Link Quality Estimation of Cross-Technology Communication: The Case with Physical-Level Emulation

JIA ZHANG, XIUZHEN GUO, and HAOTIAN JIANG, Tsinghua University, China
XIAOLONG ZHENG, Beijing University of Posts and Telecommunications, China
YUAN HE, Tsinghua University, China

Research on **cross-technology communication** (**CTC**) has made rapid progress in recent years. While the CTC links are complex and dynamic, 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. Furthermore, the channel distortion can be modeled and observed through the signal strength and the noise strength. We, in this article, 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 ***In-phase and Quadrature*** (**IQ**) domain. We accurately describe the superimposed impact on the received signal. We further design a lightweight link estimation approach including two different methods to estimate C-LQI and in turn the **packet reception rate** (**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 estimation error by 49.8% and 51.5% compared with s-PRR and EWMA, respectively.

CCS Concepts: • **Networks** → **Network protocols;**

Additional Key Words and Phrases: Cross-technology, link quality estimation

ACM Reference format:
Jia Zhang, Xiuzhen Guo, Haotian Jiang, Xiaolong Zheng, and Yuan He. 2021. Link Quality Estimation of Cross-Technology Communication: The Case with Physical-Level Emulation. *ACM Trans. Sen. Netw.* 18, 1, Article 14 (September 2021), 20 pages.
https://doi.org/10.1145/3482527

1 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.

This work is supported in part by National Key R&D Program of China Grant No. 2017YFB1003000 and the R&D Project of Key Core Technology and Generic Technology in Shanxi Province (2020XXX007) and the Smart Xingfu Lindai Project.
Authors' addresses: J. Zhang, X. Guo, H. Jiang, and Y. He (corresponding author), Tsinghua University, 30 Shuangqing Rd., Haidian, Beijing, China; emails: {j-zhang19, guoxz16, jht19}@mails.tsinghua.edu.cn, heyuan@mail.tsinghua.edu.cn; X. Zheng, Beijing University of Posts and Telecommunications, 10 Xitucheng Rd., Haidian, Beijing, China; email: zhengxiaolong@bupt.edu.cn.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.
1550-4859/2021/09-ART14 $15.00
https://doi.org/10.1145/3482527

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 [36], transmission timing [7, 22], transmission power [5, 9, 15, 17, 19, 21], and so on, to carry information, as is called packet-level CTC. The state of the arts in this area is physical-level CTC [16, 18, 24]. 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 [26] that the **packet reception rate** (**PRR**) of WEBee varies between 45% and 55%. It requires up to six retransmissions to get 99% reliable transmission of a packet. WIDE [14] 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 [20] 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 [23], transmission strategy [10, 32, 37, 38], and routing structure [8, 25, 33, 34]. 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 [30], SNR [40], and so on. (2) Metrics derived from physical-level measurements, such as LQI and CSI [29]. (3) Packet-level indicators. For example, PRR [35] measures the rate of successfully received packets. ETX [11, 23] 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 [12].

Based on our observation and study, we find that none of the existing approaches can be applied to estimate the link quality of CTC, especially the emulation-based CTC. The reason is that a CTC link is fundamentally different from 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** (**SERs**). 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 article, 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 an emulation-based 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 the emulation-based CTC and describes their superimposed impact on the received signal in the ***In-phase and Quadrature*** (**IQ**) domain. To estimate C-LQI, we design a lightweight link estimation approach: According to the receiver's ability, the CTC sender either sends normal data frames or periodically sends to the CTC receiver-specific probe frames, in which selected symbols are embedded. According to the reception state returned from the receiver, the sender can estimate the channel parameter and C-LQI of the CTC link. Given the composition of the packet payload, C-LQI can be utilized to estimate the PRR over the CTC link.

The idea of C-LQI has been preliminarily explored in our previous work [39], and we have carried out further extensions and explorations in this version. Compared with the published INFOCOM version, we first conduct an accurate analysis of the joint impact of the two factors

on the channel quality in the IQ domain. Then we modify the design of C-LQI to support the direct estimation of the CTC link quality without any probe frames. We further design experiments to strengthen our motivation. Finally, we conduct a series of experiments to study the direct path of C-LQI performance under different settings and give a detailed analysis of the experimental results.

Our contribution can be summarized as follows:

- To the best of our knowledge, C-LQI is the first metric of emulation-based CTC link quality. The joint link model underneath C-LQI comprehensively characterizes the impacting factors of a CTC link. It clearly explains the phase changes caused by the emulation error and the channel distortion in the IQ domain, and the impact of these phase changes on signal decoding.
- Given the receiver's ability and the calculated mapping table between the channel parameter and the value of C-LQI, we provide two methods to estimate the CTC link quality. Then we address a series of technical challenges, including the mapping table construction, probe frame composition, channel parameterization, and PRR calculation. We further address several issues related to the applicability of our approach in practice.
- We implement WEBee-based C-LQI on USRP N210 and conduct extensive experiments under various settings. The results demonstrate that the two paths of C-LQI are both highly accurate and the direct path of C-LQI performs better. When using the direct path of C-LQI to estimate the PRR of a CTC link, it achieves 4.47% relative error in average, which is 49.8% and 51.5% lower than that of two representative link estimation approaches.

The rest of the article is organized as follows. Section 2 discusses the important observation of the CTC link, which motivates our research in this article. Section 3 introduces the definition of C-LQI and presents the joint link model. Section 4 elaborates on the link estimation approach. The implementation and evaluation results are presented in Section 5. Section 6 discusses the related work. We conclude our work in Section 7.

2 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 shifts of the receiver. We illustrate that emulating different target signals produces different emulation errors. We further find that when the emulated signals pass through the wireless channel, the actual distribution of decoding errors is not the same as that of the emulation errors. At the end of this section, we theoretically explain this observation.

2.1 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 an example, WEBee [26] successfully emulates the waveform of a ZigBee signal using a WiFi signal. 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 emulate ZigBee signals perfectly. Figure 2 shows the target and emulated signal waveform of ZigBee symbol 7 in WEBee. 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.

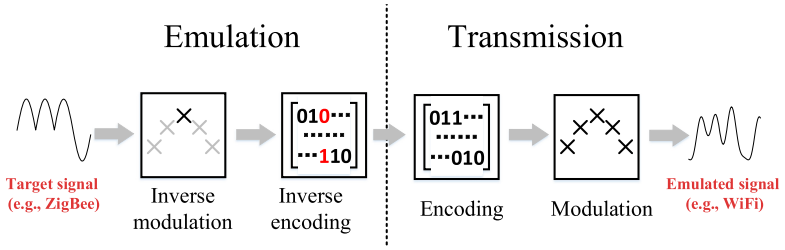


Fig. 1. The typical process of physical emulation.

