

# BLITZ: Wireless Link Quality Estimation in the Dark

Michael Spuhler<sup>1</sup>, Vincent Lenders<sup>2</sup>, and Domenico Giustiniano<sup>1</sup>

<sup>1</sup> ETH Zürich, Switzerland

<sup>2</sup> armasuisse, Switzerland

**Abstract.** We present BLITZ, a novel link quality estimator that relies on physical-layer synchronization errors to estimate the expected packet delivery ratio of wireless links. In contrast to all existing link quality estimators which estimate the packet delivery based on statistics from packets that are *successfully* decoded, our technique works even when packets at the receiver are *not correctly received*, i.e., when the synchronization fails. The core idea of BLITZ is to exploit information from chip errors in the received preamble of any transmitted direct sequence spread spectrum signals such as IEEE 802.15.4. Using extensive measurements over cable, wireless static and wireless mobile scenarios, we show that our proposed estimator outperforms existing estimators in terms of both accuracy and speed. Across diverse environmental conditions and the full range of possible link qualities, BLITZ provides packet delivery ratio estimates with an absolute error below six percent within just a few milliseconds.

## 1 Introduction

A fundamental problem of wireless network protocols consists of link quality estimation. Link quality estimation is a crucial building block for higher layer protocols in wireless networks. The performance of routing, rate selection, handover, jamming detection, or network coding heavily depends on accurate and fast wireless link quality estimator. For example, long prediction time for rate selection would rapidly deteriorate the link quality before action has been taken. Despite these needs, the task of estimating the effective link quality in real-life wireless networks remains a challenge. Particularly in dynamic link environments, where moving nodes foster the unpredictable and location-sensitive nature of wireless channels, estimating the packet delivery probability *efficiently*, *quickly* and *accurately* proves to be a difficult task.

Current wireless link quality estimators generally fail to provide accuracy, reactivity and stability at the same time across diverse environments and conditions. For example, existing SNR [1] and chip errors [2] based estimators tend to be fast but empirical studies have shown that they often fail to accurately predict the packet delivery ratio (PDR) [2–5]. Estimators that rely on historical packet count statistics [6–8] provide a more accurate representation of the channel conditions. However, these models are relatively slow as they require a few packets for building up meaningful packet statistics (typically more than ten packets). Hybrid models combining the above approaches [9–11] have shown to further improve the accuracy. Yet, the estimation time remains high as a few packets are still required to build up a reliable link estimation.

A common limitation of all current link quality estimators is that they update statistics exclusively when packets are successfully transmitted or received. We argue in this

paper that this approach has two fundamental drawbacks. First, in existing SNR and chip errors-based estimators, the measured metrics are biased towards an estimation of good channel conditions since only successful packets are considered. Second, and most important, in all current link quality estimators, low quality links that exhibit a high number of packet losses require considerable amounts of time to be sampled. For example, a link with a low packet delivery ratio of 20% requires statistically the transmission of 5 packets to obtain one successful packet for actual estimation.

This paper proposes BLITZ, a novel estimator of the packet delivery ratio for point-to-point communication that analyzes errors in *frame synchronization*. In contrast to all existing estimators, BLITZ does not require any successful packet transmissions to estimate the link quality as it is able to analyze the synchronization of packets even when packets are lost<sup>1</sup>. BLITZ operates on direct sequence spread spectrum (DSSS) systems in which each information symbol is mapped to multiple chips. The key idea is to analyze chip error patterns in preamble symbols during the synchronization phase. Given this mapping of one symbol to many chips, BLITZ relies on a rich source of information to estimate the channel state over a short duration. For example in IEEE 802.15.4, a popular sensor network standard that relies on DSSS, the preamble consists of 8 symbols of 32 chips each, resulting in 256 chips per preamble that can be used to estimate the channel state in 128  $\mu$ s.

We have implemented BLITZ on the USRP software-defined radio platform and tested its performance under idealized and real-world wireless channel conditions. In particular, we have compared its performance to existing estimators that rely on packet statistics, SNR, and chip errors in the payload. Our results show that BLITZ is superior to all estimators in terms of both accuracy and reactivity. In particular, we show that

- BLITZ has the best performance over all wireless scenarios with an average PDR estimation error of 5.7 %,
- because of its fast estimation, BLITZ is particularly suitable in mobile settings where it outperforms on average all other estimators by 10 % to 20 % with regard to the absolute PDR error,
- because BLITZ does not rely on successful transmissions to estimate the link quality, the PDR is estimated accurately within just a few milliseconds, and regardless of the (good or bad) link quality.

To the best of our knowledge, this work is the first to show that errors in the synchronization phase of wireless frame transmissions can be used to accurately model the packet delivery over wireless links.

The rest of this paper is structured as follows. In the next section, we present relevant background information on the IEEE 802.15.4 standard which is the base for our analysis. Section 3 analyzes the characteristics of wireless synchronization errors in different environmental conditions. Section 4 presents BLITZ, our novel link quality estimator. The performance of BLITZ is compared to other estimators in Section 5. Section 7 discusses related work and Section 8 finally concludes the paper.

---

<sup>1</sup> we refer to this capability as link quality estimation in the **dark** because traditional estimators do not see packets that are lost due to synchronization failures.

## 2 Background on IEEE 802.15.4

Our work on link quality estimation focuses on direct sequence spread spectrum (DSSS) communication systems and in particular on the IEEE 802.15.4 standard [12]. This section gives a brief overview of DSSS and how it is employed in IEEE 802.15.4.

