# Link Quality Estimation of Cross-Technology Communication

Jia Zhang[1], Xiuzhen Guo[1], Haotian Jiang[1], Xiaolong Zheng[2], Yuan He[1]
[1]School of Software & BNRist, Tsinghua University, China
[2]School of Computer Science, Beijing University of Posts and Telecommunications, China
{j-zhang19, guoxz16, jht19}@mails.tsinghua.edu.cn, zhengxiaolong@bupt.edu.cn
heyuan@mail.tsinghua.edu.cn

*Abstract*—**Research on Cross-technology communication (CTC) has made rapid progress in recent years, but how to estimate the quality of a CTC link remains an open and challenging problem. Through our observation and study, we find that none of the existing approaches can be applied to estimate the link quality of CTC. Built upon the physical-level emulation, transmission over a CTC link is jointly affected by two factors: the emulation error and the channel distortion. We in this paper propose a new link metric called C-LQI and a joint link model that simultaneously takes into account the emulation error and the channel distortion in the process of CTC. We further design a light-weight link estimation approach to estimate C-LQI and in turn the PRR over the CTC link. We implement C-LQI and compare it with two representative link estimation approaches. The results demonstrate that C-LQI reduces the relative error of link estimation respectively by 46% and 53% and saves the communication cost by 90%.**

*Index Terms*—**cross technology communication, link quality estimation**

## I. Introduction

Cross-technology communication (CTC) enables direct communication among heterogeneous devices that follow different standards. CTC not only creates a new way for interoperation and exchange among wireless devices, but also enhances the ability to manage wireless networks.

Research on CTC has received a lot of attention in the past few years. Early works mainly utilize packet-level metrics, such as packet length [1], transmission timing [2, 3], transmission power [4–11], etc. to carry information, as is called packet-level CTC. The state of the arts in this area is physical-level CTC [12–14]. The core idea is physical emulation, namely to directly emulate the desired signals of the receiver with the sender's radio. Compared to packet-level CTC, physical-level CTC can achieve much better performance in terms of communication throughput. Given the rapid progress in CTC, how to manage and utilize the wireless links created by CTC becomes an increasingly important problem.

Recent works on CTC already pay more or less attention to the quality of CTC. For instance, it is reported in [15] that the packet reception rate (PRR) of WEBee varies between 45% and 55%. It requires up to 6 retransmissions to get 99% reliable transmission of a packet. WIDE [16] enhances the reliability of CTC based on WEBee and achieves a PRR from 80% to 90%. Under changeable channel condition and mobility, the frame reception ratio of BlueBee [17] ranges from 73% to 99%. Those results reflect that the quality of CTC links is indeed a dynamic factor. When CTC links are included in a wireless network, the quality of those links concerns many aspects of network operation, such as link selection [18], transmission strategy [19–22], and routing structure [23–26]. But how to estimate the quality of a CTC link remains an open problem.

Link quality estimation is a classic problem in wireless networks. In terms of the metric to measure the link quality, there have been many different approaches, which fall into three main categories: (1) Raw physical-level indicator directly obtained from the radio, e.g. RSSI [27], SNR [28], etc; (2) Metrics derived from physical-level measurement, such as LQI and CSI [29]; (3) Packet-level indicators. For example, PRR [30] measures the rate of successfully received packets. ETX [18, 31] quantifies the transmission efficiency on a link by the expected number of transmissions to successfully send a packet. Some existing works combine the above two or three types of metrics to jointly estimate link quality, e.g. 4BitLE [32].

Based on our observation and study, we find that none of the existing approaches can be applied to estimate the link quality of CTC. The reason is that a CTC link is fundamentally different with a conventional wireless link. Built upon the physical-level emulation, transmission over a CTC link is jointly affected by two factors: the emulation error and the channel distortion. Different emulated symbols under different channel conditions have different symbol error rates (SER). The physical-level link metrics can't completely characterize the process of CTC, while the packet-level metrics overlook the differences at the lower layers of the protocol stack. Using the existing metrics to estimate the CTC link quality generally means poor accuracy and uncontrollable overhead.

In order to address the above problem, we in this paper propose a new metric called C-LQI, which is defined as the expected probability for a symbol to be correctly decoded by the receiver of a CTC link. C-LQI is built upon a joint link model that simultaneously takes into account the emulation error and the channel distortion in the process of CTC. In order to estimate C-LQI, we design a light-weight link estimation approach: The CTC sender periodically sends to

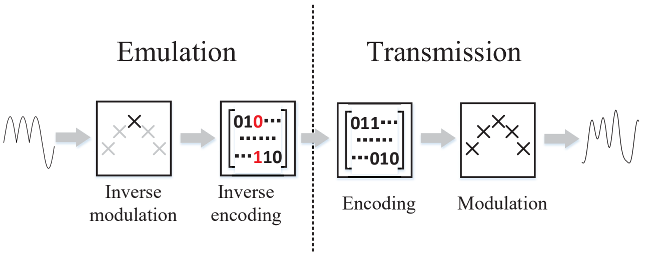


Fig. 1: The typical process of physical emulation

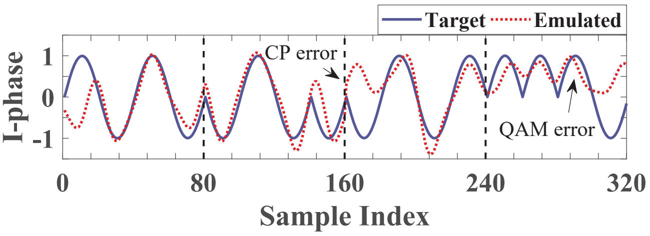


Fig. 2: The target signal waveform and the emulated signal waveform

the CTC receiver probe frames, in which selected symbols are embedded. According to the reception status returned from the receiver, the sender can estimate C-LQI of the CTC link. Given the composition of packet payload, C-LQI can be utilized to estimate the PRR over the CTC link.

Our contribution can be summarized as follows:

- To the best of our knowledge, C-LQI is the first metric of CTC link quality. The joint link model underneath C-LQI comprehensively characterizes impacting factors of a CTC link, by taking both the emulation error and the channel distortion into account.
- We address a series of technical challenges in estimating the quality of a CTC link, including probe frame composition, channel parameterization, and PRR calculation. We further address several issues that concern the applicability of our approach in practice.
- We implement C-LQI on USRP N210 and conduct extensive experiments under varied settings. The results demonstrate that C-LQI is highly accurate and lightweight. When using C-LQI to estimate the PRR of a CTC link, it achieves 4.18% relative error in average, which is respectively 46% and 53% lower than that of two representative link estimation approaches.