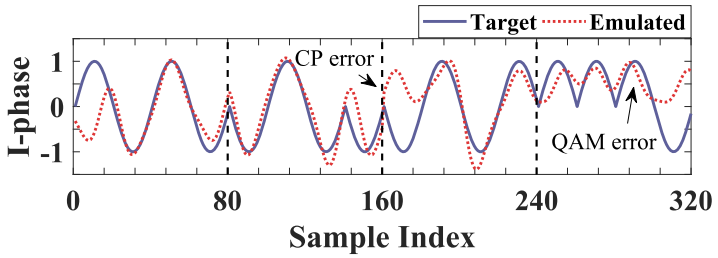


Fig. 2. The target signal waveform and the emulated signal waveform.

When the emulated signals arrive at the ZigBee receiver, the receiver first down-converts the received signals and digitalizes them into IQ samples by **Analog-to-Digital Converter** (**ADC**). Then the phase shift between the consecutive IQ samples is used to decode the chips. If a phase shift is greater than zero, the receiver outputs the chip "1," otherwise it outputs the chip "0." Every 32 chips will be mapped into four bits (a ZigBee symbol) with the **Direct Sequence Spread Spectrum** (**DSSS**) mapping table.

As the ZigBee receiver decodes the chip sequence according to the phase shifts between the sampling points, there exist chip errors in the decoded results because of the emulation error. 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 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 be decoded correctly. As the decoding threshold is usually 12 in the ZigBee receiver, all the emulated signals can be received correctly if there is no channel distortion. On the other hand, LQI [4] is measured based on the number of correct chips in the first eight symbols of the received packet. So it is obvious that the value of LQI will be relatively low even if the symbols are all received successfully, as the emulated signal itself brings certain chip errors.

2.2 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 signals is not the case. Our previous work verified that the SER distribution and the chip error distribution of different symbols are not consistent. The result is shown in Figure 4. We can also see that the emulated signal cannot be decoded completely correctly with the channel distortion. For example, the SER of symbol 3 is the lowest while 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 on the signal waveform. Because the ZigBee receiver decodes the chip by judging whether the phase shift is greater than zero, the

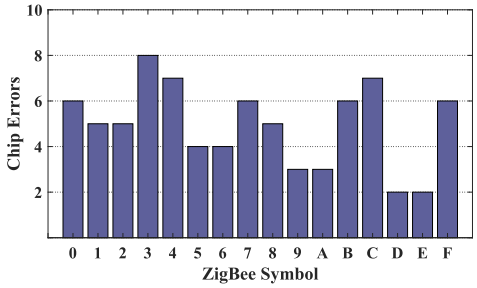


Fig. 3. The chip errors caused by emulation without the channel distortion.

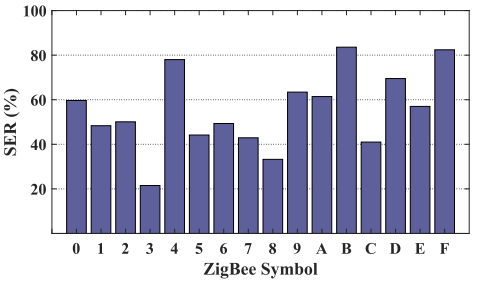


Fig. 4. The SER of each ZigBee symbol with the channel distortion is not the same. The SER of each ZigBee symbol without the channel distortion is zero as the chip errors are all less than 12.

channel distortion affects the phase of each sampling point, which in turn affects the decoding result jointly with the emulation error.

We further find that a larger absolute value of the phase shift is more robust against the random distortion in the wireless channel. When the absolute values of the phase shift between two emulated signals are different, through the same channel distortion, the emulated signal with the larger absolute value is more likely to maintain the original decoding result than the other, that is, the phase shift is still greater than or less than 0.

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.

3 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 the 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. At the end of the section, according to the variation of phase shifts, we calculate C-LQI.

3.1 Joint Link Model

Figure 5 shows our further analysis of the CTC link. The emulation process that introduces the emulation error is the *logical link*, and the wireless channel that introduces the channel distortion 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 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:

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

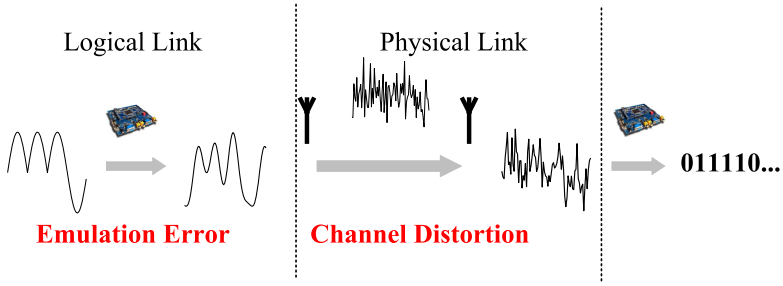


Fig. 5. The link model.

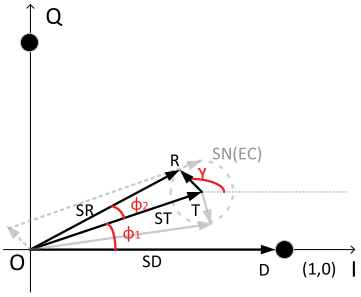
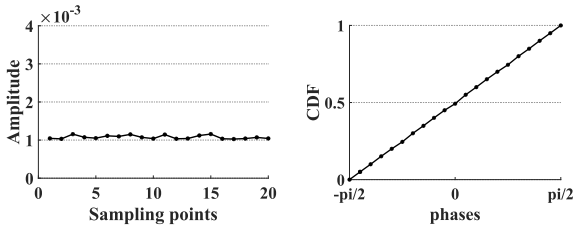


Fig. 6. The effect of the emulation errors and the random distortion on the phases.



(a) The amplitude of the noise (b) The distribution of the noise phase

Fig. 7. (a) The amplitude of the noise. Each sampling point represents the average value within a short period of time. (b) The distribution of the noise phase, which is basically uniformly distributed.