### 2.1 Modulation and Decoding

The IEEE 802.15.4 standard defines a 16-ary quasi-orthogonal modulation technique based on DSSS. This modulation spreads a low rate sequence of bits to a higher rate sequence of so called chips. The binary source data is divided into groups of 4 bits (referred to as symbols) and mapped to a nearly 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.

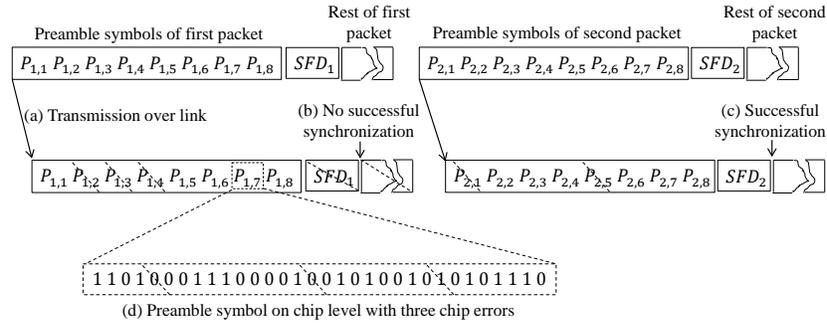
A frame in IEEE 802.15.4 consists of a preamble sequence (eight 0 symbols), a start of frame delimiter (symbol 7 followed by symbol 10), a frame length field and a MAC protocol data unit (MPDU). The MAC protocol data unit ends with a frame check sequence (FCS) which is used to detect errors in the MAC payload.

At the receiver side, the signal is decoded using a correlator to map the received 32-chip sequences back to symbols. When the receiver detects a signal on the channel, it synchronizes on the frame by checking the incoming chip sequences and comparing them to the expected sequences of the preamble and start of frame delimiter. The received chips may contain errors caused by fading or interference. The received chip sequence  $R$  is interpreted correctly as  $C$ , when the hamming distance is smaller than a threshold:  $h(R, C) < \text{threshold}$ , where  $h(\cdot, \cdot)$  is the hamming distance (number of positions containing different chips) between the two arguments. For example  $h((01101), (01110)) = 2$  because the two vectors vary in the last two bits which are flipped. After successful synchronization, the following chip sequences are decoded according to the best match, i.e., the received chip sequence  $R$  is compared to the 16 predefined chip sequences  $C_i, i = 1, 2, \dots, 16$ . The receiver chooses the best match, i.e. the  $C_i$ , such that  $h(R, C_i)$  is minimized. If too many chips are flipped, the expression  $h(R, C_i)$  may be minimized for the wrong chip sequence  $C_i$ , such that the receiver interprets the received chip sequence as a wrong symbol.

### 2.2 Synchronization and Packet Losses

The preamble sequence and the start of frame delimiter are the basis of the synchronization process between a transmitter and a receiver in IEEE 802.15.4. Figure 1 illustrates schematically the synchronization phase of two packets, where the first packet is lost due to a synchronization error and the second packet is transmitted successfully. In (a), the sender starts to transmit the preamble sequence, the SFD (start of frame delimiter) and the corresponding part of the packet (named here as the rest of packet). During the transmission of the eight preamble symbols of the first packet,  $P_{1,2}, P_{1,3}, P_{1,4}$  could not be decoded correctly due to high number of chip errors. E.g.  $P_{1,7}$  was transmitted successfully because as shown in (d) only three chips were flipped during the transmission, and the maximum error threshold to discriminate between a correct or wrong preamble symbol is not exceeded. Due to a corrupted symbol in the  $SFD_1$  the synchronization of

the first packet fails and the receiver is not able to decode this entire packet. Contrary to the first packet, the second packet is transmitted successfully (c) and only the preambles  $P_{2,1}$  and  $P_{2,5}$  were not correctly decoded.



**Fig. 1.** Examples of how chip errors in the preamble affect packet losses.

### 3 Analysis of Packet Synchronization Errors in IEEE 802.15.4

This section analyzes the packet synchronization errors in IEEE 802.15.4. Our analysis focuses on environments that are free of interference. First, we show that in such environments most packet errors in IEEE 802.15.4 are due to synchronization failures while packet errors caused by symbol errors in the payload are rare. Then, our analysis deepens on two key error patterns in synchronization: (i) the number of preamble symbols that are wrong at the receiver and (ii) the number of chip errors in the received symbols of the preamble. The insights of this analysis will serve as the basis for the development of our novel estimator BLITZ in the next section.

#### 3.1 Experimental Setup

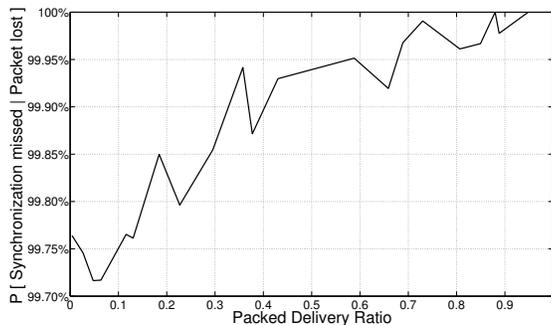
For our analysis, we conduct experiments with a software-based implementation of IEEE 802.15.4. As hardware platform, we use the USRP software-defined radio from Ettus Research. For the software, we use a slightly optimized version of the UCLA IEEE 802.15.4 implementation [13] that runs on the GNU Radio framework. We perform multiple tests in various indoor environments which are summarized as *cable*, *static* and *mobile* experiments. In the *cable* experiments, sender and receiver are connected by a shielded 60 cm coaxial cable with a 30 dB attenuator. The *static* experiments correspond to scenarios in which a stationary sender and receiver communicate over omni-directional antennas. The *mobile* experiments are similar to the static scenarios 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 bit 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 respectively. The true packet delivery ratio (PDR) of a link 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 which is smaller than the channel coherence time<sup>2</sup> when moving the receiver at maximum  $v = 1$  cm/s and at a frequency of 2.4 GHz. 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 hence relatively conservative with regard to the mobility.

