WIDE: Physical-level CTC via Digital Emulation

Yuan He, *Senior Member, IEEE,* Xiuzhen Guo*, *Student Member, IEEE,* Jia Zhang, *Student Member, IEEE,* and Haotian Jiang, *Student Member, IEEE*

Abstract*—Cross-Technology Communication (CTC) is an emerging technique that enables direct communication across different wireless technologies. Recent works achieve physical-level CTC by emulating the standard time-domain waveform of the receiver. This method faces the challenges of inherent unreliability due to the imperfect emulation. Different from analog emulation, we propose a novel concept named digital emulation, which stems from the following insight: The receiver relies on phase shift rather than the phase itself to decode signals. Instead of emulating the original time-domain waveform, the sender emulates the phase shift associated with the desired signals. Clearly there are multiple different phase sequences that correspond to the same signs of phase shifts. Digital emulation has flexibility in setting the phase values in the emulated signals, which is effective in reducing emulation errors and enhancing the reliability of CTC. The key point of digital emulation is generic and applicable to a set of CTCs, where the transmitter has a wider bandwidth for emulation and the receiver decoding is based on the phase shift. In this paper, we implement our proposal as WIDE, a physical-level CTC via digital emulation from WiFi to ZigBee. We conduct extensive experiments to evaluate the performance of WIDE. The results show that WIDE significantly improves the Packet Reception Ratio (PRR) from 41.7% to 86.2%, which is $2\times$ of WEBee's, an existing representative physical-level CTC.

***Index Terms*—Cross-Technology Communication; Digital emulation**

I. Introduction

The proliferation of Internet of Things (IoT) applications brings about the increasingly dense deployments of various wireless devices, which causes a more serious coexistence of heterogeneous wireless technologies [1], [2], [3]. Under such circumstances, cross-technology communication (CTC) is an emerging technique to enable direct communication among devices that follow different communication standards. CTC provides a cost-effective and efficient solution to build connections among heterogeneous wireless devices, with benefits in the ability of interference management [4], [5], [6], in-situ data exchange [7], [8], [9], [10], and inter-operation among wireless devices [11], [12], [13].

CTC technique can be applied in many scenarios. Here are some examples: (1) CTC provides more efficient channel coordination by exchanging explicit channel allocation messages among coexisting WiFi and ZigBee devices, in replace of passive listening or channel assessment [12]. (2) Instead of multi-hop transmission of ZigBee nodes in ad-hoc networks, CTC can directly update the firmware and software of ZigBee nodes by using the CTC link from WiFi to ZigBee, so as to save energy and reduce latency of ZigBee nodes [14]. (3) In our daily life, the smartphone with WiFi radio can directly communicate with a workout equipment with ZigBee radio to make customized workout plan in the gym [15]. Similarly, in the smart home and smart offices, the smartphone can directly control ZigBee devices (e.g. sensors) through the CTC link, even when a special gateway is absent.

Early CTC works establish communication channels based on the packet-level, which manipulate transmitted packets and use the packet length [16], the received signal strength [17], [18], [19], [20], or the transmission timings [21], [22] as the information carrier. Recent works propose physical-level CTC. WEBee [23] uses the high-speed WiFi radio to emulate the standard half sine waveform of the low-speed ZigBee radio by carefully selecting the payload of the WiFi packet. BlueBee [24] modifies the payload of BLE to emulate the signal of ZigBee. XBee [15] realizes CTC from ZigBee to BLE based on cross-demapping, which decodes the ZigBee packet by observing the bit patterns obtained at the BLE receiver.

Almost all existing physical-level CTCs are realized by analog emulation method, namely that the sender emulates the standard time-domain waveform of the receiver. Whereas, the analog emulation-based CTCs may not be suitable for applications that require high reliability (e.g. data dissemination and reliable network flooding) due to the limited Packet Reception Ratio (PRR). This is because the analog emulated signal cannot perfectly match the desired signal. The protocol standard of the sender is different from that of the receiver. The hardware restrictions also affect the emulation result. So there are inevitable distortions between the desired waveform and the emulated waveform. To realize high reliability in the practical applications, the emulated packets have to be retransmitted. As a result, the efficiency and throughput of analog emulation-based CTC degrade.