The rest of the paper is organized as follows. Section II discusses important observation of the CTC link, which motivates our research in this paper. Section III introduces the definition of C-LQI and presents the joint link model. Section IV elaborates on the link estimation approach. The implementation and evaluation results are presented in Section V. Section VI discusses the related work. We conclude our work in Section VII.

## II. Observation

This section first analyzes the intrinsical emulation error in physical-level CTC. Physical-level CTC mainly utilizes the signal of the sender to emulate the standard waveform or phase shift of the receiver. We give an example to illustrate that emulating different target signals produces different emulation errors. We find that when the emulated signals pass through the wireless channel, the actual distribution of decoding errors is not the same with that of the emulation errors. In the end of this section, we theoretically explain this observation.

### A. *Errors of physical emulation*

According to Figure 1, physical-level CTC causes emulation errors in the transmitted signal. The target signal is first fed into the emulation module, including the inverse modulation and the inverse channel encoding, to obtain the corresponding payload. Then the payload is encoded and modulated to obtain the desired emulated signal.

As shown in Figure 1, in order to get the emulated signal, the corresponding target signal must be used as an input for emulation. Different target signals correspond to different emulated signals, and have different emulation errors due to the distortion in the emulation process. We use WEBee as an example. Since ZigBee uses Offset Quadrature Phase-Shift Keying (OQPSK) modulation and WiFi uses Orthogonal frequency-division multiplexing (OFDM) and Quadrature amplitude modulation (QAM), the WiFi transmitter cannot perfectly emulate ZigBee signals. Figure 2 shows the target and emulated signal waveform of ZigBee symbol 7. Due to the QAM modulation and the cyclic prefix of the WiFi transmitter, there are intrinsical emulation errors between the emulated signals and the target signals.

When the emulated signals are decoded by the ZigBee receiver, there exist chip errors in the decoded results. The reason is that the ZigBee receiver decodes the chip sequence according to the phase shifts between the sampling points in the waveform. The decoded result is shown in Figure 3. The emulated signals of different ZigBee symbols have different chip errors. Since the ZigBee receiver uses the Direct Sequence Spread Spectrum (DSSS) mapping table to decode chips into symbols, it can tolerate some chip errors. As long as the chip errors are within the decoding threshold, the received signal can still be decoded correctly.

### B. *Impact of wireless channels*

In general, when the chip errors introduced by the emulation are fewer, the emulated signals should be more accurate and there should be fewer decoding errors. Whereas, the actual performance of these emulated signal is not the case. We conduct experiments to observe the symbol error rates (SER) of all the symbols. The result is shown in Figure 5. We can find the SER of symbol 3 is the lowest. However, according to Figure 3, its emulation error is the highest. The reason is that besides the emulation errors introduced during the emulation process, the wireless channel also introduces random distortion

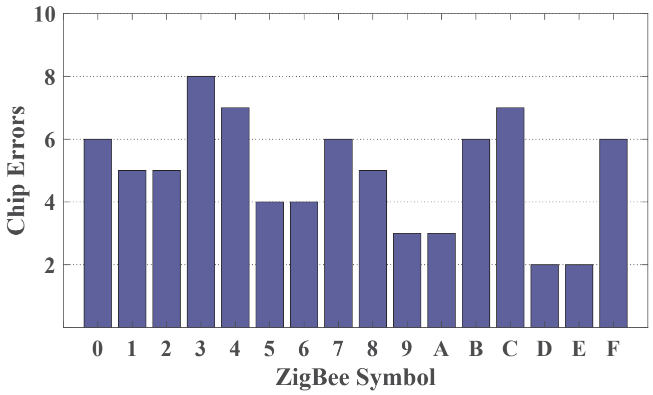


Fig. 3: The chip errors caused by emulation

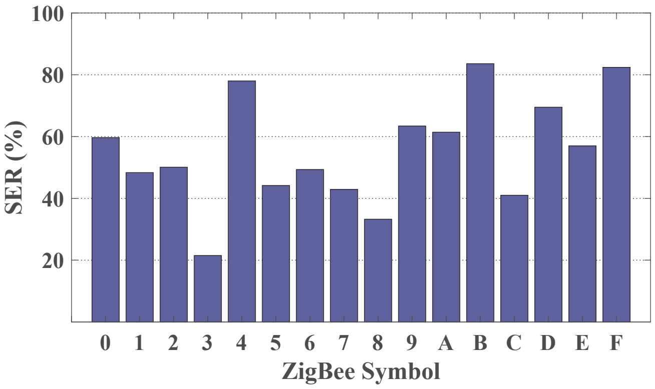


Fig. 5: The SER of each ZigBee symbol

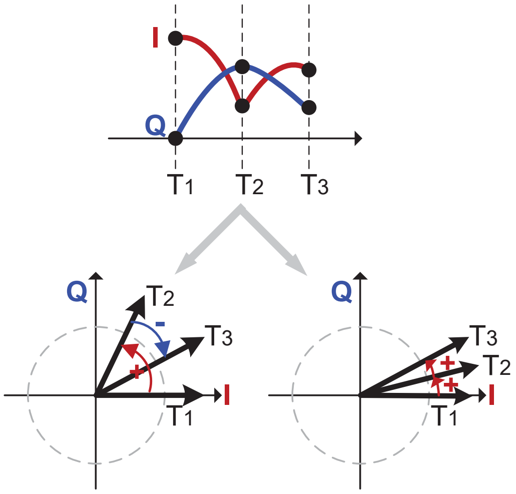


Fig. 4: Phase shifts are mapped to chips and the distortion makes the decoded results changed

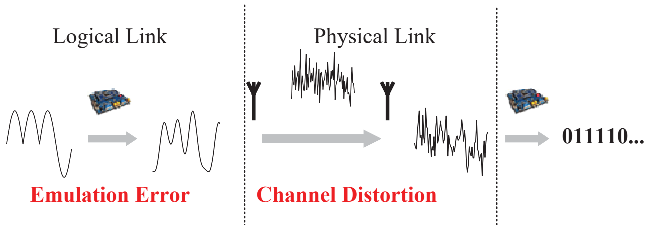


Fig. 6: The link model

on the signal waveform. Because the ZigBee receiver decodes the chip by judging whether the phase shift is greater than zero, the channel distortion affects the phase of each sampling point, and affects the decoding result.