### 3.2 Analysis

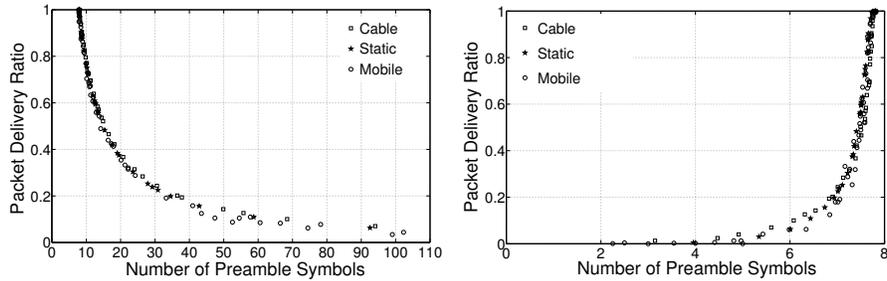
**Root cause of packet errors.** There are two reasons why packets in wireless networks, and in particular in IEEE 802.15.4 networks, may get lost: (i) The receiver is not able to synchronize on the preamble and start of frame delimiter of the frame and misses the entire frame or (ii) the receiver is able to synchronize on the frame but decoding errors occur in the received symbols of the payload such that the frame check sequence (FCS) is incorrect and the frame is discarded. Our experiments clearly show that in the absence of interference, the dominating cause of packet errors in IEEE 802.15.4 is the synchronization failure. Figure 2 shows experimentally the probability that a packet is lost because of synchronization failures. This probability is larger than 99.7% across the entire range of PDR values. Since synchronization failures are dominating, the best place to analyze packet errors is therefore at the synchronization phase in the preamble. Note that larger packet sizes may increase the probability of packet losses due to erroneous frame check sequences, however even for the maximum packet size of 127 bytes, missed packets caused by synchronization failures remain the dominating root cause of errors.



**Fig. 2.** Probability of a synchronization failure given that the packet is lost.

<sup>2</sup> The coherence time is the time duration over which the channel impulse response is considered to be not varying and is approximately  $\frac{1}{4D}$ , where  $D$  is the Doppler spread.

**Symbol error pattern in the preamble.** Looking at errors in the preamble, we analyze first how many preamble symbols a receiver misses/receives on average before it successfully receives a packet. Figure 3(a) shows the distribution of the PDR versus the average number of preamble symbols that a receiver decodes before synchronizing on a frame for the cable, static and mobile scenarios. We see that the average number of decoded preamble symbols is not affected by the type of environment and link conditions. Another observation is that the average number of decoded preamble symbols can be much larger than eight (the fixed number of preamble symbols in the IEEE 802.15.4 standard) for low quality links. For example, an average of 15 preamble symbols is obtained for a  $\text{PDR} = 50\%$  while it increases up to more than 100 when the PDR drops below 10%. The reason is that the receiver decodes preamble symbols from multiple packets before eventually synchronizing successfully on a frame. Overall, the distribution is monotonic and the average number of decoded preamble symbols per received packet is clearly an indicator of the link quality. However, the correlation is not conclusive. For example, the Pearson product-moment correlation coefficient which provides a measure of the strength of linear dependence between two variables as a value between  $+1$  and  $-1$  is here  $r = -0.5585$ , which indicates a relatively low linear correlation between the two metrics<sup>3</sup>.



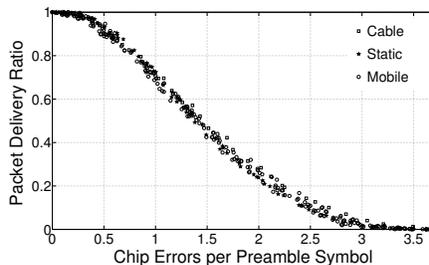
(a) Average number of decoded preamble symbols per successfully delivered packet. (b) Average number of consecutively decoded preamble symbols per transmitted packet.

**Fig. 3.** PDR versus the average number of received preamble symbols (a) per successfully decoded packet and (b) per sent packet.

In a second step, we analyze the number of consecutively decoded preamble symbols per transmitted packets. An average of this number versus the PDR is visualized in Fig. 3(b). Again, the distribution is unaffected by the type of scenario. The Pearson correlation coefficient is here  $r = 0.7617$  indicating a higher linear correlation for the average number of consecutively decoded preamble symbols, but for all links with a PDR above 20%, the average number of decoded symbols lies in a narrow, not strictly well distinctive range of values between 7 and 8.

<sup>3</sup> Values close to 0 refer to a low correlation while values close to  $+1$  and  $-1$  represent a high correlation.

**Chip error pattern in the preamble.** Figure 4 shows the distribution of chip errors per preamble symbol versus the PDR. As before, the environmental conditions do not affect the distribution. The Pearson correlation coefficient is  $r = -0.9650$ , showing the highest correlation for symbol error patterns in the preamble.