We find that the decoding of the ZigBee receiver doesn't rely on the specific shape of the time-domain waveform. Intrinsically, the ZigBee receiver decodes data based on phase shift rather than phase itself. Therefore, different from analog emulation, we propose a novel concept **Digital Emulation** for physical-level CTC. Instead of emulating the standard time-domain waveform of the receiver, the sender emulates the phase shift associated with the desired signals. There are lots of phase sequences which satisfy the requirement of the phase shift for the receiver decoding. We have the opportunity to select an appropriate phase sequence with relatively small emulation errors to achieve a reliable CTC.

The concept of digital emulation is generic and applicable to a set of CTCs, where the transmitter has a wider bandwidth

Yuan He, Xiuzhen Guo, Jia Zhang, and Haotian Jiang are with the School of Software and BNRist, Tsinghua University, P.R. China.

E-mail: heyuan@mail.tsinghua.edu.cn, {guoxz16, jia-zhan15, jht15}@mails.tsinghua.edu.cn

*Xiuzhen Guo is the corresponding author.

for emulation and the receiver decoding is based on the phase shift. In this paper, we implement our proposal as WIDE, a physical-level CTC via digital emulation from WiFi to ZigBee. WIDE selects an appropriate phase sequence for WiFi emulation, which makes the decoded binary phase shift sequence at the ZigBee receiver more accurate and robust.

Specifically, we first select the square wave as a basic unit to generate a set of ladder shaped phase sequences. The corresponding phase shift sequence of the ladder shaped phase sequence is stable within a demodulation period and satisfies the requirement of a ZigBee symbol. WiFi modifies the content of the payload to accomplish the process of emulation [23], which makes the phase shift of the payload resembles that of the desired phase shift. Then we adopt a greedy algorithm to generate the initial phase sequence. In addition, we analyze the errors caused by Cyclic Prefix (CP) during the WiFi emulation and propose an algorithm named Secondary Adjustment based on FEedback (SAFE) to further optimize the phase sequence. In this way, we can get the appropriate phase sequence for phase shift emulation. The ZigBee packet reception ratio of WIDE can be improved from 41.7% to 86.2%, which is $2\times$ of WEBee's, an representative physical-level CTC. Our contributions are summarized as follows.

- We propose a novel concept, digital emulation, for physical-level CTC. Instead of emulating the standard time-domain waveform of the receiver, we select an appropriate phase sequence to emulate the phase shift of the receiver directly. Without modifying the firmware or hardware of both WiFi and ZigBee devices, our design is a transparent method that can be easily deployed in existing WiFi infrastructure with broad applicability. The method of digital emulation is generic and applicable to a set of CTCs, where the transmitter has a wider bandwidth for emulation and the receiver decoding is based on the phase shift.
- We design WIDE, a physical-level CTC via digital emulation from WiFi to ZigBee. In WIDE, we address several challenges, including the phase sequence generation and the phase sequence optimization, to select an appropriate phase sequence for phase shift emulation.
- We implement WIDE on both the USRP N210 platform and the commodity device. The experimental results demonstrate that WIDE achieves high reliable CTC from WiFi to ZigBee. WIDE improves the Packet Reception Ratio (PRR) of ZigBee packets from 41.7% to 86.2%, which is $2\times$ of WEBee's, an existing representative physical-level CTC.

The rest of this paper is organized as follows. Section II discusses related works. Section III compares the analog emulation and the digital emulation. We elaborate on our design in Section IV. Section V presents the evaluation results. We conclude this work in Section VI.

II. Related Works

The incompatibility between technologies and the asymmetry of device capacity are the two major challenges of CTC. According to the method to cope with the challenges, we can classify the existing works into two categories: packet-level CTC and physical-level CTC.