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

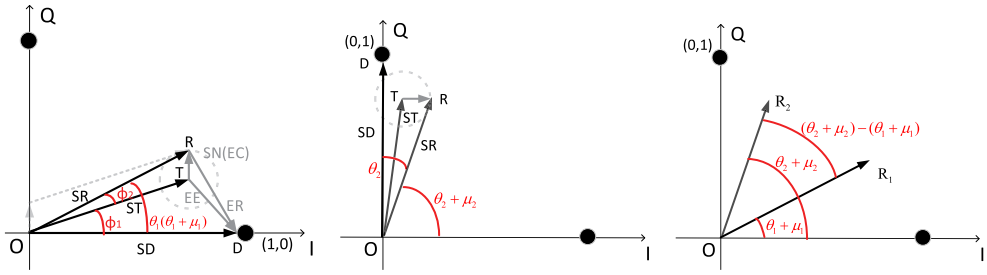
3.2 Variation of Phase Shifts

Since the ZigBee receiver decodes the waveform according to the phase shifts between the adjacent sampling points, we should first consider the variation of phase shifts which determines the variation of the decoding result. As we mentioned before, the CTC link quality is affected by both the emulation errors and the random distortion. Both of them affect the phase shifts, which reflect the link quality.

As we all know, the errors caused by the physical layer emulation will cause a fixed deflection of the sampling points' phases. In addition, the influence of the channel on the sampling points in the waveform is random. It is natural to think that the variation of phases is the algebraic sum of these two, but it is not, in fact.

Figure 6 shows the effect of the emulation errors and the random distortion on the phases. Since ZigBee uses OQPSK modulation, the corresponding constellation points on the I-Q map are $(\pm\frac{\sqrt{2}}{2}, \pm\frac{\sqrt{2}}{2})$. For the sake of clarity, we rotate the coordinate system so that the constellation points fall on the coordinate axis.

Figure 6 shows an example of phase change at a single sampling point. The original constellation point is D; due to the emulation error, the constellation point changes to T and the new signal vector $\vec{ST}$ has the phase change ϕ_1 compared to the original signal vector $\vec{SD}$. We consider a fixed amplitude noise with uniformly random phases at this moment. As shown in Figure 7, the



(a) The first sampling point (b) The second sampling point (c) The phase shift

Fig. 8. (a) The phase change is θ_1 and the actual received phase is $\theta_1 + \mu_1$ on the first sampling point. (b) The phase change is θ_2 and the actual received phase is $\theta_2 + \mu_2$ on the second sampling point. (c) The phase shift between two sampling points is $(\theta_2 + \mu_2)$–$(\theta_1 + \mu_1)$.

amplitude of the noise remains stable and the noise phase is basically uniformly distributed. So the noise vector can be regarded on the circle centered on T as shown in the figure. Then the received signal vector $\vec{SR}$ can be obtained.

According to the sine theorem of triangles, the angle ϕ_2 between the received vector $\vec{SR}$ and the transmitting vector $\vec{ST}$ can be obtained by the following formula:

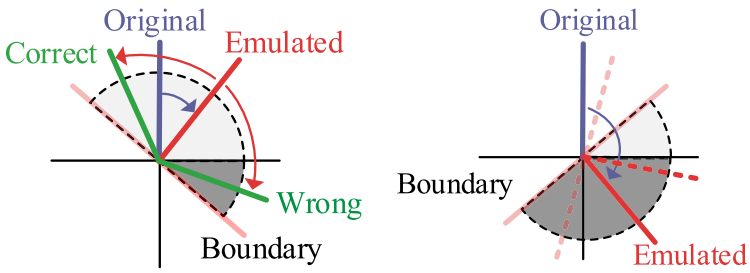
$$\frac{SN}{\sin(\phi_2)} = \frac{SR}{\sin(\pi - \gamma + \phi_1)}, \tag{2}$$

where γ is the phase of the noise signal, varying from 0 to 2π. SN and SR are the noise and received signal strength, respectively. We can further obtain the probability cumulative distribution function of the angle ϕ_2:

$$\begin{aligned} H(y) = P(\phi_2 \le y) &= \frac{x_2 - x_1}{2\pi} \\ &= \begin{cases} \frac{\pi + 2\arcsin(x * \sin(y))}{2\pi} & -\arcsin\frac{1}{x} \le y \le \arcsin\frac{1}{x} \\ 0 & y < -\arcsin\frac{1}{x} \\ 1 & y > \arcsin\frac{1}{x} \end{cases}, \end{aligned} \tag{3}$$

where $\pi - \gamma + \phi_1$ are x_1 and x_2 when ϕ_2 is y in Equation (2). Note that ϕ_1 is constant and γ varies from 0 to 2π, so each ϕ_2 corresponds to two different values of $\pi - \gamma + \phi_1$ when ϕ_2 is in $[-\arcsin\frac{1}{x}, \arcsin\frac{1}{x}]$. The probability that ϕ_2 is less than a certain y can be calculated by $\frac{x_2 - x_1}{2\pi}$. On the other hand, as shown in Figure 6, since the point R is a point on the circle, ϕ_2, which is the angle between the vector $\vec{OR}$ and the vector $\vec{OT}$, can only change within a certain interval $[-\arcsin\frac{1}{x}, \arcsin\frac{1}{x}]$. The value of $P(\phi_2 \le y)$ will be zero if the value of y is less than $-\arcsin\frac{1}{x}$ and the value of $P(\phi_2 \le y)$ will be 1 if the value of y is greater than $\arcsin\frac{1}{x}$. x is the ratio of SR to SN. Since ϕ_1 is fixed, the phase change between the original signal and the received signal can be obtained by $\theta = \phi_1 + \phi_2$.

Figure 8(a) shows the error model. The received signal error $\vec{ER}$ is composed of the emulation signal error $\vec{EE}$ and the channel error $\vec{EC}$. Now we consider the effect of these errors on the phase shifts. We assume that the original phases of two consecutive sampling points are μ_1 and μ_2, μ_1 is 0 in Figure 8(a) and μ_2 is $\frac{\pi}{2}$ in Figure 8(b). The corresponding phase changes are θ_1 and θ_2. In this case, the actually received phases are $(\theta_1 + \mu_1)$ and $(\theta_2 + \mu_2)$. Then the phase shift between these two sampling points in the received signal can be calculated by $(\theta_2 + \mu_2) - (\theta_1 + \mu_1)$ as shown in



(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. 9. Phase shift changes within a certain range due to the channel.

Figure 8(c), and its cumulative distribution function is

$$\begin{aligned} F(y) &= P((\theta_2 + \mu_2) - (\theta_1 + \mu_1) \leq y) \\ &= P((\phi_{21} + \phi_{22} + \mu_2) - (\phi_{11} + \phi_{12} + \mu_1) \leq y) \\ &= P((\phi_{22} - \phi_{12}) + (\phi_{21} + \mu_2) - (\phi_{11} + \mu_1) \leq y). \end{aligned} \tag{4}$$

Since ϕ_{i2} $(i = 1, 2)$ follows Equation (3) and $(\phi_{21} + \mu_2) - (\phi_{11} + \mu_1)$ is actually the phase shift in the emulation signal, which can be obtained in advance, we can obtain the cumulative distribution function of each phase shift as long as the channel parameter x is known.

