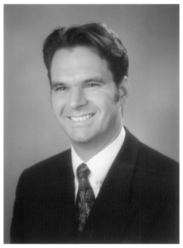
[20] D. Giustiniano, V. Lenders, J. B. Schmitt, M. Spuhler, and M. Wilhelm, “Detection of reactive jamming in DSSS-based wireless networks,” in *Proc. 2013 ACM WiSec*, pp. 43–48.

**Michael Spuhler** received his M.Sc. in electrical engineering and information technology in 2012 from ETH Zurich, Switzerland. In his master thesis, his research interests included radio link quality estimation and jamming in wireless networks. After receiving his degree, he joined the Swiss bank UBS.

**Domenico Giustiniano** is Research Assistant Professor at IMDEA Networks. Before joining IMDEA, he was a Senior Researcher and Lecturer in the Communication System Group (CSG) of ETH Zurich, and Post-Doctoral Researcher at Disney Research Zurich (2010-2012) and at Telefonica Research Barcelona (2008-2010). He holds a PhD in Telecommunication Engineering from the University of Rome Tor Vergata. Dr. Giustiniano devotes most of his current research to emerging areas in the field of wireless networking and localization systems. The original contributions he has made to his field of research are exemplified by first-author publications in top dissemination venues such as ACM MobiCom, ACM CoNEXT and IEEE INFOCOM, and in journals such as IEEE/ACM TRANSACTIONS ON NETWORKING, and best paper award at IFIP Wireless Days '12 for his contribution on visible light networking with LED-to-LED communication. His approach to scientific work oriented to devise solutions to real-world problems based on real-world assumptions is further proved by four patents.

**Vincent Lenders** is a research program manager at armasuisse. Since 2012, he is responsible for the Cyberspace and Information research program. Besides research, he has developed the IT security and privacy concepts of the operational C4 and ISTAR systems at the Swiss Air Force for which he co-received the Swiss department of defense's Security Award in 2011. Before joining armasuisse, Vincent Lenders was a postdoctoral researcher at Princeton University, New Jersey (2007). He earned his Ph.D. degree (2006) and M.sc (2001) in electrical engineering both at ETH Zurich, Switzerland. Since 2012, Vincent Lenders serves as the industrial director of the Zurich Information Security and Privacy Center (ZISC) at ETH Zurich, Switzerland. He is the recipient of the best paper award at the IEEE International Conference on Wireless On-demand Network Systems and Services (WONS) 2012 and a member of the IEEE and the ACM.

**Matthias Wilhelm** received the M.Sc. in computer science and electrical engineering from the University of Kaiserslautern, Germany, in 2009. He is currently pursuing the Ph.D. degree in computer science at the same institution. His research interests include securing wireless networks using physical layer techniques, RF jamming and interference, and the applications of software-defined radio technology. Mr. Wilhelm was a recipient of the EWSN/CONET M.Sc. Academic Award in 2010 and a winner in the ACM Student Research Competition Grand Finals in 2012.



**Jens B. Schmitt** is a professor for Computer Science at the University of Kaiserslautern, Germany. Since 2003 he has been heading the Distributed Computer Systems Lab (disco). His research interests are broadly in performance and security aspects of networked and distributed systems. He received his Ph.D. in 2000 from TU Darmstadt, Germany.