**Fig. 4.** Correlation of average chip errors per preamble symbol at the receiver and PDR.

**Summary of results.** In summary, we conclude that the dominating root cause of packet errors in IEEE 802.15.4 are synchronization failures. Looking at synchronization failures is therefore well suited to observe packet error behavior. Across cable, wireless static and mobile settings, the number of missed symbols and the number of chip errors in the synchronization phase both correlate well with the PDR. However, the correlation of missed preamble symbols to the PDR is not as distinctive and hence not well suited to model the PDR. In contrast, the average number of chip errors in the preamble has a high linear correlation to the PDR. To model the link quality with BLITZ, we therefore exclusively rely on chip errors in the preamble symbols.

## 4 BLITZ: Preamble-based Link Quality Estimation

This section presents BLITZ, our novel link quality estimator that relies on chip errors in the preamble to estimate the PDR of wireless links in point-to-point communication. To meet the requirements of accuracy and stability of the quality estimation, BLITZ operates at 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 short-term fluctuations of the estimation method.

### 4.1 Instantaneous PDR

In a first step, BLITZ estimates an instantaneous PDR after the reception of the preamble of packet  $k$  as

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

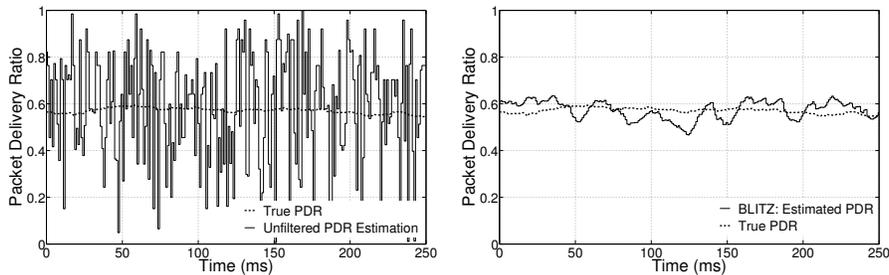
where  $P_{k,j}[i]$  is a vector containing the 32 chips of the  $j$ -th received preamble symbol of packet  $k$  for  $i = 1, 2, \dots, 32$ ,  $P[i]$  denotes a vector with the correct chips of the known preamble symbol,  $\oplus$  is the exclusive or operator, and  $|\mathcal{S}_k|$  is the number of received preamble symbols for packet  $k$ . The function  $g(\cdot)$  models the empirical distribution of the PDR versus chips errors per preamble symbols as shown in Figure 4. In this work, we fit a polynomial of the 5-th degree to the distribution according to

$$g(p) = p_0 p^5 + p_1 p^4 + p_2 p^3 + p_3 p^2 + p_4 p + p_5$$

while minimizing the root square of the error. The obtained parameters of the fit are  $p_0 = 0.016$ ,  $p_1 = -0.33$ ,  $p_2 = 2.41$ ,  $p_3 = -7.26$ ,  $p_4 = 8.83$ ,  $p_5 = -3.24$  resulting in a root mean square error below 3% across the entire distribution<sup>4</sup>.

While the instantaneous PDR provides a very fast estimation of the link quality, it is subject to high fluctuations as shown in Figure 5(a). In order to provide a more stable link quality metric, BLITZ further averages and filters consecutive instantaneous PDR estimations as described next.

## 4.2 Averaged and Filtered PDR



(a) Instantaneous PDR: high fluctuations.

(b) Filtered PDR: low fluctuations.

**Fig. 5.** BLITZ: Fluctuation of the instantaneous (left) and filtered (right) PDR estimation on a wireless static link with a true PDR of approximately 55%.

A classical approach to increase the stability of a metric is to weight sequential estimations in form of a weighted moving average. For example, the estimator in [6]

<sup>4</sup> Note that we use a polynomial of the 5-th degree to understand the best performance that an estimator like BLITZ may achieve. A polynomial of a smaller degree may alternatively be used in real-world deployments at the cost of a marginal error increase since the underlying empirical distribution is close to linear.

makes use of this technique to increase the stability of packet count statistics estimators. BLITZ applies a similar approach in order to smooth consecutive estimations of the instantaneous PDR. For this reason, BLITZ performs a low-pass filtering of the weighted average over a window  $w$  of consecutive link estimations. Suppose  $\mathcal{PDR}_{\text{inst}}(k)$  is the set of the past  $l + 1$  estimations of the instantaneous PDR at the position  $k$ . Let further  $PDR_{\text{inst}}(k-l), PDR_{\text{inst}}(k-l+1), \dots, PDR_{\text{inst}}(k-1), PDR_{\text{inst}}(k) \in \mathcal{PDR}_{\text{inst}}(k)$  be the past  $l + 1$  estimations. Then, the weighted average  $\text{wa}(k)$  over these recent  $l + 1$  estimations at the position  $k$  is calculated as

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

with the weighting factors  $\beta_m$  such that  $\sum_{m=0}^l \beta_m = 1$ . 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  $\alpha \in [0, 1]$  controls the smoothness. For example, a small factor  $\alpha$  gives more importance to current link behavior. Finally the  $k$ -th estimation is obtained as

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

The benefits of the averaging and filtering are illustrated in Figure 5 (b). The figure shows the resulting estimation error after filtering and weighting the estimations. With the parameters  $l, \beta_m$ , and  $\alpha$ , the estimation window can be changed to tune the reactivity of the estimator. In this work we set  $l = 6$ ,  $\beta_0 = 0.3, \beta_1 = 0.2, \beta_{2,\dots,6} = 0.1$ , and  $\alpha = 0.9$  as it provides the best trade-off in our experiments.