According to Figure 9(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 9(b) shows, it 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.

The above analysis explains why the actual chip error distributions differ from the emulation results. 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.

3.3 Error Calculation

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

$$C_i = \begin{cases} F(0) & \textit{when standard chip is } 0 \\ 1 - F(0) & \textit{when standard chip is } 1. \end{cases} \tag{5}$$

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 the chip is $P((\theta_2+\mu_2)-(\theta_1+\mu_1) \leq 0)$, which

is $F(0)$. When the standard chip is "1," the phase shift should be larger than 0°. So the probability should be $P((\theta_2 + \mu_2) - (\theta_1 + \mu_1) > 0)$, which is $1 - F(0)$.

In this way, we can get a set of probabilities $C_1, C_2 \ldots C_{30}$ for each symbol corresponding to a certain x, representing the correct probability of the i-th decoded chip, respectively. Notice that different receivers may implement different decoding methods. Our design is based on the soft-decision mechanism, which is applied to many popular ZigBee transceivers, such as CC2420. In this case, the ZigBee receiver uses only 30 out of 32 chips to decode every symbol. The first and 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:

$$P(S_a \rightarrow S_b) = \sum_{n=0}^{30} P(S_a \rightarrow S_b|n) \times P(n, S_a). \tag{6}$$

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, 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:

$$P(n, S_a) = \sum (1 - C_{i_1})(1 - C_{i_2}) \ldots (1 - C_{i_n}) C_{i_{n+1}} \ldots C_{i_{30}}, \tag{7}$$

where $\{i_1, i_2 \ldots i_n\}$ is any combination of n elements taken from $\{i_1, i_2 \ldots i_{30}\}$. 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 case, $(1 - C_{i_1})(1 - C_{i_2}) \ldots (1 - C_{i_n}) C_{i_{n+1}} \ldots C_{i_{30}}$ represents that there are n wrong chips and $(30 - n)$ correct chips. In order to get the probability that any n chips are wrong and the other chips are correct, we need to traverse every combination as described in Equation (7).

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 values of C-LQI.

4 DESIGN

This section first introduces the workflow of C-LQI. According to whether the ZigBee receiver can directly obtain the ratio of the received signal strength to the noise strength, the workflow is divided into two parts, namely, directly obtaining the C-LQI through calculation and indirectly estimating the C-LQI using probe frames. The basis of the entire workflow is a mapping table between the channel parameters and the values of C-LQI. Without abusing notations, we also use C-LQI to denote the link estimation approach proposed in this article. Then we consider some practical issues, including the cross-correlation module, the link burstiness, the estimation of bidirectional CTC links, the applicability of C-LQI in co-existing networks, and the generalizability of C-LQI.

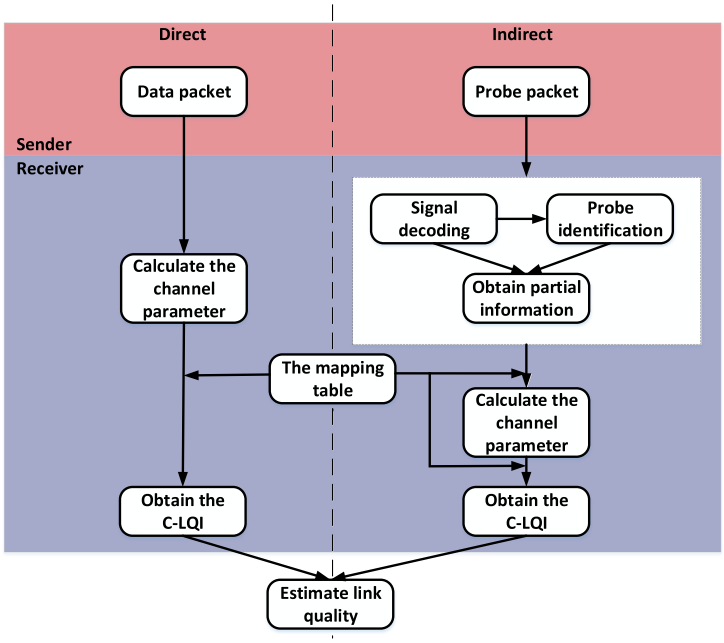


Fig. 10. The workflow of C-LQI.

4.1 The Workflow of C-LQI

The entire workflow of C-LQI is shown in Figure 10. When the receiver can directly obtain the ratio of the received signal strength to the noise strength, the transmitter does not need to send additional probe frames. Having the mapping table between the channel parameter and the C-LQI, the ZigBee receiver can directly obtain the C-LQI that can be used to estimate the CTC link quality.

If the channel parameter cannot be obtained directly, the sender can send some probe frames to assist the receiver to estimate the link quality. 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 mapping table. At the end of this section, the information can be used to estimate link quality.

4.2 The Mapping Table

As we mentioned before, given the channel parameter x, namely, the ratio of the received signal strength to the noise strength, the symbol decoding probability can be obtained, including all the values of C-LQI. However, the calculation of the symbol decoding probability from the channel parameter is very complicated and almost irreversible. Considering that the commercial ZigBee devices might not support such calculation, it's more feasible to obtain the values of C-LQI corresponding to different parameters in advance. Then we can obtain the mapping table between the values of the channel parameter x and the symbol decoding probabilities. This mapping table will be used in the calculation of C-LQI later.

4.3 The Direct Path

If the receiver can directly obtain the channel parameter, which is supported by some commercial devices such as TelosB, the receiver can obtain the channel parameter from the data packets. As shown in Figure 11(a), since the channel parameter can be obtained directly, the sender just sends

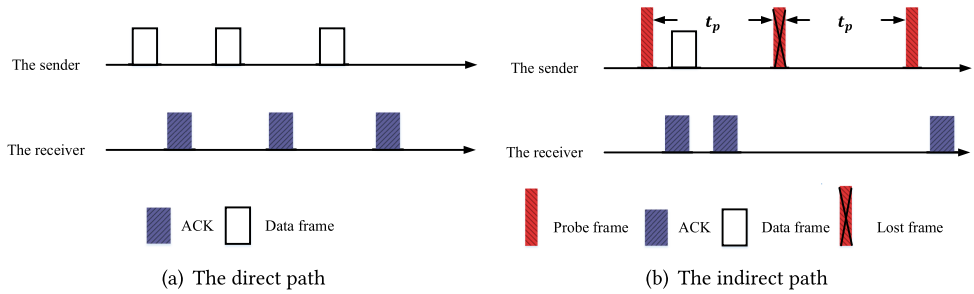


Fig. 11. The protocol design of C-LQI.