Packet-level CTC. By manipulating the packets as information carrier, packet-level CTC builds an accessible side channel for CTC, such as the received signal strength [17], [18], [19], [20], the packet length [16], the transmission timings [21], [22], and the channel state information [25], [26]. FreeBee [21] embeds symbols into beacons by shifting their transmission timings. The date rate of FreeBee is limited by the beacon rate which is usually 102.4ms per beacon for commercial WiFi devices, however. Other works propose the energy profile as a new information carrier to exchange the data without a gateway. Esense [17] is the first work that uses energy sampling realizing data transmission from the WiFi to the ZigBee device. It aims at building an alphabet of implicit data using the packet duration information. Since the communication channel is intrinsically noisy, it is not a trivial to reduce the harmful impact of noise. The impact of noise on CTC throughput is analyzed in WiZig [18], which adjusts the transmission power to encode multiple bits. StripComm [19] is a novel interference resilient CTC tailored to the coexisting environment. StripComm leverages the idea of Manchester Coding and proposes a novel interference-aware coding mechanism. HoWiEs [16] controls the WiFi packet length and encodes bits by the length of packet on-air time. C-Morse [20] uses the combination of the short WiFi packets and the long WiFi packets with short intervals to construct the recognizable energy patterns at the ZigBee receiver. EMF [27] leverages the independency among different window sizes for embedding different pieces of information in a string of existing packets. B2W2 [25] and ZigFi [26] exploit the feature of Channel State Information (CSI) to realize communication from BLE to WiFi and ZigBee to WiFi respectively. The throughput of packet-level CTC, however, is bounded by the granularity of packet manipulation, which is at the magnitude of millisecond.

Physical-level CTC. Physical-level CTC aims at creating compliance across technologies and building the CTC channel right at the physical layer [28], [11]. WEBee [23] proposes physical-level emulation, which uses the high-speed WiFi radio to emulate the standard ZigBee time-domain signals of the low-speed ZigBee radio. Specifically, WEBee chooses the payload of a WiFi frame so that a portion of this WiFi frame is recognized by commodity ZigBee devices transparently as a legitimate ZigBee frame. In order to improve the reliability of WEBee, TwinBee [29] proposes a chip-combining coding scheme to recover chip errors introduced by imperfect signal emulation. LongBee [30] is another improved CTC work of WEBee, which extends the communication range of CTC to support long-range IoT applications. In terms of signal emulation, LongBee works similar to WEBee. Moreover, LongBee combines the high transmission power of WiFi and the fine receiving sensitivity of ZigBee together to increase the CTC communication range significantly. LEGO-Fi [31] achieves physical-level CTC from ZigBee to WiFi by leveraging cross-demapping, which stems two key technique insights. First, a ZigBee packet leaves distinguishable features when passing the WiFi modules. Second, compared to ZigBee's