## 5 Evaluation and Comparison to Existing Link Quality Estimators

In this section, we show how BLITZ performs under different link conditions. To assess its performance, we compare BLITZ to competitive estimators from the literature. We consider a wide class of estimators ranging from estimators that rely on packet count statistics, signal strength, and chip errors in the payload.

### 5.1 Considered Estimators

We compare the performance of BLITZ with five estimators:

**ETX [7]:** ETX is an estimator that relies on packet count statistics. By sending broadcast probes at an average period  $\tau$  within a window of  $w$  seconds, each node can estimate the probe reception rate. The PDR is calculated in one direction as  $PDR_{\text{ETX}} = \frac{\text{probes}(t-w,t)}{w/\tau}$ , where  $\text{probes}(t-w, t)$  is the number of probe packets received during the window  $w$ , and  $w/\tau$  is the number of probes that should have been received. We set  $\tau = 1$  second and  $w = 10$  seconds as proposed in [7].

**WMEWMA [6]:** Like BLITZ, this estimator uses low pass filtering and calculates a weighted moving average over sequential estimations. The estimator relies on packet count statistics from regular data frames. In our implementation, we consider a rate of one frame per second for the link monitoring frames. The PDR is calculated as  $PDR_{\text{WMEWMA},i} = \alpha PDR_{\text{WMEWMA},i-1} + (1 - \alpha) \text{WMEWMA}_i$ . As proposed by Baccour et al. in [14], we set the smoothing factor  $\alpha = 0.6$  and calculate the instantaneous PDR ( $\text{WMEWMA}_i$ ) over a window size of five received packets.

**SNR [5]:** The SNR-based estimator relies on the SNR at the receiver to estimate the PDR. This metric is shown to be more accurate than simply relying on the RSSI [1]. An a priori known correlation between the average SNR of the packet and the PDR is used to estimate the link quality. This correlation is approximated by a linear fit. Calibrated on our platform, the correlation fit is given by  $PDR_{\text{SNR}} = 0.12 \text{ dSNR} - 1.7$ , where dSNR is the average signal power of a packet divided by the noise floor around the packet in dB.

**Four-Bit [9]:** Four-Bit is closely related to WMEWMA but applies an additional filtering step. For unicast transmissions the value FourBit is calculated as  $\text{FourBit}_i = \alpha \text{FourBit}_{i-1} + (1 - \alpha) \left( \frac{1}{\text{WMEWMA}_i} - 1 \right)$ , and the PDR is estimated as a function of  $PDR_{\text{Four-Bit},i} = \frac{1}{1 + \text{FourBit}_i}$ .

**CEPS [2]:** Similar to BLITZ, CEPS relies on chip errors to estimate the PDR. However, CEPS relies on the payload symbols whereas BLITZ models the errors from the preamble. As proposed in [2], the correlation between the chip errors and the PDR is approximated by a linear fit. In this case, the PDR is calculated as  $PDR_{\text{CEPS}} = 1 - \frac{\text{CEPS}}{\text{Chiplimit}}$ , where CEPS is the average number of chip errors per payload symbol and Chiplimit is a threshold that we calibrate to 3.44 from our measurements. For  $\text{CEPS} > \text{Chiplimit}$ ,  $PDR_{\text{CEPS}}$  is set to zero.

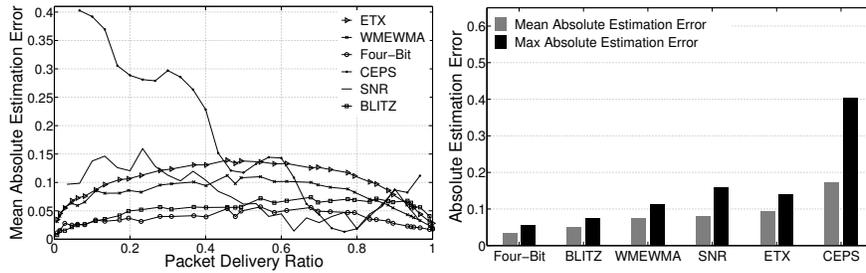
## 5.2 Evaluation Methodology

To ensure a systematic and reproducible evaluation of the estimators, we follow an evaluation methodology in which we transmit and record real packets over radios while the estimators are implemented in Matlab and evaluated offline by reproducing the recorded traces in Matlab. This approach has the advantage that we can replay the exact same packet traces that were recorded under identical link conditions for all estimators, making a direct comparison possible. Furthermore, this approach has more flexibility and allows determining the true PDR at a particular time from observation windows that encompass historical but also future packet events which would not be possible when determining the accuracy of the estimator in real-time.

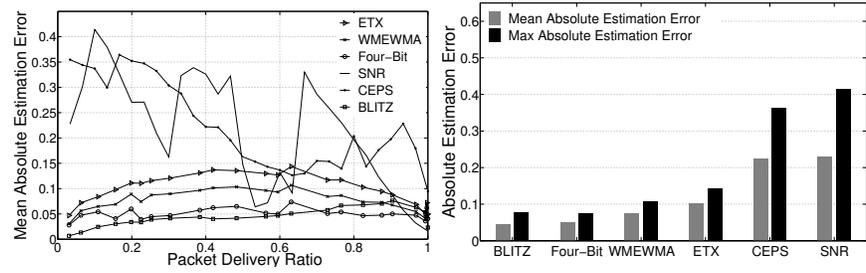
To evaluate the accuracy of an estimator, we use the absolute error. The absolute error at packet  $k$  is defined as

$$\text{absolute error}(k) = |PDR_{\text{true}}(k) - PDR_{\text{estimated}}(k)|,$$