normal data frames and the receiver sends the ACK back once it receives the data frames. With the mapping table, the values of C-LQI corresponding to the channel parameter can be obtained. The channel parameters in the mapping table are represented as $\{I_1, I_2 \ldots I_l, I_r \ldots I_n\}$, the values of C-LQI are $\{P_1, P_2 \ldots P_l, P_r \ldots P_n\}$, where n represents the number of channel parameters in the mapping table, and P_i is the value of C-LQI corresponding to the channel parameter I_i. Specifically, the actually obtained channel parameter I_c is an arbitrary value and is probably not in the mapping table. Assuming that the actual channel parameter is in the interval $[I_l, I_r]$, the values of C-LQI corresponding to the end points of the interval are P_l and P_r, respectively. We calculate the weighted average of these two values to get the calculation result; the weight of each value is the interval length from the corresponding endpoint to the actual channel parameter:

$$P_c = \frac{L_{lc} * P_l + L_{cr} * P_r}{L_{lr}}, \tag{8}$$

where L_{lc}, L_{cr}, and L_{lr} are the interval lengths from the actual channel parameter to the left endpoint, the actual channel parameter to the right endpoint, and the left endpoint to the right endpoint, respectively. Taking L_{lc} as an example, it can be calculated by $L_{lc} = I_c - I_r$. P_c represents the calculated value of C-LQI.

After obtaining the C-LQI, the link quality can be estimated, which we will introduce later.

4.4 The Indirect Path

When the ratio of the received signal strength to the noise strength cannot be obtained directly, The indirect path should be used to estimate the link quality, including the collection of partial symbol decoding information, the calculation of the channel parameter, and the estimation of the link quality.

Obtain partial information. In order to obtain the partial symbol decoding information, we first need to design the practical transfer protocol to send some probe frames. As shown in Figure 11(b), 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 sender starts to send normal data frames until the next probe frame should be sent. If the receiver doesn't receive the probe, it doesn't send the ACK back. The receiver can know whether a probe frame is lost by the sequence number.

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 [17] 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. 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 ensure a small probe frame length. This measure can improve the accuracy of the statistical symbol decoding probability.

Calculate the channel parameter. The link model reveals the relationship between the symbol decoding probability and the channel parameter x. The mapping table between the channel parameter and the symbol decoding probabilities has been obtained in advance. When we get a set of symbol decoding probabilities, we can compare them with the symbol decoding probabilities in the mapping table. Then we can 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:

$$\min\left(\sum_{i=1}^{2}(P_s(S_{a_i}\rightarrow S_{a_i})-P_c(S_{a_i}\rightarrow S_{a_i}))^2\right). \tag{9}$$

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

4.5 Estimate Link Quality

The last step of the workflow, whether it is the direct path or the indirect path, is to estimate the link quality. 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's correct decoding probability, and estimates the PRR with the content to be sent. The following formula can be used:

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

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. The receiver needs to calculate all the symbol correct decoding probabilities and make a judgment. We assume that the probability of transmitting each symbol next is the same; then we can estimate the link quality by

the following formula:

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

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

4.6 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 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.

The link burstiness. In actual links, the link quality often remains stable within a certain period of time. This means that we don't need to estimate the link quality all the time before the link quality changes significantly. Instead, only when the channel parameters have changed, we will re-estimate the link quality. Specifically, we periodically calculate their channel parameters for the packets within a period of time. If there are only a few packets whose channel parameters change significantly compared with the previous value, we consider the entire link to be stable at this time. Conversely, the link quality needs to be estimated again. The threshold for determining whether to re-estimate can be obtained through empirical experiments.

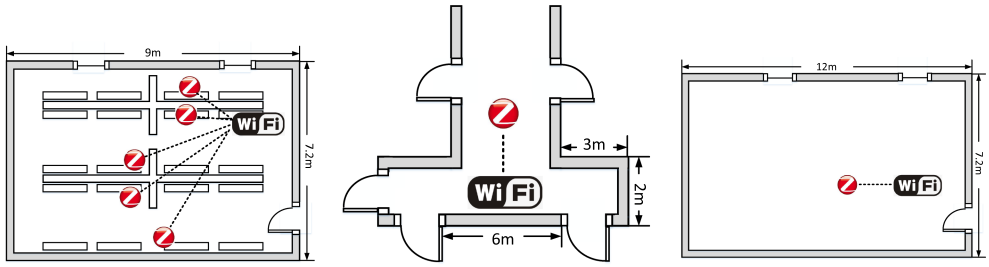
The estimation of bidirectional CTC links. The estimation of bidirectional CTC links is difficult because the links between heterogeneous devices are asymmetric. The link quality is related to the specific implementation of the CTC. If there is a packet-level CTC in the bidirectional CTC, there is no way to use the metric designed for physical-level CTC. In this condition, we can only use higher-level metrics such as PRR or ETX. Even if they are both physical-level CTCs, the corresponding estimation metrics could be different.

Applicability of C-LQI in co-existing networks. The physical layer and the 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 layers or CTC physical layers. 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 emulation errors and channel distortion. They can also use C-LQI to estimate their link qualities. Because 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 to the devices, it can be easily implemented on commodity devices.

5 EVALUATION

This section conducts extensive experiments to evaluate the performance of C-LQI. We compare C-LQI with s-PRR and EWMA. s-PRR [2] utilizes the previous PRR in a short period of time to estimate the current PRR. Using the name s-PRR is to prevent confusion with PRR, which is the metric we need to measure. EWMA [35] utilizes the result of the exponentially weighted moving average on PRR. We use USRP N210 devices to conduct our experiments, as they can be used to measure the physical level information, such as phase shifts and chip sequences. Our prototype is



(a) Lab (b) Hallway (c) Empty room

Fig. 12. Experiment settings in the lab, the hallway, and the empty room.

suitable for the commercial devices whose decoding mechanism is based on phase information. For example, mainstream commercial ZigBee devices CC2530 provided by TI [31], MC1321 provided by FREESCALE [13], AT86RF230 provided by ATMEL [1], and nRF24E1/nRF9E5 provided by Nordic [28] use the phase information to decode ZigBee signals.

We try to implement our WEBee-based design on commercial devices. Unfortunately, our experience indicated that there might be some differences between the GNUradio-based 802.15.4 physical layer implementation in USRP [3] and that on commercial devices. For example, many parameters in the Clock Recovery MM model will affect the sampling results in the GNUradio-based 802.15.4 physical layer implementation. The result further affects the signal decoding results. Due to these factors, so far we didn't have a successful implementation of WEBee on commercial devices. Before these parameters are obtained, it may be uneasy to port C-LQI on commercial devices. However, as our design doesn't require modification to the hardware or the physical layer standard, we believe that C-LQI has the potential to be implemented on other platforms.