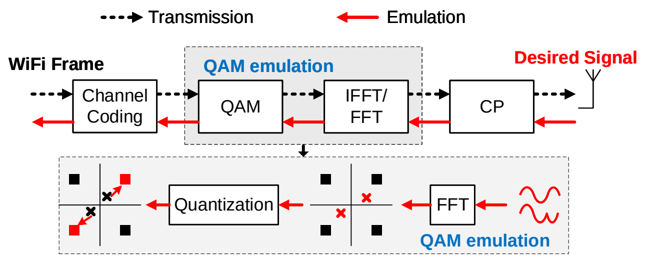


Fig. 1. The workflow of analog emulation

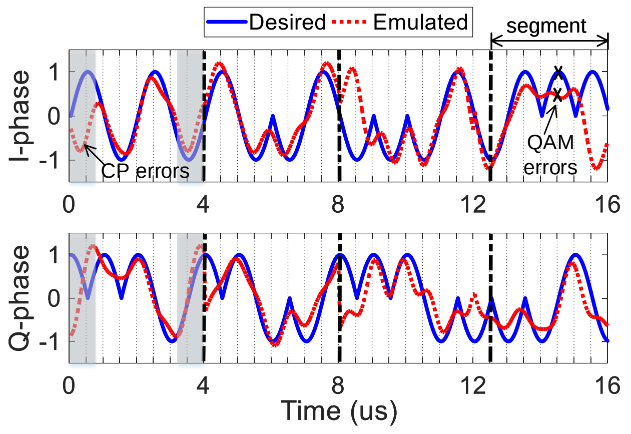


Fig. 2. Desired and emulated signals via analog emulation

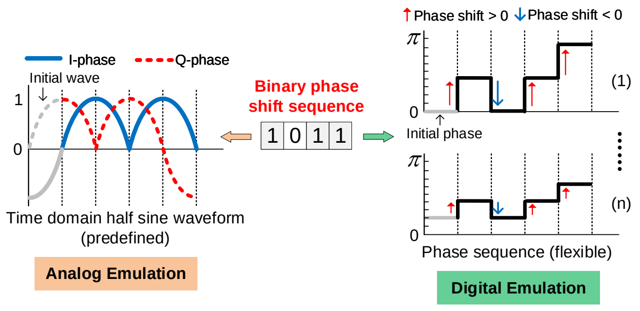


Fig. 3. The comparison of analog emulation and digital emulation

simple encoding and modulation schemes, the rich processing capacity of WiFi offers extra flexibility to process a ZigBee packet. PMC [32] enables parallel communication to multiple ZigBee and WiFi devices. BlueBee [24] modifies the payload of BLE to emulate the signal of ZigBee. XBee [15] realizes CTC from ZigBee to BLE based on cross-decoding, which decodes a ZigBee packet by observing the bit patterns obtained at the BLE receiver. Scylla [13] is a software control CTC which allows multiple wireless stacks to coexist on top of a single radio chip, thereby simultaneously offering multiple communication interfaces.

III. Analog Emulation Vs. Digital Emulation

In this Section, we compare the analog emulation and the digital emulation. We introduce the workflow and limitation of the physical-level CTC via analog emulation. We further introduce the motivation and benefit of the physical-level CTC via digital emulation.

A. Analog Emulation

1) The workflow of analog emulation: In order to achieve the physical-level CTC via analog emulation, the payload of a WiFi frame is elaborately selected to construct a legitimate ZigBee frame via emulating the ZigBee standard half sine waveform closely. A ZigBee symbol with $16\mu s$ has to be segmented and each $4\mu s$-segment is emulated by a WiFi symbol. The process of analog emulation is shown in Fig .1. The desired ZigBee signal is fed into the FFT module and we select the nearest QAM constellation points to construct the payload and emulate the ZigBee signal. After selecting the payload, the process of WiFi transmission is a reverse direction. The WiFi sender adds the cyclic prefixing (CP) to the time-domain signal and then sends it by using the RF radio, just like sending the normal WiFi signal. The entire procedure is transparent to the hardware layer of WiFi device and all of the modification is conducted on the software layer.

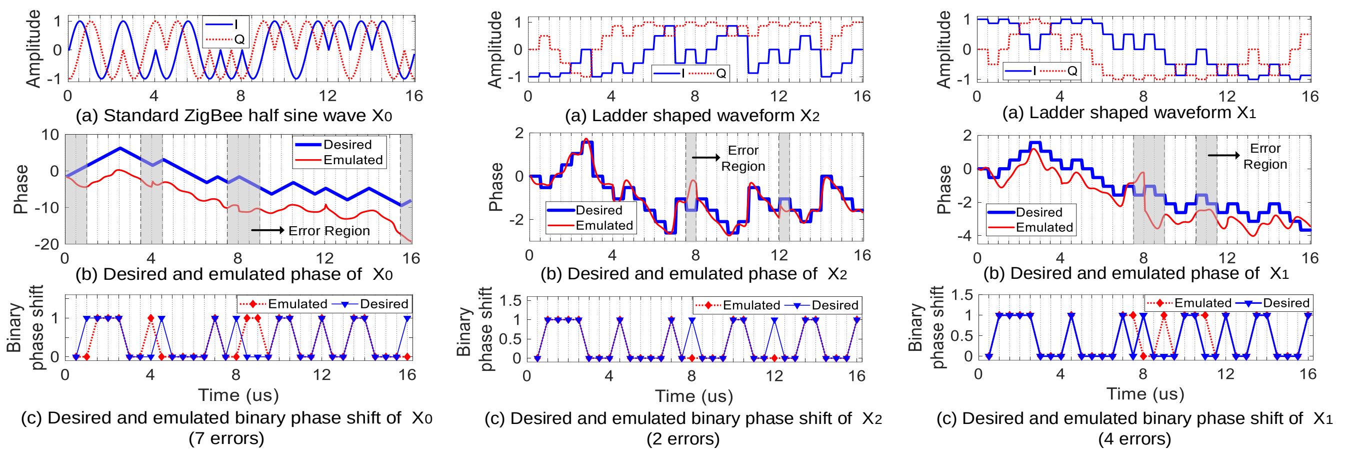
2) The limitation of analog emulation: Due to the incompatibility of different protocol standards and the hardware restrictions, the analog emulated signal cannot perfectly match the desired signal. The analog emulation result of ZigBee symbol “0” is shown in Fig. 2. We can find that the emulated signals have distortion compared with the standard ZigBee half sine signals. As for analog emulation, there are mainly two types of intrinsic errors.