We may further find that a larger absolute value of the phase shift is more robust against the random distortion in the wireless channel. Figure 4 shows an example. The upper subfigure shows a segment of the desired waveform, which is generated by the CTC sender and will be transmitted to the receiver through the channel. If perfectly received without any distortion, the waveform will be decoded into chips $'10'$ because the signs of phase shifts are “+-”, as shown in the lower left subfigure. There is channel distortion, however. Suppose there is a certain distortion on $T_2$, as shown in the lower right subfigure, the phase shifts are changed. Note that the phase shift between $T_1$ and $T_2$ has a larger absolute value than that between $T_2$ and $T_3$. Hence, we can see that the sign of the first phase shift is kept unchanged while the second is changed to “+”, which leads to a chip error.

Based on the above observation, we find that the decoding errors come from two sources, namely the emulation errors and the channel distortion. Both of them should be taken into account when estimating the quality of a CTC link.

## III. Model

From the observation, we find that proposing a metric is fundamental to estimate the CTC link quality, which is relative to the emulation errors and the channel distortion. We propose **C-LQI, which is defined as the expected probability for a symbol to be correctly decoded by the receiver of a CTC link.** We first present the joint link model to analyze and calculate C-LQI. In next section, we will further introduce the specific approach to estimate the link quality by using C-LQI.

This section first introduces the model of CTC links, including the logical link and the physical link. Secondly, we analyze the variation of phase shifts after the signals pass through the entire link. In the end of the section, according to the variation of phase shifts, we calculate C-LQI.

### A. Joint link model

Figure 6 shows our further analysis of the CTC link. The emulation process that introduces the emulation error is the logical link, and the wireless channel is the physical link. These two parts form the entire CTC link together.

To estimate the CTC link quality, we must consider the entire link. Both the emulation errors and the random distortion introduce the decoded errors. We use $S_r$ to represent the received signal. $S_i$ represents the initial signal. Then $E_e$ and $E_d$ represent the emulation errors and the errors from the random distortion, respectively. Their relationship is shown as follows:

$$S_r = S_i \cdot E_e \cdot E_d \tag{1}$$

That means the initial signal is effected by the emulation errors and the random distortion. When estimating the link quality, both of them should be considered.

### B. Variation of phase shifts

The influence of the channel on the sampling points in the waveform is random, and the corresponding phase changes effect on the sampling points are also random. In addition, different channels have different influences on the phase. The emulated phases of two consecutive sampling points in a symbol are represented as $p_1$ and $p_2$. The channel causes the phases to produce a random variation within the range of [-x,x]. We assume that the changed phases passing through the channel are $a_1$ and $a_2$, respectively. $a_1$ and $a_2$ follow the rules:

$$a_1 \sim U(p_1 - x, p_1 + x), a_2 \sim U(p_2 - x, p_2 + x), \tag{2}$$

Then we consider the probability distribution of $a_2 - a_1$. Let $X_1$ and $X_2$ be independent $U(0,1)$ random variables. Let $Y = X_1 - X_2$, and the cumulative distribution function of $Y$ is:

$$\begin{aligned} F_Y(y) &= P(Y \le y) \\ &= P\left(X_1 - X_2 \le y\right) \\ &= \begin{cases} \int_0^{1+y} \int_{x_1 - y}^{1} 1 dx_2 dx_1 & -1 < y < 0 \\ 1 - \int_y^1 \int_0^{x_1 - y} 1 dx_2 dx_1 & 0 \le y < 1 \end{cases} \\ &= \begin{cases} y^2/2 + y + 1/2 & -1 < y < 0 \\ -y^2/2 + y + 1/2 & 0 \le y < 1 \end{cases} \end{aligned} \tag{3}$$

With this conclusion, now we consider:

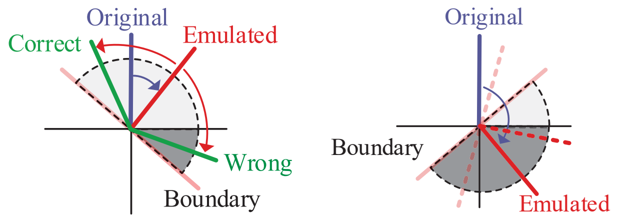
$$\frac{a_1 - p_1 + x}{2x} \sim U(0,1), \frac{a_2 - p_2 + x}{2x} \sim U(0,1) \tag{4}$$

Then the cumulative distribution function of $a_2 - a_1$ is:

$$\begin{aligned} F(y) &= P\left(a_2 - a_1 \le y\right) \\ &= P\left(\frac{a_2 - a_1 - (p_2 - p_1)}{2x} \le \frac{y - (p_2 - p_1)}{2x}\right) \\ &= F_Y(\frac{y - (p_2 - p_1)}{2x}) \end{aligned} \tag{5}$$

Note that the value range of $a_2 - a_1$ is from $p_2 - p_1 - 2x$ to $p_2 - p_1 + 2x$. So if the value of $y$ is out of the range, the value of $F(y)$ is 0 or 1.

According to Figure 7(a), the standard phase shift of two consecutive sampling points in a ZigBee signal is 90° as shown by the blue line. The emulation process makes it change and the phase shift of the emulated signal is shown by the line marked "Emulated". The wireless channel will cause the phase shift to change again. If the phase shift changes to below 0°, it will be wrongly decoded. In another case, if the phase shift of the emulated signal changes as Figure 7(b) shows, which has already been wrongly decoded. It still has the opportunity to be correctly decoded after passing through the channel. Moreover, even if all the phase shifts below 0° have been wrongly decoded, the opportunities of correctly decoding are different after passing through the channel. The closer to 0°, the more chance the phase shift is correctly decoded.



(a) The phase shift above 0° may vary to below 0° and be wrongly decoded

(b) The phase shift below 0° may vary to above 0° and be correctly decoded

Fig. 7: Phase shift changes within a certain range due to the channel

The above analysis explains why the actual chip error distributions differ from the emulation results. That is because their phase shifts are different, so the probabilities to be correctly decoded are different through the wireless channel.

In summary, in the case of knowing the phase shifts between the sampling points after emulation, different channel parameters correspond to different probability distributions of received phase shifts.

### C. Error calculation

After obtaining the probability distribution of each phase shift in each symbol, we consider the probability of correctly decoding of each phase shift. we use $C_i$ to represent the probability that the i-th chip is correctly decoded, then we have:

$$C_i = \begin{cases} F(0) & when\ standard\ chip\ is\ 0 \\ 1 - F(0) & when\ standard\ chip\ is\ 1 \end{cases} \tag{6}$$

When the standard chip corresponding to the phase shift is "0", the phase shift should be no larger than 0°. So the probability of correctly decoding of the chip is $P(a_2 - a_1 \le 0)$, which is $F(0)$. When the standard chip is "1", the phase shift should be larger than 0°. So the probability should be $P(a2 - a1 > 0)$, which is $1 - F(0)$.