5.1 Implementation

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 determined by the threshold of correlation instead of CRC, 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 these metrics' accuracy and communication cost. As shown in Figure 12, we conducted experiments in our lab, a hallway, and an empty room. The interference in the empty room is the least because there are only a few WiFi devices and Bluetooth devices. The interference in the hallway is the strongest because we specifically chose to place the experiment site next to multiple labs. We set the ZigBee at channel 19 and the central frequency of the WiFi at 2440 MHz. The weight of the previous result in EWMA is set to 0.2. To ensure statistical validity, we use the average result of five 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.

5.2 Comparison with Existing Metrics

We first explain that the existing metrics, such as LQI or RSSI, cannot fully characterize the CTC link quality's dynamic change. We don't quantitatively compare these metrics with our method because they cannot be easily converted to each other. In order to show the limitations of these existing metrics, we designed a qualitative experiment where the sender periodically sends a group

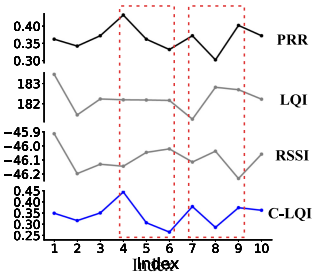


Fig. 13. Comparison with LQI and RSSI.

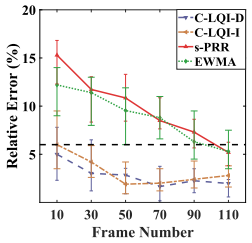


Fig. 14. Overall performance comparison.

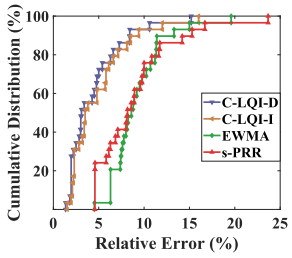


Fig. 15. Relative error distributions of C-LQI, s-PRR, and EWMA.

of 100 packets and transmits a total of 10 groups. According to the decoding result of the receiver, we can calculate the corresponding PRR, the average value of LQI, the average value of RSSI, and the estimation result of C-LQI. We use the value of PRR to represent the current link quality. Note that the value of LQI in each packet is calculated based on the number of the correct chips in the first eight symbols of the received packet.

The result is shown in Figure 13, where each point represents a calculated value using 100 packets. We can find that the change of LQI and RSSI over time follows a similar pattern, but there is a significant difference between their trends and that of PRR. It means that LQI and RSSI are not sensitive to the change of the CTC link quality, and sometimes may give the wrong estimation results. The reason is that neither LQI nor RSSI considers the joint impact of the emulation error and the channel distortion on signal decoding, which is a unique feature in CTC links. On the contrary, our C-LQI considers the joint impact of these two factors. It can accurately estimate the link quality. The experimental result shows that the change of C-LQI and PRR over time follows a similar pattern. We change the number of packets sent in each group to 50 and 200, and the changing pattern of the experimental results is similar to the case when the number of packets sent is 100.

5.3 Overall Performance Comparison

We conduct experiments to compare the performance between our work, s-PRR, and EWMA. The result is shown in Figure 14, where "C-LQI-D" and "C-LQI-I" represent the direct path method and the indirect path method of C-LQI. The payload is 6 bytes and the distance between the sender and the receiver is 1 m. When the frame number is 10, the relative error of C-LQI-D, C-LQI-I, s-PRR, and EWMA is 5.02%, 6.1%, 15.22%, and 12.3%, respectively. When the frame number increases to 110, the relative error of C-LQI-D, C-LQI-I, s-PRR, and EWMA decreases to 1.97%, 2.8%, 5.19%, and 5.3%, respectively. The relative error of C-LQI is still much lower than that of s-PRR and EWMA because C-LQI gets the channel parameter from the packets, while s-PRR and EWMA only know whether the packet is received or not. On the other hand, C-LQI only requires 10 data packets to achieve a similar estimation accuracy compared with that obtained by s-PRR and EWMA using 100 data packets. This result shows that our design has much higher efficiency in CTC link quality estimation.

On the other hand, as shown in Figure 15, there are 93.10% of the results of C-LQI, whose relative errors are less than 10%, whether it is C-LQI-D or C-LQI-I. It is only 75.86% for s-PRR and 69.00% for EWMA. Specifically, the average relative estimation error of C-LQI-D is 4.47% and the average relative estimation error of C-LQI-I is 4.94%, while those of s-PRR and EWMA are 8.91% and 9.21%, respectively. The reason is that C-LQI can sense the change of the channel more clearly

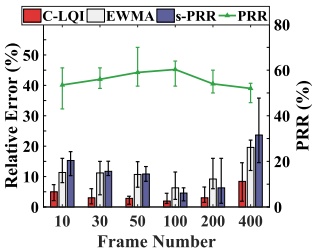


Fig. 16. Relative error and PRR with different frame numbers.

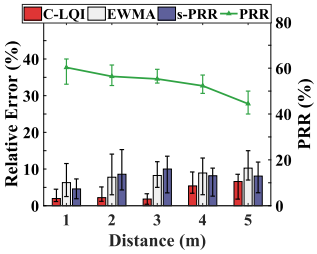


Fig. 17. Relative error and PRR with different distances.

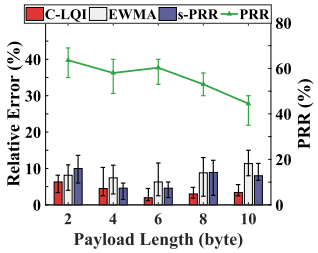


Fig. 18. Relative error and PRR with different frame lengths.

by the channel parameter, so the estimation is more accurate. Furthermore, the result of C-LQI-D is better than the result of C-LQI-I because the former can directly obtain the channel parameter calculated from the physical information, while the latter must use the symbol decoding information to calculate indirectly. Meanwhile, the average relative error of EWMA is larger than s-PRR because EWMA considers older data and may have more estimation errors. Since the result of the direct path method is better, we will use this method by default in the following experiments.

5.4 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.

5.4.1 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 1 m. The result is shown in Figure 16. Note that when the frame number is 200 or 400, C-LQI still utilizes the latest 100 packet information to estimate the link quality. As the frame number increases, more effective information is collected by C-LQI and the emulation error gradually decreases. When the frame number is 400, C-LQI incurs a large estimation error, which may be caused by a change of the channel. When the frame length is 30, the relative errors of C-LQI, s-PRR, and EWMA are 3.02%, 11.74%, and 11.2%. 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 channel parameter is enough to accurately estimate the link quality. 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 estimate link quality, it will introduce more errors.