QAM emulation errors. QAM emulation is the core of analog emulation, where the standard ZigBee time-domain signals are fed into the FFT of WiFi to find the corresponding QAM constellation points. WiFi’s predefined QAM points are limited and discrete, so time-domain signals of ZigBee may not be perfectly mapped to these QAM points predefined by WiFi. In addition, WiFi has 64 subcarriers and there are only seven subcarriers overlapping with ZigBee. As a result, only seven nearest QAM points with the minimum Euclid Distance to the FFT coefficients are selected to emulate the standard ZigBee time-domain signals. When the ZigBee receiver demodulates the emulated signal, quantization errors cannot be avoided.

CP errors. Another source of emulation errors comes from the WiFi’s cyclic prefixing (CP). CP is a 0.8us guard interval in each WiFi symbol, which is copied from the right WiFi symbol and pasted into the left of this symbol. In this way, the front segment of WiFi signals is same with the end segment of WiFi signals. Whereas, there is no such repetition in ZigBee signals. As a result, the CP errors of emulated signals are also out of the control. Furthermore, the desired signals are predefined and fixed. So the above analog emulation errors are inevitable.

B. Digital Emulation

1) The feasibility of digital emulation: We find that the decoding of the ZigBee receiver doesn’t directly rely on the specific shape of waveform. Intrinsically, ZigBee uses phase shift to modulate symbols. ZigBee outputs “1” if the phase



(a) The standard half sine wave (b) A ladder shaped waveform (c) Another ladder shaped waveform

Fig. 4. The emulation result of different waveforms

shift is bigger than 0° and otherwise outputs “0”. After collecting 32 binary phase shifts, the ZigBee receiver maps this binary phase shift sequence into a 4-bit symbol, according to DSSS process.

Based on the finding that the receiver decoding is based on the phase shift, we propose digital emulation, where the sender directly produces proper sequence of phase shift for emulation. As shown in Fig. 3, there are lots of phase sequences which satisfy the requirement of binary phase shift sequence of ZigBee. Different phase sequences correspond to different waveforms. WiFi can construct different payloads to emulate these waveforms. So in addition to the standard ZigBee half sine waveform, other types of waveforms can also be correctly decoded as long as these waveforms have the same binary phase shift sequences.

We conduct several experiments to verify the feasibility of digital emulation. The standard half sine waveform of ZigBee symbol “F” is shown in Fig. 4(a)(a). The emulated phase sequence and the binary phase shift sequence are shown in Fig. 4(a)(b) and Fig. 4(a)(c) respectively. Due to the emulation distortion, except the first and the last, there are 6 wrong binary phase shift values. Another ladder shaped waveform is shown in Fig. 4(b)(a). Its corresponding desired phase sequence and emulated phase sequence are shown in Fig. 4(b)(b). The decoded binary phase shift sequence is shown in Fig. 4(b)(c). We find that there are only 2 wrong bits, which can be easily mapped to the ZigBee symbol “F” according to DSSS.