In this way, we can get a set of probabilities $C_1, C_2 ... C_{30}$ for each symbol corresponding to $x$, representing the correct probability of the i-th decoded chip respectively. It should be noticed that the ZigBee receiver uses only 30 out of every 32 chips to decode symbols. The first and the last chips are not used.

Then we hope to take advantage of the correct probability of these 30 chips in each symbol to calculate the decoding probabilities of these symbols. We use $S_a$ and $S_b$ to represent the ZigBee symbol $a$ and the symbol $b$ respectively. $P(S_a \rightarrow S_b)$ is used to represent the probability that the emulated waveform of symbol $a$ is wrongly decoded into the symbol $b$ after passing through the wireless channel. We can get the following relationship:

$$P(S_a \rightarrow S_b) = \sum_{n=0}^{30} P(S_a \rightarrow S_b | n) \times P(n, S_a) \tag{7}$$

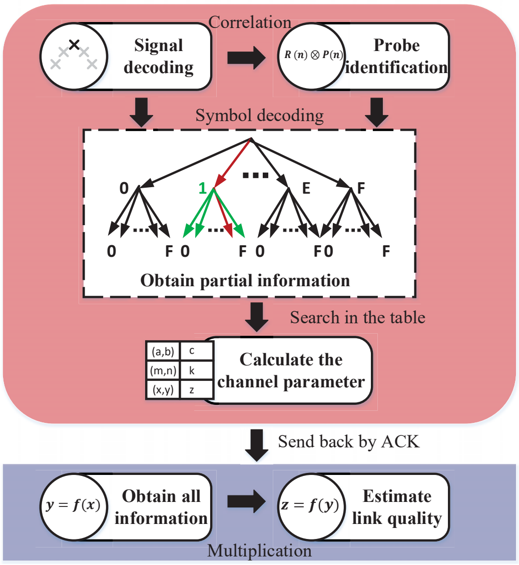


Fig. 8: The workflow of C-LQI

We use $P(n, S_a)$ to represent the probability that $n$ of 30 chips in the waveform of $S_a$ are wrongly decoded. $P(S_a \rightarrow S_b|n)$ is used to represent the probability that the emulation waveform of symbol $a$ is wrongly decoded into the symbol $b$ in the case that $n$ of 30 chips are wrongly decoded. If we can get all the $P(S_a \rightarrow S_b|n)$ and $P(n, S_a)$, then we can calculate the corresponding result by the above relationship. Now we have the standard chip sequences of 16 ZigBee symbols, so it is easy to get all $P(S_a \rightarrow S_b|n)$ by traversing.

On the other hand, $P(n, S_a)$ is related to all of 30 chip correct probability of symbol $a$, then we can calculate $P(n, S_a)$ by the following formula:

$$P(n, S_a) = \sum(1 - C_{i_1})(1 - C_{i_2})...(1 - C_{i_n})C_{i_{n+1}}...C_{i_{30}} \tag{8}$$

Where $\{i_1, i_2...i_n\}$ is the arrangement of $n$ elements taken from $\{i_1, i_2...i_{30}\}$. $C_i$ is the correct probability of the $i$th chip of the symbol $a$, which is determined by the channel parameter $x$.

In this way, we can get all the $P(n, S_a)$ as long as the value of the channel parameter $x$ is known. We can also obtain all the values of $P(S_a \rightarrow S_b)$ for any $S_a$ and $S_b$ utilizing $P(S_a \rightarrow S_b|n)$ and $P(n, S_a)$. Then We obtain the value of C-LQI.

## IV. Design

This section first introduces the workflow of C-LQI, including the collection of partial symbol decoding information, the calculation of the channel parameter $x$ and the estimation of the link quality. Without abusing notations, we also use C-LQI to denote the link estimation approach proposed in this paper. Then we have some discussion, including the cross-correlation module, the estimation of bidirectional CTC links, the applicability of C-LQI in co-existing networks and the generalizability of C-LQI.

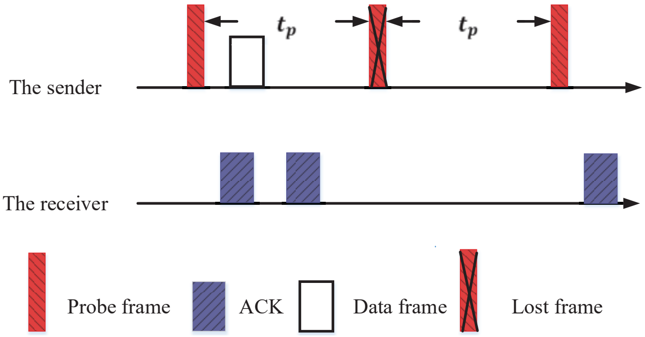


Fig. 9: The protocol design of C-LQI

### A. The workflow of C-LQI

The entire workflow of C-LQI is shown in Figure 8. The ZigBee receiver receives and decodes the probe frames, and then obtains partial symbol decoding information to calculate the channel parameter $x$. After obtaining the value of $x$, we can estimate all symbol decoding information utilizing the link model. At the end of the section, the information can be used to estimate link quality.

### B. Obtain partial information

In order to achieve the workflow, we first need to design the practical transfer protocol. As shown in Figure 9, the sender periodically sends probe frames to the receiver. This period is set to $t_p$. Once the receiver receives the probe, it immediately sends back an ACK containing the channel parameter $x$. The sender uses the channel parameter to estimate current link quality. After receiving the ACK, the transmitter starts to send normal data frames until the next probe frame should be sent. If the receiver doesn't receive the ACK, it doesn't send the ACK back. The receiver can know whether a probe frame is lost by the sequence number.

After designing the transfer protocol, we need to consider what information should be collected and used to calculate the channel parameter. The best solution is to get these chip error probabilities, and then we can calculate the channel parameter directly. Whereas, we cannot get the information because the existing commercial ZigBee devices generally implement the decoding of phase shifts by hardware. So there is no way to obtain the phase shifts, not even the chips. The lowest level data that we can get is the symbols obtained after the mapping. Our solution is utilizing the link model to calculate the value of $x$ with statistical partial symbol decoding probability.

Then the following three problems must be solved: how we design the content of probe frames to obtain partial symbol decoding information, how we calculate the channel parameter $x$ to estimate all symbol decoding information, and how to estimate link quality utilizing the information.