5.4.2 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 1 m to 5 m. The frame number is 100 and the payload is 6 bytes. Figure 17 shows the result. As the distance continues to increase, the channel distortion increases and the PRR continues to decrease. The reason is that C-LQI utilizes the information from actually received packets to estimate the entire link quality. When the number of the received packets decreases, the effective information collected by C-LQI decreases and the emulation error gradually increases. The average relative errors of estimation results are 3.60%, 7.88%, and 8.30%, respectively.

5.4.3 Impact of Frame Length. The impact of frame length on C-LQI, s-PRR, and EWMA is shown in Figure 18. 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

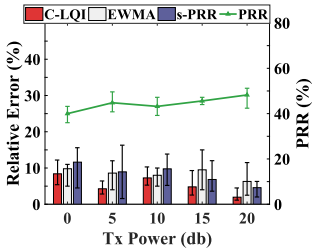


Fig. 19. Relative error and PRR with different transmission powers.

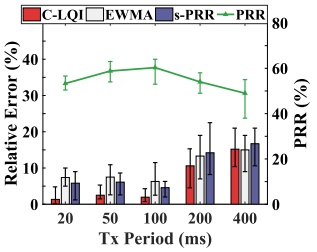


Fig. 20. Relative error and PRR with different frame transmission frequencies.

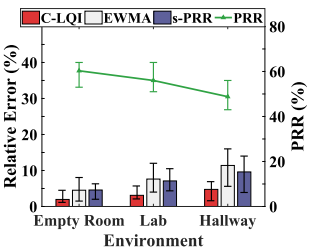


Fig. 21. Relative error and PRR with different environments.

when the payload is longer, resulting in a lower PRR. On the other hand, as the frame length increases, the PRR is lower but the payload is longer. The overall effective information collected by C-LQI increases and the emulation error gradually decreases. The average relative errors of C-LQI, s-PRR, and EWMA are 3.83%, 7.21%, and 8.40%, respectively.

5.4.4 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 19. We change the transmission power gain from 0 db to 20 db. The distance is 1 m 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 5.34%, 8.35%, and 8.44%, respectively.

5.4.5 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 20. We change the transmission period from 20 ms to 400 ms. When the transmission period is 200 ms, the relative errors of C-LQI, s-PRR, and EMWA are 10.60%, 14.2%, and 13.3%, 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 6.32%, 9.48%, and 9.90%, respectively.

5.4.6 Impact of Environment. We finally study the impact of different environments on C-LQI, s-PRR, and EWMA. In our experimental settings, different environments correspond to different levels of interference. The result is shown in Figure 21. There are several labs around the site where we conduct experiments in the hallway, so the interference of the hallway is the highest and the PRR in the hallway is the lowest. As shown in Figure 21, when the interference is strong, the estimation results of C-LQI and s-PRR are better than that of EWMA because the interfered channel changes quickly and the previous data can cause estimation errors. When the interference is low, such as in the empty room, the PRR is relatively high and the effective information collected by C-LQI increases. So the relative error of the estimation result is low. This experiment also confirms that C-LQI can accurately estimate the channel quality under different levels of interference, with a maximum relative error of about 5%.

6 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 [27] and TwinBee [6] 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 [37] 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 [34] 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 [11] proposes to use the number of data packets required to complete a bidirectional transmission to estimate link quality. Four-bit [12] proposes combining different levels of information, including decoding information of the physical layer, the packet reception of the link layer, and the routing of the 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 [29] 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.

Our previous work [39] proposed a simplified joint model for CTC links, which simultaneously takes into account the emulation error and the channel distortion but simplifies the actual joint impact of the two factors. We also designed a method to estimate the CTC link quality by periodically sending probe frames from the transmitter. Compared with the published INFOCOM version, we first conduct an accurate analysis of the joint impact of the two factors on the channel quality in Section 3, which clearly explains the phase changes caused by the emulation error and the channel distortion in the IQ domain, and the impact of these phase changes on signal decoding. We further modify the design of C-LQI to support the direct estimation of the CTC link quality without any probe frames in Section 4. We add experiments to strengthen our motivation in Section 5.2. We evaluate and analyze the overall performance of the two kinds of C-LQI in Section 5.3. We further conduct a series of experiments to study the direct path of C-LQI performance under different settings and give the detailed analysis of the experimental results in Section 5.4.

7 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 article presents the first comprehensive study on the emulation-based CTC link and brings to light the two important impacting factors of the emulation-based 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 accurately characterize the emulation-based CTC link, and a ready-to-use link estimation approach including the two different paths, namely, the direct path and the indirect path. We

implement C-LQI on USRP N210 and demonstrate its advantages over the existing approaches through extensive experiments.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their valuable comments and helpful suggestions.

REFERENCES

[1] ATMEL. 2020. AT86RF230 Provided by ATMEL. https://www.alldatasheet.com/datasheet-pdf/pdf/313502/ATMEL/AT86RF230.html.

[2] Nouha Baccour, Anis Koubâa, Luca Mottola, Marco Antonio Zúñiga, Habib Youssef, Carlo Alberto Boano, and Mário Alves. 2012. Radio link quality estimation in wireless sensor networks: A survey. *ACM Transactions on Sensor Networks (TOSN)* 8, 4 (2012), 1–33.

[3] Bastian Bloessl. 2020. An IEEE802.15.4 OQPSK Transceiver for GNU Radio, Based on Thomas Schmid's Implementation. https://github.com/bastibl/gr-ieee802-15-4.

[4] Carlo Alberto Boano, Thiemo Voigt, Adam Dunkels, Fredrik Österlind, Nicolas Tsiftes, Luca Mottola, and Pablo Suarez. 2009. Poster abstract: Exploiting the LQI variance for rapid channel quality assessment. In *Proceedings of ACM/IEEE IPSN*.

[5] Kameswari Chebrolu and Ashutosh Dhekne. 2009. Esense: Communication through energy sensing. In *Proceedings of ACM MobiCom*.

[6] Yongrui Chen, Zhijun Li, and Tian He. 2018. TwinBee: Reliable physical-layer cross-technology communication with symbol-level coding. In *Proceedings of IEEE INFOCOM*.

[7] Zicheng Chi, Zhichuan Huang, Yao Yao, Tiantian Xie, Hongyu Sun, and Ting Zhu. 2017. EMF: Embedding multiple flows of information in existing traffic for concurrent communication among heterogeneous IoT devices. In *Proceedings of IEEE INFOCOM*.

[8] Zicheng Chi, Yan Li, Zhichuan Huang, Hongyu Sun, and Ting Zhu. 2019. Simultaneous bi-directional communications and data forwarding using a single ZigBee data stream. In *Proceedings of IEEE INFOCOM*.