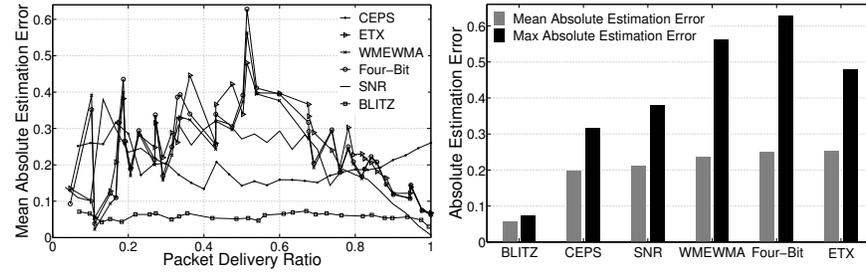
where  $PDR_{\text{true}}(k)$  is the true PDR at the  $k$ -th packet and  $PDR_{\text{estimated}}(k)$  is the PDR estimated by the link quality estimator at the  $k$ -th packet. The true PDR is assessed by counting the number of correctly received packets over a sliding estimation window of size  $w$  centered at packet  $k$ . The window size  $w$  is set to 100 packets for all experiments.



(a) Cable.



(b) Static.



(c) Mobile.

**Fig. 6.** Comparison of mean estimator error for (a) cable, (b) static, and (c) mobile links.

This value provides a reasonable amount of packets to assure that the average PDR over the window has converged enough to the expected mean. Furthermore, it is assured that the window size is small enough such that the coherence time of the channel is larger than the window time, ensuring that the channel is always in steady state over the whole observation window.

### 5.3 Evaluation of Estimator Accuracy

Figure 6 presents the results of the estimation error comparison in the (a) cable, (b) static, and (c) mobile settings. The left figures show the mean absolute estimation error versus the true PDR. The right figures show a ranking from left (lowest error) to right (highest error) of the mean estimation error averaged over all PDR values.

The cable scenario is characterized as a very stable link condition because unpredictable and uncertain factors (e.g. small scale fading) as observed for wireless links do not occur. Therefore, the cable experiments lead to optimal link conditions for any estimator and the results can be regarded as a benchmark for the best-case performance. Not surprisingly, the estimators that rely on packet count statistics (Four-Bit, WMEWMA, and or ETX) perform quite well in this case. Four-Bit shows the best performance as a double filtered packet statistic based estimator with an average mean absolute estimation error of 3.5 %. Remarkable is the performance of BLITZ that even shows smaller errors than packet statistic approaches like WMEWMA or ETX over cable. SNR-based estimator is expected to perform well in cable scenarios. This is confirmed by the average mean estimator error of 7.9 %, leading to the 4-th rank. Interestingly, the performance of CEPS with an average mean estimation error of 18 % is the worst. Note however the large difference in error for links with a PDR above and below 50 %. CEPS tends to be better at estimating good quality links. This result is expected considering that CEPS only assesses the chip errors in the payload of packets which tend to be rare in low quality links since most packets are lost due to synchronization failures.

Turning to the wireless static link scenario (Figure 6 (b)), we obtain a new order in the ranking. BLITZ is the winner and shows the smallest average error of 4.7 %. Since the link can still be assumed as relatively stable, the packet count statistics based approaches should perform further on well. The ranking of these estimators reflect this as the second, third and the fourth best estimators are Four-Bit, WMEWMA and ETX with average mean estimation errors below 10 %. CEPS still does not show superior results compared to packet count statistics based estimators while it now slightly beats SNR-based estimator.

Last, and most interesting are the wireless mobile scenarios. These scenarios are clearly the most challenging for all estimators as the link conditions may vary quickly and the estimators must hence be able to estimate the link quality rapidly. Since the packet count statistics estimators are the slowest, we can expect that they will perform here worst. This assumption is confirmed looking at the results of the mobile scenario in Figure 6 (c). ETX, Four-Bit, and WMEWMA are this time ranked last. Compared to the static scenarios, they have now a much higher absolute estimation errors between 23 % and 26 %. The average absolute estimation error of BLITZ remains low at 5.7 %. Despite the weighted average over the last six estimations at the cost of reactivity, BLITZ is still reactive enough to outperform all other estimators. Between BLITZ and the packet count statistics estimators are ranked CEPS and SNR-based estimator with average mean estimation errors of 19.7 % and 20.1 %, respectively. Note that the mobility experiments have a relatively low node speed of maximum 1 cm/s for the sake of comparison. We intentionally kept the node mobility low so to calculate the true PDR. We expect an even more pronounced performance gap in favor of BLITZ for settings with higher mobility.