To obtain partial symbol decoding information, we need to design the format and content of the probe frames. We set the content of probe frames at the transmitter and the receiver in advance, and the receiver identifies the probe frames with the relevance of the content. If there is a bidirectional link, the sender and the receiver will behave as described before. If there is no bidirectional link, then the following all estimation process will be done by the receiver. We explain that later. In practice, we use a packet-level CTC technology similar to WiZig [4] to achieve the transmission from ZigBee to WiFi.

To obtain a more accurate estimation result, the impact of random errors should be minimized. So we should choose the symbols with the highest probability to be correctly or wrongly decoded as the payload. Then the number of samples will increase. The experiment finds that the probabilities to be wrongly decoded between all symbols are less than 50%, and the probabilities to be correctly decoded are obtained through Figure 5. So we finally select two symbols with the highest correct decoding rate as the payload content. The reason for choosing two symbols is to further reduce the impact of statistical errors and to ensure a small probe frame length. This measure can improve the accuracy of statistical symbol decoding probability.

### *C. Calculate the channel parameter*

The link model reveals the relationship between the symbol decoding probability and the channel parameter $x$. Although theoretically the channel parameter can be obtained by using a single symbol decoding probability, this method might cause a large calculation error due to the statistical error. Consider that the process needs quantity of calculation and is very difficult, commercial ZigBee devices might don't support such calculation. Instead, since the range of $x$ is from 0 to $\pi$, we can take $x$ as 0, $\frac{\pi}{10}$, $\frac{2\pi}{10}$...$\pi$. Then we calculate the corresponding decoding probability of all the symbols, so that we can get a mapping table between the value of $x$ and the symbol decoding probabilities in advance. When we get a set of symbol decoding probabilities, we can compare them with the symbol decoding probabilities in the mapping table, and select the value of $x$ corresponding to the most similar symbol decoding probability as the calculation result.

Here we consider that the set of symbol decoding probabilities only contains two symbols, and the Euclidean distance is used as the metric to judge the similarity. Then we can use the following formula to calculate $x$:

$$\begin{cases} \min\ (\sum_{i=1}^{2}(P_s(S_{a_i} \to S_{a_i}) - P_c(S_{a_i} \to S_{a_i}))^2) \\ \qquad\qquad x \in \{0, \frac{\pi}{10}, \frac{2\pi}{10}...\pi\} \end{cases} \tag{9}$$

Our idea is to turn the problem into an optimization problem, and find the $x$ that will minimize the distance. We use $P_s(S_{a_i} \to S_{a_i})$ to represent the symbol correct decoding probability obtained by statistics and use $P_c(S_{a_i} \to S_{a_i})$ to represent the corresponding probability in the mapping table. $S_{a_i}$ represents the ZigBee symbol $a_i$. In this way, we can solve the second problem.

### *D. Estimate link quality*

For the last question, we design separately for two different configurations: a single transmitter with bidirectional communication and multiple transmitters with unidirectional communication. For the first case, the transmitter and the receiver can communicate with each other. In this case, after receiving the probe frames, the receiver can calculate the channel parameter $x$ and send it back to the transmitter by ACK. Then the transmitter uses the information to obtain each symbol correct decoding probability, and estimates the PRR with the content to be sent. The following formula can be used:

$$PRR_e = \prod_{0 \le i \le 15} P_c(S_{a_i} \to S_{a_i})^{n_i} \tag{10}$$

Where $PRR_e$ represents the estimated PRR by calculating and $n_i$ represents the number of symbol $i$ in the packet to be sent.

Whereas, not all bidirectional communication between heterogeneous devices has been achieved. In this case, the receiver must judge the link quality by itself. What the receiver needs to do is to calculate all the symbol correct decoding probabilities and make a judgment. Our assumption is that the probability of transmitting each symbol next is the same, then we can estimate the link quality by the following formula:

$$\frac{\sum_{i=0}^{15} P_c(S_{a_i} \to S_{a_i})}{16} \tag{11}$$

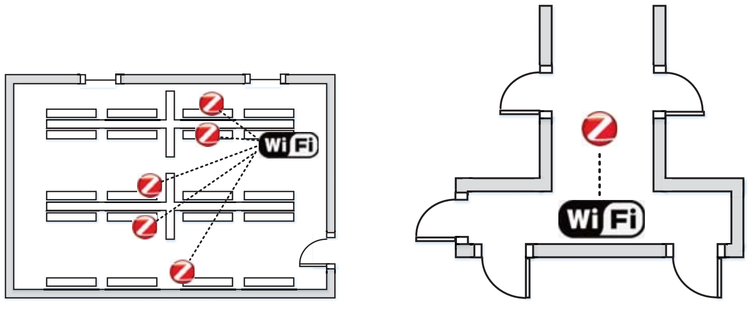
In this way, the receiver can estimate the link quality corresponding to each transmitter and select the best link.

### *E. Practical issues*

**Cross-correlation module.** In order to make the estimation more accurate, we remove the Cyclic Redundancy Check (CRC). Due to the existence of the emulation errors and random distortion, it is difficult to correctly decode all the symbols of a probe frame. If we still use the CRC, the packet reception rate of the probe frames is greatly reduced. It also affects our statistics on the symbol decoding probability. So we remove the CRC module and introduce a cross-correlation module of the payload. When the correlation coefficient is greater than a threshold, the decoded frame is considered as a probe frame. In this way, we can greatly improve the packet reception rate and increase the total number of symbol samples.

**Applicability of C-LQI in co-existing networks.** Physical layer and link layer are independent, so all the link estimation metrics can be used by the link layer, no matter the metrics are estimated by homogeneous physical layer or CTC physical layer. Therefore, C-LQI can be widely used in the co-existing networks.

**Generalizability of C-LQI.** C-LQI can be used to estimate the qualities of different CTC links. All of the CTCs in which the receiver decodes with phase shifts like WEBee, WIDE and BlueBee have the emulation errors and the channel distortion. they can also use C-LQI to estimate their link qualities. Due to their emulation methods are not the same, their emulation errors are different. When they use C-LQI to estimate the link quality, they should calculate their emulation errors first. On the other hand, because C-LQI does not make any modification