[9] Zicheng Chi, Yan Li, Hongyu Sun, Yao Yao, Zheng Lu, and Ting Zhu. 2016. B2W2: N-Way concurrent communication for IoT devices. In *Proceedings of ACM SenSys*.

[10] Zicheng Chi, Yan Li, Yao Yao, and Ting Zhu. 2017. PMC: Parallel multi-protocol communication to heterogeneous IoT radios within a single WiFi channel. In *Proceedings of IEEE ICNP*.

[11] Douglas S. J. De Couto, Daniel Aguayo, John C. Bicket, and Robert Tappan Morris. 2003. A high-throughput path metric for multi-hop wireless routing. In *Proceedings of ACM MobiCom*.

[12] Rodrigo Fonseca, Omprakash Gnawali, Kyle Jamieson, and Philip Levis. 2007. Four-bit wireless link estimation. In *Proceedings of ACM HotNets*.

[13] FREESCALE. 2020. MC1321 Provided By FREESCALE. https://www.datasheetarchive.com/MC1321-datasheet.html.

[14] Xiuzhen Guo, Yuan He, Jia Zhang, and Haotian Jiang. 2019. WIDE: Physical-level CTC via digital emulation. In *Proceedings of IEEE IPSN*.

[15] Xiuzhen Guo, Yuan He, Xiaolong Zheng, Liangcheng Yu, and Omprakash Gnawali. 2018. Zigfi: Harnessing channel state information for cross-technology communication. In *Proceedings of IEEE INFOCOM*.

[16] Xiuzhen Guo, Yuan He, Xiaolong Zheng, Zihao Yu, and Yunhao Liu. 2019. LEGO-Fi: Transmitter-transparent CTC with cross-demapping. In *Proceedings of IEEE INFOCOM*.

[17] Xiuzhen Guo, Xiaolong Zheng, and Yuan He. 2017. Wizig: Cross-technology energy communication over a noisy channel. In *Proceedings of IEEE INFOCOM*.

[18] Wenchao Jiang, Song Min Kim, Zhijun Li, and Tian He. 2018. Achieving receiver-side cross-technology communication with cross-decoding. In *Proceedings of ACM MobiCom*.

[19] Wenchao Jiang, Zhimeng Yin, Song Min Kim, and Tian He. 2017. Transparent cross-technology communication over data traffic. In *Proceedings of IEEE INFOCOM*.

[20] Wenchao Jiang, Zhimeng Yin, Ruofeng Liu, Zhijun Li, Song Min Kim, and Tian He. 2017. BlueBee: A 10,000× faster cross-technology communication via PHY emulation. In *Proceedings of ACM SenSys*.

[21] Bryce Kellogg, Vamsi Talla, Shyamnath Gollakota, and Joshua R. Smith. 2016. Passive Wi-Fi: Bringing low power to Wi-Fi transmissions. In *Proceedings of USENIX NSDI*.

[22] Song Min Kim and Tian He. 2015. FreeBee: Cross-technology communication via free side-channel. In *Proceedings of ACM MobiCom*.

[23] Song Min Kim, Shuai Wang, and Tian He. 2015. cETX: Incorporating spatiotemporal correlation for better wireless networking. In *Proceedings of ACM SenSys*.

[24] Yan Li, Zicheng Chi, Xin Liu, and Ting Zhu. 2018. Chiron: Concurrent high throughput communication for IoT devices. In *Proceedings of ACM MobiSys*.
[25] Yan Li, Zicheng Chi, Xin Liu, and Ting Zhu. 2018. Passive-ZigBee: Enabling ZigBee communication in IoT networks with 1000×+ less power consumption. In *Proceedings of ACM SenSys*.
[26] Zhijun Li and Tian He. 2017. WEBee: Physical-layer cross-technology communication via emulation. In *Proceedings of ACM MobiCom*.
[27] Zhijun Li and Tian He. 2018. LongBee: Enabling long-range cross-technology communication. In *Proceedings of IEEE INFOCOM*.
[28] Nordic. 2020. nRF24E1/nRF9E5 Provided By Nordic. https://www.keil.com/dd/docs/datashts/nordic/nrf24e1.pdf.
[29] Daniele Puccinelli and Martin Haenggi. 2008. DUCHY: Double cost field hybrid link estimation for low-power wireless sensor networks. In *Proceedings of Hot EmNets*.
[30] Kannan Srinivasan, Prabal Dutta, Arsalan Tavakoli, and Philip Levis. 2006. Understanding the causes of packet delivery success and failure in dense wireless sensor networks. In *Proceedings of ACM SenSys*.
[31] TI. 2020. CC2530 Provided By TI. https://www.ti.com.cn/product/cn/CC2530.
[32] Shuai Wang, Zhimeng Yin, Zhijun Li, and Tian He. 2018. Networking support for physical-layer cross-technology communication. In *Proceedings of IEEE ICNP*.
[33] Wei Wang, Xin Liu, Yao Yao, Yan Pan, Zicheng Chi, and Ting Zhu. 2019. CRF: Coexistent routing and flooding using WiFi packets in heterogeneous IoT networks. In *Proceedings of IEEE INFOCOM*.
[34] Wei Wang, Tiantian Xie, Xin Liu, Yao Yao, and Ting Zhu. 2018. ECT: Exploiting cross-technology transmission for reducing packet delivery delay in IoT networks. In *Proceedings of IEEE INFOCOM*.
[35] Alec Woo, Terence Tong, and David E. Culler. 2003. Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proceedings of ACM SenSys*.
[36] Zhimeng Yin, Wenchao Jiang, Song Min Kim, and Tian He. 2017. C-Morse: Cross-technology communication with transparent Morse coding. In *Proceedings of IEEE INFOCOM*.
[37] Zhimeng Yin, Zhijun Li, Song Min Kim, and Tian He. 2018. Explicit channel coordination via cross-technology communication. In *Proceedings of ACM MobiSys*.
[38] Zihao Yu, Chengkun Jiang, Yuan He, Xiaolong Zheng, and Xiuzhen Guo. 2018. Crocs: Cross-technology clock synchronization for WiFi and ZigBee. In *Proceedings of ACM EWSN*.
[39] Jia Zhang, Xiuzhen Guo, Haotian Jiang, Xiaolong Zheng, and Yuan He. 2020. Link quality estimation of cross-technology communication. In *Proceedings of IEEE INFOCOM*.
[40] Marco Zuniga and Bhaskar Krishnamachari. 2007. An analysis of unreliability and asymmetry in low-power wireless links. *ACM TOSN* 3, 2 (2007).

Received October 2020; revised June 2021; accepted August 2021