2) The flexibility and challenges of digital emulation: Compared with analog emulation, digital emulation is more flexible and robust. The phase sequence with desirable phase shift sequence is not unique. Although the binary phase shift sequence of a ZigBee symbol is predefined and fixed, there are lots of phase sequences which satisfy the requirement of the binary phase shift sequence. Different phase sequences correspond to different time-domain waveforms. The performance of WiFi to emulate different phase sequence based on QAM emulation is different. Therefore, we have the opportunity to reduce the QAM errors and CP errors of WiFi emulation by

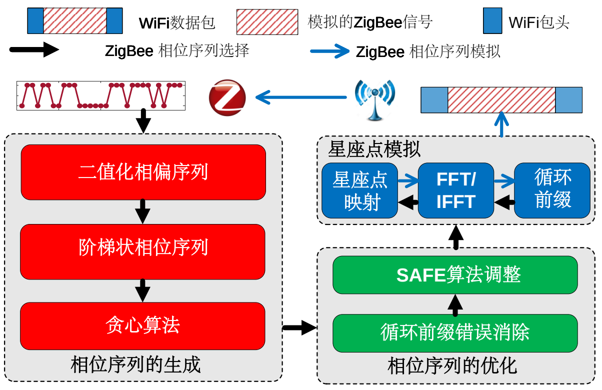


Fig. 5. The framework of WIDE

selecting an appropriate phase sequence. But the selection of an appropriate phase sequence is challenging. Not all phase sequences that satisfy the binary phase shift requirement will have a better emulation result. Another ladder shaped waveform is shown in Fig. 4(c)(a). Its phase sequence and decoding result are shown in Fig. 4(c)(b) and Fig. 4(c)(c) respectively. There are 4 wrong binary phase shift values except the first and the last. So how to select an appropriate phase sequence for WiFi emulation remains a challenging task and needs further study.

IV. Design

In this section, we will first present an overview of WIDE and then introduce the design details.

A. Overview

The framework of WIDE is shown in Fig. 5 and the workflow of WIDE is as follows.

(i) **Phase sequence generation:** First, WIDE selects the square wave as a basic unit to generate a set of ladder shaped phase sequences, which satisfy the binary phase shift requirement. We propose the metric Statistical Hamming Distance

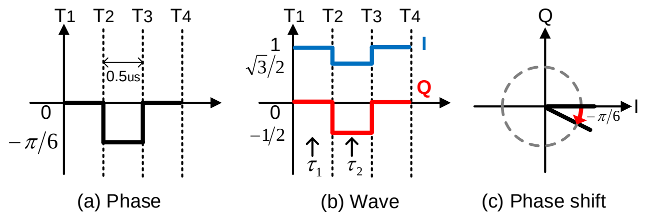


Fig. 6. The basic idea of WIDE

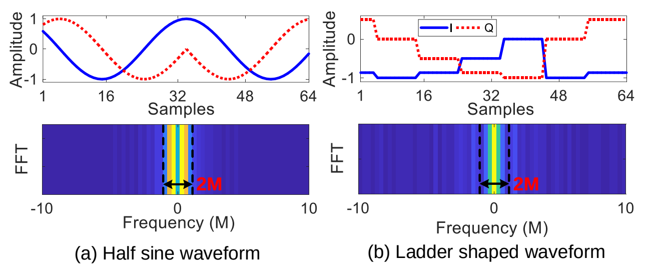


Fig. 7. The FFT results of half sine waveform and ladder shaped waveform

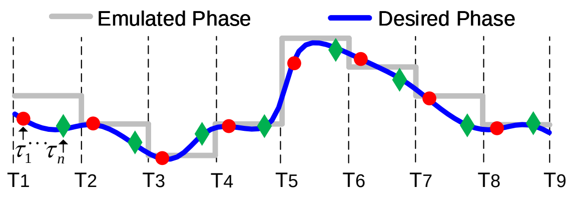


Fig. 8. The illustration of SHD

(SHD) to quantify the degree of distortion between the desired phase sequence and the emulated phase sequence. Then we adopt a greedy algorithm to generate an initial phase sequence for phase shift emulation.