#### 5.4 Evaluation of Estimator Reactivity

Another important aspect of link quality estimation is reactivity. By reactivity, we refer to the ability of the estimators to quickly determine a new estimation of the PDR. An overview of the temporal reactivity for all considered estimators is given in Table 1

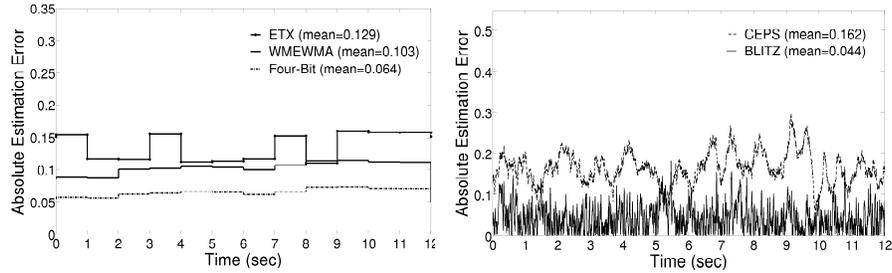
assuming 26 bytes packets. BLITZ is able to estimate a new PDR value after 0.13 ms, the duration of one preamble. Depending on whether the signal strength is calculated over the preamble or over the entire packet, SNR has reactivities of 0.13 or 1 ms respectively. CEPS has reactivity of 1 ms. The packet statistic based estimators provide a new estimation after 10 s (ETX) and  $\geq 5$  s (WMEWMA and Four-Bit). Table 1 further gives the mean error converging time of the estimators as defined as the average time it takes for the estimation error to drop below an absolute error of 0.15 for links with PDR equal to 50%.

**Table 1.** Reactivity of different estimators assuming 26 bytes packets.

Estimator	Input	Window size	Time for first estimation	Mean error convergence time
BLITZ	chip errors	$\geq 1$ preamble	0.13 ms	1.9 ms
ETX	packet statistics	10 packets	10 s	10 s
WMEWMA	packet statistics	$\geq 5$ packets	5 s	6.7 s
SNR	signal strength	1 preamble or 1 packet	0.13 or 1 ms	0.13 or 1 ms
Four-Bit	hybrid	$\geq 5$ packets	5 s	6.1 s
CEPS	chip errors	1 packet	1 ms	1 ms

### 5.5 Evaluation of Estimator Stability

A third important factor of the estimators is their stability. By stability, we refer to ability to provide a stable estimation of the PDR when the link conditions are static. Note that to some extent, this requirement is in contradiction to reactivity. Figure 7 illustrates the stability of the different estimators by showing the absolute error over a few seconds for a static link with true PDR of approximately 50%. While the packet statistics based estimators generally provide better stability, the stability of BLITZ and CEPS is still remarkable given their high reactivity.



(a) Packet statistic based link quality estimators (b) Chip error based link quality estimators

**Fig. 7.** Tracking over time of the absolute estimation error for a link with 50 % PDR for (a) ETX, WMEWMA, Four-Bit and (b) BLITZ, CEPS.

## 5.6 Summary of the Results

To conclude the evaluation in this section, we summarize the following main findings:

- In cable scenarios, i.e., when the stable link conditions for the estimators are optimal, BLITZ is competitive to the best-performing packet count statistic estimators.
- In static wireless scenarios, BLITZ has the best performance with an average PDR estimation error of 4.7 %.
- BLITZ is particularly suitable in mobile settings where it outperforms all other estimators by an absolute estimation error of 10 % to 20 % on average.
- Because BLITZ relies on chip errors in the preamble, it manages to estimate significantly more quickly and accurately the PDR compared to CEPS which estimates the PDR based on chip errors in the payload.
- BLITZ manages to estimate the PDR of low quality links accurately and within just a few milliseconds.

## 6 Limitations

BLITZ considers only physical layer information. This comes at some limitations. First is the issue of multiple nodes transmitting data to one receiver. In its current implementation, BLITZ does not infer which is the transmitting node on a per-frame basis for data with failed synchronization. Potential workarounds include introducing a form of physical address, using timing information, or signal fingerprints to differentiate between the different transmitters. A second issue is the interference originated by collisions. Since collisions usually occur because two or more stations start to transmit at the same time, it tends to cause further physical synchronization and chip errors. This work analyzed only scenarios without interference and further research is necessary to understand the performance of BLITZ with multiple transmitters in the same collision domain.

## 7 Related Work

The authors in [15] 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 [2] models the PDR from chip errors in the payload of successfully received packets. In contrast, BLITZ models the PDR from chip error measurements in the synchronization phase. We show in this paper that the approach of BLITZ is much more accurate and faster than CEPS because packet errors are more probable due to synchronization failures than due to chip/symbol errors in the payload.

The link quality indication (LQI) measurement [16] is a characterization of the quality of a received packet on CC2420 hardware chips. The RSSI LQI reports the received signal strength. However, to reduce the impact of narrowband interference inside the channel, the CC2420 also provides an LQI based on the average correlation value of the first 8 incoming symbols following the start frame delimiter (SFD) which presumably relates to the chip error rate as observed in the payload. Unfortunately, the LQI

is a proprietary hardware implementation for the CC2420 chip and there is no exact description presented in the datasheet that could be used for comparison on other chips or on software-defined radios. According to various independent studies [17–19], the LQI on the CC2420 further suffers trustworthiness due to variations of the correlation coefficient between LQI and PDR. Relating the LQI directly to a PDR performance therefore results in unreliable and unstable predictions.

Chen et al. propose in [20] to leverage codes to estimate the number of bit errors in the payload and model the packet delivery. The proposed technique focuses on WiFi transmissions where bit errors in the payload are common. This approach is not well suited for IEEE 802.15.4 communication because packets rarely get lost because of bit errors in the payload but rather due to synchronization failures in the preamble as we show in Section 3. Halperin et al. [5] propose a model that relies on channel state information measurements to predict 802.11 packet delivery from wireless channel measurements. This model is specific to OFDM, and not applicable to DSSS communication.