(a) Lab

(b) Hallway

Fig. 10: Experiment settings in the lab and the hallway

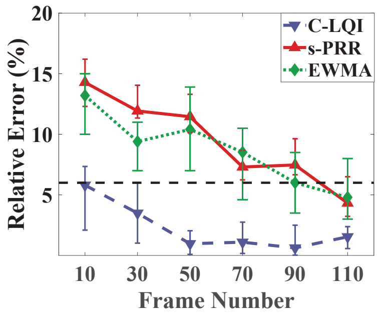
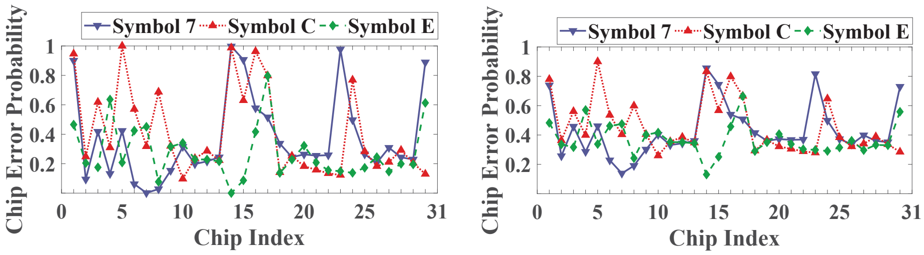


Fig. 11: Overall performance comparison



(a) x=$\pi/3$

(b) x=$2\pi/3$

Fig. 12: Chip error probabilities with different channel parameters

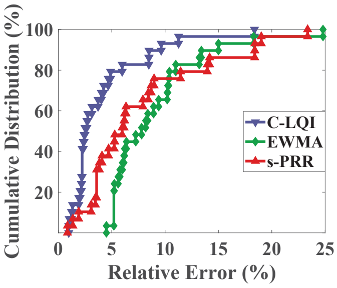


Fig. 13: Relative error distributions of C-LQI, s-PRR and EWMA

to the devices, it can be easily implemented on commodity devices.

## V. Evaluation

This section conducts extensive experiments to evaluate the performance of C-LQI. We compare C-LQI with s-PRR and EWMA. s-PRR utilizes the previous PRR in a short period of time to estimate the current PRR. EWMA utilizes the result of exponentially weighted moving average on PRR. We use USRP N210 devices to conduct out experiments, as it can be used to measure the physical level information, such as phase shifts and chip sequences. Our prototype can also be easily implemented on the commercial devices. We did not implement this part simply because we were unable to implement WEBee on commercial devices.

### *A. Implement*

Our transmitter is a USRP N210 device with 802.11 a/g PHY. Another USRP N210 device with 802.15.4 PHY is used as the receiver. Each packet consists of the header, packet length, and payload specified in the ZigBee standard while the CRC is removed. Since the PRR is no longer determined by CRC but the threshold of correlation, the method of estimating PRR by C-LQI also changes accordingly. The PRR in a certain period of time after the estimation is used as the indicator to evaluate the accuracy and communication cost of these metrics. As shown in Figure 10, we conducted experiments in our lab, hallway, and an empty room. We set the ZigBee at channel 19 and the central frequency of the WiFi at 2440MHz. The weight of the previous result in EWMA is set to 0.2. To ensure statistical validity, we use the average result of 5 experiments. Each experiment sends 100 packets under a wide range of settings including small/large frame number, short/long distance, short/long frame length, weak/strong transmission power, small/large transmission frequency and different environments.

### *B. Chip error probability*

First, we observe the effect of the channel parameter on the correct decoding probabilities of each chip. As shown in Figure 12, in the case of the same channel parameter, the waveform obtained after WEBee emulation and the phase shifts at each location are different because different ZigBee symbols have different emulation errors. Moreover, the chip error probabilities are different. On the other hand, when the channel parameters are different, the influence on the same phase shift are different, and the chip error probability at the same position also changes accordingly.

### *C. Overall performance comparison*

We conducted experiments to compare the performance between our work, s-PRR and EWMA. The result is shown in Figure 11. The payload is 6 bytes and the distance between the sender and the receiver is 1m. When the frame number is 10, the relative error of C-LQI, s-PRR and EWMA is 5.8%, 14.29% and 13.2%, respectively. When the frame number

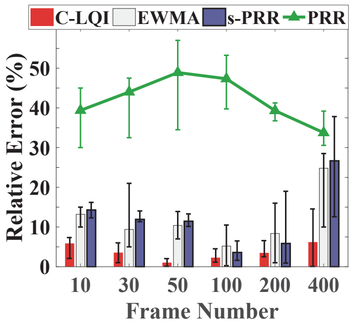


Fig. 14: Relative error and PRR with different frame numbers

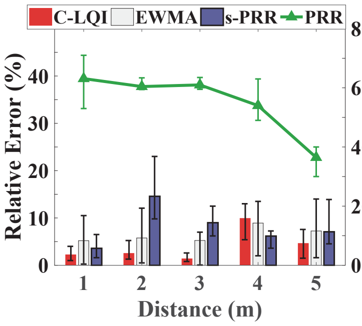


Fig. 15: Relative error and PRR with different distances

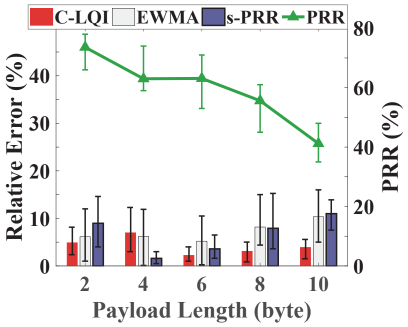


Fig. 16: Relative error and PRR with different frame lengths

increase to 110, the relative error of C-LQI, s-PRR and EWMA decrease to 1.54%, 4.32% and 4.8%, respectively. The relative error of C-LQI is still much lower than that of s-PRR and EWMA, because we get a lot of symbol decoding information from a probe frame, while s-PRR and EWMA only knows whether the packet is received or not.

On the other hand, as shown in Figure 13, there are 97.10% of the results of C-LQI whose relative errors are less than 10%. It is only 75.86% for s-PRR and 65.52% for EWMA. Precisely, the average of the relative estimation error of C-LQI is 4.18% while those of s-PRR and EWMA are 7.78% and 8.92%, respectively. The reason is that C-LQI can sense the change of the channel more clearly by symbol decoding information, so the estimation is more accurate. Meanwhile, the average relative error of EWMA is larger than s-PRR because EWMA considers more previous data and may have more estimation errors.

### *D. C-LQI performance under different settings*

This section observes the performance of C-LQI, s-PRR and EWMA under different settings, including frame number, distance, frame length, transmission power, frame transmission frequency and environment.

*1) Impact of frame number:* We study the impact of frame number on C-LQI, s-PRR and EWMA. We change the frame number from 10 to 400 when the payload is 6 bytes and the distance is 1m. The result is shown in Figure 14. Note that when the frame number is 200 or 400, C-LQI still utilizes the latest 100 packet information to estimate the link quality. When the frame length is 30, the relative errors of C-LQI, s-PRR and EWMA are 3.49%, 11.93% and 9.4%. That means the precision of using s-PRR and EWMA is limited because the numbers of the packet received information is not enough, while the number of symbol decoding information is enough to estimate the link quality accurately. When the frame number is 400, the relative errors of s-PRR and EWMA are more than 10% again, which means that the channel has changed. If the earliest information is still used to estimated link quality, it will introduce more errors. Since C-LQI always utilizes the latest information to estimate, the results are more accurate.