(ii) **Phase sequence optimization:** We analyze the CP errors after WiFi emulation and alleviate the CP errors by slightly adjusting the phase at some specific positions. Furthermore, WIDE proposes an algorithm named Secondary Adjustment based on FEedback (SAFE), which adjusts the phase sequence again according to the feedback result of WiFi emulation. In this way, we get an appropriate phase sequence for WiFi emulation.

(iii) **Phase sequence emulation:** The phase sequence corresponds to a desired waveform. The desired waveform is fed into the FFT module and we select the nearest QAM constellation points to construct the WiFi payload and emulate the desired waveform. After emulation, the phase sequence of WiFi payload resembles to desired phase sequence. WiFi header, preamble, and tail are ignored by the ZigBee receiver. Then the WiFi payload can be considered as a legitimate ZigBee frame and ZigBee symbols can be decoded successfully.

Therefore, the key point of the digital emulation is the selection of the appropriate phase sequence. Among 16 ZigBee symbols, each of them corresponds to a binary phase shift sequence. We need to select 16 appropriate phase sequences to realize the digital emulation for ZigBee symbols. We operate the process of selecting phase sequence locally and then get a mapping table from symbol to phase sequence. This mapping table can be loaded on the WiFi device prior to running WIDE so that the WiFi device is able to emulate these phase sequences by elaborately construct the payload. We introduce the design techniques, including phase sequence generation and phase sequence optimization more clearly as follows.

B. Phase Sequence Generation

1) Waveform unit: First, we need to select an wave as a basic unit to generate the phase sequence which satisfies the phase shift requirement of the receiver. Specifically, each ZigBee symbol corresponds to a 32-bit binary phase shift sequence and the ZigBee receiver demodulates the phase shift every $0.5\mu s$. If the binary phase shift is 1, the phase within two demodulation periods needs to increase and vice versa. For example, we vary the phase from 0 to $-\frac{\pi}{6}$ within $T_1 - T_3$ as shown in Fig. 6(a) and its corresponding time-domain waveform is shown in Fig. 6(b). There are two samples at τ_1 and τ_2, which satisfy that $T_1 \leq \tau_1 \leq T_2$, $T_2 \leq \tau_2 \leq T_3$, and $\tau_2 - \tau_1 = 0.5\mu s$. The phase shift between these two samples is $-\frac{\pi}{6}$, which is lower than $0°$ and the corresponding binary phase shift is 0, as shown in Fig. 6(c).

In order to guarantee the phase shift within a demodulation period is stable, we select the square wave as a basic unit to generate the phase sequence and the corresponding waveform. The frequency component required for emulating the square wave is higher than the sine wave, however, this problem is not difficult to handle. What we need to emulate is a ladder shaped waveform as shown in Fig. 7(b), as the binary phase shift sequence is composed of many consecutive “0” or “1”. From the FFT results of the standard half sine waveform and the ladder shaped waveform, we can find that the frequency components of the ladder shaped waveform are also concentrated in 2M. Therefore, it is feasible to emulate the ladder shaped waveform with the limited number of subcarriers within 2M bandwidth at the WiFi sender.

2) SHD Metric: Due to the limited availability of subcarriers, QAM quantization errors, and the impact of CP, there are inevitable distortions between the desired waveform and the emulated waveform. As shown in Fig. 8, the desired phase sequence and the emulated phase sequence are separately shown by the gray line and the blue line. Quantifing the degree of distortion helps us optimize the phase sequence and reduce the emulation errors. Usually, the hamming distance between the decoded binary phase shift sequence and the predefined binary phase shift sequence can be used to characterize the distortion between the emulated phase sequence and the desired phase sequence. Whereas, the hamming distance changes with the variation of the position of the first sample. Because the ZigBee receiver decodes different binary phase shift sequences when the first sample starts at different positions.

Therefore, we propose Statistical Hamming Distance (SHD) to quantify the degree of distortion between the desired phase sequence and the emulated phase sequence. SHD is also used as an objective function to select a phase sequence with minimal emulation errors regardless of the sampling positions.