## 8 Conclusions

We have explored various properties of preamble symbols to design a fast, accurate and stable link quality estimator. With a software-defined radio based implementation of IEEE 802.15.4, we showed that chip errors in the preamble symbols serve as a good indicator to predict a link’s PDR in point-to-point communication. Our estimator, BLITZ, is a chip error-based estimator that weights and filters sequential estimations on a per-packet level. BLITZ proved to be at least three times faster than state-of-the-art rapid estimators and more accurate than other link quality estimators under various wireless channel conditions as static, mobile, poor and good quality links. In average, BLITZ showed half the absolute estimation error (5%) of the best performing packet statistics-based link quality estimators and four times less error than signal strength and payload chip error-based estimators. Especially in dynamic environments like mobile scenarios, BLITZ is particularly superior due to its ability to react to fast changing link qualities and to exploit information from packets that fail at the frame synchronization phase.

To the best of our knowledge, there exist no commercial hardware radio chips that report the number of chip errors in the preamble symbols. The application of BLITZ is therefore currently limited to software-defined radios in which chip-level information is accessible from software. However, we do not expect any fundamental challenge in implementing such a chip error indicator on radio chips. Indicators like the RSSI or the LQI, which are similar in complexity, have for example already been integrated on the CC2420 low-power radio chips. We hope that our work will motivate chip manufacturers to implement chip error-based indicators in next-generation radio chips in order to provide higher-layer applications with a powerful metric to estimate quickly and accurately the wireless link quality.

## References

1. Baccour, N., Mottola, L., Niga, Z., Boano, Alves, M.: Radio Link Quality Estimation in Wireless Sensor Networks: a Survey. *ACM Transactions on Sensor Networks* (2012)

2. Heinzer, P., Lenders, V., Legendre, F.: Fast and Accurate Packet Delivery Estimation based on DSSS Chip Errors. In: IEEE INFOCOM 2012, Orlando, Florida, USA (March 2012)
3. Aguayo, D., Bicket, J., Biswas, S., Judd, G., Morris, R.: Link-level Measurements from an 802.11b Mesh Network. In: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications. ACM SIGCOMM '04, New York, NY, USA, ACM (2004)
4. Reis, C., Mahajan, R., Wetherall, D., Zahorjan, J.: Measurement-based models of delivery and interference in static wireless networks. In: in SIGCOMM Computer and Communications Review. (2006)
5. Halperin, D., Hu, W., Sheth, A., Wetherall, D.: Predictable 802.11 packet delivery from wireless channel measurements. SIGCOMM Comput. Commun. Rev. **40** (August 2010)
6. Woo, A., Culler, D.: Evaluation of efficient link reliability estimators for low-power wireless networks. Technical Report UCB/CSD-03-1270, EECS Department, University of California, Berkeley (2003)
7. De Couto, D.S.J., Aguayo, D., Bicket, J., Morris, R.: A high-throughput path metric for multi-hop wireless routing. In: Proceedings of the 9th ACM International Conference on Mobile Computing and Networking (MobiCom '03), San Diego, California (Sept. 2003)
8. Cerpa, A., Wong, J.L., Potkonjak, M., Estrin, D.: Temporal properties of low power wireless links: modeling and implications on multi-hop routing. In: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing. MobiHoc '05, New York, NY, USA, ACM (2005)
9. Fonseca, R., Gnawali, O., Jamieson, K., Levis, P.: Four-bit wireless link estimation. In: Proceedings of the Sixth Workshop on Hot Topics in Networks (HotNets VI). (2007)
10. Baccour, N., Koubaa, A., Youssef, H., Ben Jamaa, M., do Rosario, D., Alves, M., Becker, L.: F-lqe: A fuzzy link quality estimator for wireless sensor networks. In Silva, J., Krishnamachari, B., Boavida, F., eds.: Wireless Sensor Networks. Volume 5970 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2010)
11. Boano, C., Zuniga, M., Voigt, T., Willig, A., Romer, K.: The triangle metric: Fast link quality estimation for mobile wireless sensor networks. In: Computer Communications and Networks (ICCCN), 2010 Proceedings of 19th International Conference on. (August 2010)
12. IEEE: IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)
13. Schmid, T.: Gnu radio 802.15.4 encoding and decoding. Technical report, UCLA NESL (2005)
14. Baccour, N., Koubaa, A., Ben Jamaa, M., Youssef, H., Zuniga, M., Alves, M.: A comparative simulation study of link quality estimators in wireless sensor networks. In: Modeling, Analysis Simulation of Computer and Telecommunication Systems, 2009. MASCOTS '09. IEEE International Symposium on. (Sept. 2009)
15. Qin, Y., He, Z., Voigt, T.: Towards accurate and agile link quality estimation in wireless sensor networks. In: Med-Hoc-Net, IEEE (2011)
16. Chipcon: CC2420 - 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver. (2009)
17. Zhao, J., Govindan, R.: Understanding packet delivery performance in dense wireless sensor networks. In: Proceedings of the 1st international conference on Embedded networked sensor systems. SenSys '03, New York, NY, USA, ACM (2003)
18. Ganesan, D., Krishnamachari, B., Woo, A., Culler, D., Estrin, D., Wicker, S.: Complex behavior at scale: An experimental study of low-power wireless sensor networks. Technical report, Citeseer (2002)
19. Srinivasan, K., Dutta, P., Tavakoli, A., Levis, P.: An empirical study of low-power wireless. ACM Trans. Sen. Netw. **6**(2) (March 2010)
20. Chen, B., Zhou, Z., Zhao, Y., Yu, H.: Efficient error estimating coding: feasibility and applications. In: ACM SIGCOMM Computer Communication Review. Volume 40., ACM (2010)