*2) Impact of distance:* We then study the impact of distance on C-LQI, s-PRR and EWMA. We change the distance between the sender and the receiver from 1m to 5m. The frame number is 100 and the payload is 6 bytes. Figure 15 shows the result. As the distance increases, the PRR continues to decrease. The estimation errors of different metrics are relatively stable. The average relative errors of estimation results are 3.42%, 7.12% and 6.48%, respectively.

*3) Impact of frame length:* The impact of frame length on C-LQI, s-PRR and EWMA is shown in Figure 16. We change the frame length from 2 bytes to 10 bytes and other settings remain unchanged. Due to the emulation errors of signals, it has a higher probability of wrong decoding when the payload is longer, resulting in a lower PRR. When the frame is short, the relative error of C-LQI is higher because the symbol decoding information is less. The average relative errors of C-LQI, s-PRR and EWMA are 3.46%, 5.11% and 7.21%, respectively.

*4) Impact of transmission power:* The impact of transmission power on C-LQI, s-PRR and EWMA is similar to the impact of distance. The result is shown in Figure 17. We change the transmission power gain from 0db to 20db. The distance is 1m and the payload length remains 6 bytes. The smaller the transmission power, the lower the PRR. The average relative errors of C-LQI, s-PRR and EWMA are 5.02%, 9.48% and 8.13%, respectively. We believe that when the transmission power is 15db, the channel changes, making the estimation result of s-PRR poor.

*5) Impact of frame transmission frequency:* We then study the impact of frame transmission frequency on C-LQI, s-PRR and EWMA. The result is shown in Figure 18. We change the transmission period from 20ms to 400ms. When the transmission period is 200ms, the relative errors of C-LQI, s-PRR and EMWA are 11.27%, 19.01% and 18.3% respectively. Since the long transmission period will result in a less sensitive estimation, some previous information becomes obsolete. The average relative errors of C-LQI, s-PRR and EWMA are 7.12%, 8.5% and 9.88%, respectively.

*6) Impact of environment:* We finally study the impact of different environments on C-LQI, s-PRR and EWMA. The result is shown in Figure 19. There are several labs around

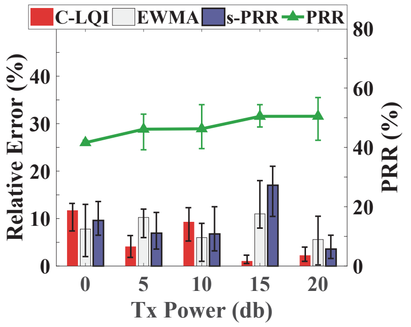


Fig. 17: Relative error and PRR with different transmission powers

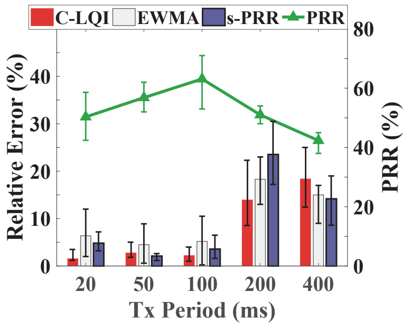


Fig. 18: Relative error and PRR with different frame transmission frequencies

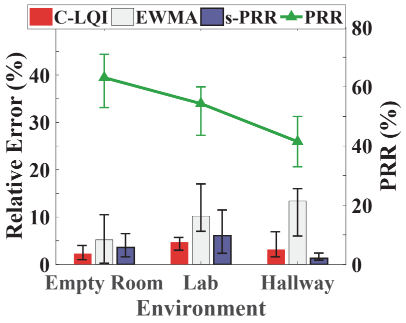


Fig. 19: Relative error and PRR with different environments

the site where we conduct experiments in the hallway, so the interference of the hallway is the highest. The estimation results of C-LQI and s-PRR are better than that of EWMA, because the channel changes quickly and the early data can cause estimation errors.

## VI. Related Work

In recent years, CTC technology has gradually attracted people's attention because it enables direct communication between heterogeneous devices. With the deepening of researches, direct communications between many different technologies have been achieved. The throughput also increases from hundreds of bps of packet-level CTC to several hundred kbps of physical-layer CTC or even higher. The gradual improvement of the underlying technology makes networking between heterogeneous devices possible, but it also needs to solve many problems faced when networking, such as link quality estimation.

As the first work to propose the physical emulation, WEBee successfully emulates the waveform similar to the ZigBee signal. Even if WEBee adopts techniques such as repeated preamble and link coding, it still needs retransmission to ensure the reliability of the link in theory. LongBee [33] and TwinBee [34] have proposed different techniques to improve the performance of WEBee, but the retransmission is still required. The physical-layer CTC has unreliable characteristics. If we want to use physical-layer CTC for networking, link quality estimation is indispensable.

In fact, most of the work of CTC is still concentrated on the implementation of physical-layer CTC technology between different communication technologies. The works related to the CTC upper layer design are limited. ECC [19] mainly uses CTC to change the start time of WiFi transmission to increase the whitespace, so that ZigBee devices can communicate with each other. ECT [23] uses CTC to perform separate transmission of different priority data. Both of those works focus on the application of CTC technology, while our work is to truly analyze the characteristics of CTC and solve the classic problem of link quality estimation in CTC links.

On the other hand, link quality estimation has been well known. By link quality estimation, network routing and network topology can be changed, thereby improving the throughput of the entire network and reducing unnecessary overhead. ETX proposes to use the number of data packets required to complete a bidirectional transmission to estimate link quality. Four-bit proposes combining different levels of information, including decoding information of the physical layer, the packet reception of link layer and the routing of network layer. But they are not directly available for estimating CTC links due to the unreliability of CTC links.

The link quality estimation of WiFi itself is realized by calculating the CSI of the packet header. Whereas, since the ZigBee receiver cannot receive the complete WiFi header, the CSI cannot be used. The LQI indicator used by ZigBee itself for link quality estimation is much too simple. The number of chip errors in the first eight symbols is counted. It cannot be used because the physical layer CTC has emulation errors.

## VII. Conclusion

With the rapid progress of CTC, how to utilize the CTC links and manage them in the networking context has great significance in both research and applications. Our work in this paper presents the first comprehensive study on the CTC link and brings to light the two important impacting factors of CTC link quality, namely the emulation error and the channel distortion. Our proposal includes a novel link metric C-LQI, a joint link model to characterize the CTC link, and a ready-to-use link estimation approach. We implement C-LQI on USRP N210 and demonstrate its advantages over the existing approaches through extensive experiments. In the future, we will design a more generic version of C-LQI, so that it can be seamlessly integrated with different physical-level CTCs. We also plan to port the implementation to different hardware platforms.

## Acknowledgment

This work is supported by the National Key RD Program of China No.2017YFB1003000 and the National Natural Science Foundation of China No. 61772306.

# References

[1] Z. Yin, W. Jiang, S. M. Kim, and T. He, “C-morse: Cross-technology communication with transparent morse coding,” in *IEEE INFOCOM*, 2017.

[2] S. M. Kim and T. He, “Freebee: Cross-technology communication via free side-channel,” in *ACM MobiCom*, 2015.

[3] Z. Chi, Z. Huang, Y. Yao, T. Xie, H. Sun, and T. Zhu, “EMF: embedding multiple flows of information in existing traffic for concurrent communication among heterogeneous iot devices,” in *IEEE INFOCOM*, 2017.

[4] X. Guo, X. Zheng, and Y. He, “Wizig: Cross-technology energy communication over a noisy channel,” in *IEEE INFOCOM*, 2017.

[5] X. Guo, Y. He, X. Zheng, L. Yu, and O. Gnawali, “Zigfi: Harnessing channel state information for cross-technology communication,” in *IEEE INFOCOM*, 2018.

[6] K. Chebrolu and A. Dhekne, “Esense: communication through energy sensing,” in *ACM MobiCom*, 2009.

[7] Z. Chi, Y. Li, H. Sun, Y. Yao, Z. Lu, and T. Zhu, “B2W2: n-way concurrent communication for iot devices,” in *ACM SenSys*, 2016.

[8] W. Jiang, Z. Yin, S. M. Kim, and T. He, “Transparent cross-technology communication over data traffic,” in *IEEE INFOCOM*, 2017.

[9] B. Kellogg, V. Talla, S. Gollakota, and J. R. Smith, “Passive wi-fi: Bringing low power to wi-fi transmissions,” in *USENIX NSDI*, 2016, pp. 151–164.

[10] W. Wang, X. Zheng, Y. He, and X. Guo, “Adacomm: Tracing channel dynamics for reliable cross-technology communication,” in *IEEE SECON*, 2019.

[11] X. Zheng, Y. He, and X. Guo, “Stripcomm: Interference-resilient cross-technology communication in coexisting environments,” in *IEEE INFOCOM*, 2018.

[12] W. Jiang, S. M. Kim, Z. Li, and T. He, “Achieving receiver-side cross-technology communication with cross-decoding,” in *ACM MobiCom*, 2018.

[13] X. Guo, Y. He, X. Zheng, Z. Yu, and Y. Liu, “Lego-fi: Transmitter-transparent CTC with cross-demapping,” in *IEEE INFOCOM*, 2019.

[14] Y. Li, Z. Chi, X. Liu, and T. Zhu, “Chiron: Concurrent high throughput communication for iot devices,” in *ACM MobiSys*, 2018.

[15] Z. Li and T. He, “Webee: Physical-layer cross-technology communication via emulation,” in *ACM MobiCom*, 2017.

[16] X. Guo, Y. He, J. Zhang, and H. Jiang, “WIDE: physical-level CTC via digital emulation,” in *IEEE IPSN*, 2019.

[17] W. Jiang, Z. Yin, R. Liu, Z. Li, S. M. Kim, and T. He, “Bluebee: a 10, 000x faster cross-technology communication via PHY emulation,” in *ACM SenSys*, 2017.

[18] S. M. Kim, S. Wang, and T. He, “cetx: Incorporating spatiotemporal correlation for better wireless networking,” in *ACM SenSys*, 2015.

[19] Z. Yin, Z. Li, S. M. Kim, and T. He, “Explicit channel coordination via cross-technology communication,” in *ACM MobiSys*, 2018.

[20] Z. Yu, C. Jiang, Y. He, X. Zheng, and X. Guo, “Crocs: Cross-technology clock synchronization for wifi and zigbee,” in *ACM EWSN*, 2018.

[21] S. Wang, Z. Yin, Z. Li, and T. He, “Networking support for physical-layer cross-technology communication,” in *IEEE ICNP*, 2018.

[22] Z. Chi, Y. Li, Y. Yao, and T. Zhu, “PMC: parallel multi-protocol communication to heterogeneous iot radios within a single wifi channel,” in *IEEE ICNP*, 2017.

[23] W. Wang, T. Xie, X. Liu, Y. Yao, and T. Zhu, “ECT: exploiting cross-technology transmission for reducing packet delivery delay in iot networks,” in *IEEE INFOCOM*, 2018.

[24] Y. Li, Z. Chi, X. Liu, and T. Zhu, “Passive-zigbee: Enabling zigbee communication in iot networks with 1000x+ less power consumption,” in *ACM SenSys*, 2018.

[25] Z. Chi, Y. Li, Z. Huang, H. Sun, and T. Zhu, “Simultaneous bi-directional communications and data forwarding using a single zigbee data stream,” in *IEEE INFOCOM*, 2019.

[26] W. Wang, X. Liu, Y. Yao, Y. Pan, Z. Chi, and T. Zhu, “CRF: coexistent routing and flooding using wifi packets in heterogeneous iot networks,” in *IEEE INFOCOM*, 2019.

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

[28] M. Zuniga and B. Krishnamachari, “An analysis of unreliability and asymmetry in low-power wireless links,” *ACM TOSN*, vol. 3, no. 2, 2007.

[29] D. Puccinelli and M. Haenggi, “Duchy: Double cost field hybrid link estimation for low-power wireless sensor networks,” in *Hot EmNets*, 2008.

[30] A. Woo, T. Tong, and D. E. Culler, “Taming the underlying challenges of reliable multihop routing in sensor networks,” in *ACM SenSys*, 2003.

[31] D. S. J. D. Couto, D. Aguayo, J. C. Bicket, and R. T. Morris, “A high-throughput path metric for multi-hop wireless routing,” in *ACM MobiCom*, 2003.

[32] R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis, “Four-bit wireless link estimation,” in *ACM HotNets*, 2007.

[33] Z. Li and T. He, “Longbee: Enabling long-range cross-technology communication,” in *IEEE INFOCOM*, 2018.

[34] Y. Chen, Z. Li, and T. He, “Twinbee: Reliable physical-layer cross-technology communication with symbol-level coding,” in *IEEE INFOCOM*, 2018.