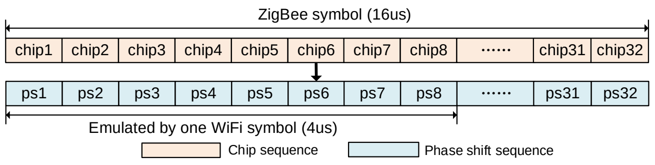

ZigBee symbol (16us)
chip1 chip2 chip3 chip4 chip5 chip6 chip7 chip8 ······ chip31 chip32
ps1 ps2 ps3 ps4 ps5 ps6 ps7 ps8 ······ ps31 ps32
Emulated by one WiFi symbol (4us)
Chip sequence
Phase shift sequence

Fig. 9. One Zigbee symbol is emulated by four WiFi symbols

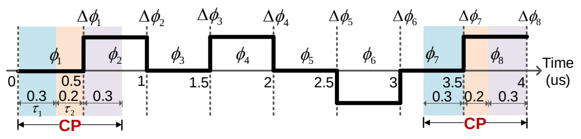


Fig. 10. The illustration of phase sequence initialization

The start position of the sampling obeys uniform distribution. We suppose the start position of sampling can be $\tau_1, \tau_2, ..., \tau_n$ and the corresponding hamming distance is $H_1, H_2, .., H_n$. SHD is defined as

$$SHD = \frac{1}{n}(H_1 + H_2 + ... + H_n) \tag{1}$$

The larger n is, the more convincing that the SHD can characterize the degree of distortion. We conduct many experiments and find that the performance gains from increasing the choice n are limited. We set n at 5 since this configuration already meets the requirement.

3) Phase sequence initialization: One ZigBee symbol corresponds to a chip sequence and a binary phase shift sequence (32-bit) as shown in Fig .9. A ZigBee symbol is 16μs and a WiFi symbol is 4μs. So the ZigBee symbol is divided into four segments. Each segment corresponds to a 8-bit phase sequence and is emulated by a WiFi symbol.

We suppose that the binary phase shift sequence of a ZigBee symbol is

$$\Delta\Phi = \{\Delta\Phi_1, \Delta\Phi_2, ..., \Delta\Phi_{n-1}, \Delta\Phi_n\}, n = 1, 2, ..., 32 \tag{2}$$

We suppose the phase sequence satisfying the requirement of binary phase shift sequence is

$$\Phi = \{\Phi_1, \Phi_2, ..., \Phi_{n-1}, \Phi_n\}, n = 1, 2, ..., 32 \tag{3}$$

The duration of each phase value Φ_n is 0.5μs, which is equal to the decoding period of ZigBee. If the initial phase value is φ and the absolute phase shift value between two consecutive phases is $\Delta\varphi$, the phase sequence in Eq. (3) can be generated by

$$\Phi_j = \begin{cases} \varphi & j = 1 \\ \Phi_{j-1} + \Delta\Phi_j \Delta\varphi & j = 2, ..., 32 \end{cases} \tag{4}$$

The waveform corresponds to this phase sequence is

$$x(n) = I(n) + Q(n) = cos(\Phi_n) + i * sin(\Phi_n), n = 1, 2, ..., 32 \tag{5}$$

The illustration of phase sequence initialization is shown in Fig. 10. We take a 4μs ZigBee segment as an example. The binary phase shift requirement is $\Delta\Phi_1\Delta\Phi_2\Delta\Phi_3\Delta\Phi_4\Delta\Phi_5\Delta\Phi_6\Delta\Phi_7\Delta\Phi_8 = 10100110$ and the corresponding result of phase sequence initialization is shown in Fig. 10.

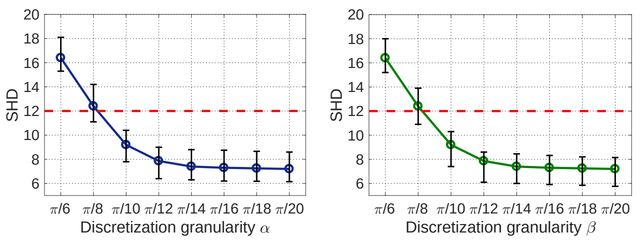


Fig. 11. SHD with the increasing of discretization granularity

According to Eq. (4), there are two factors that affect the phase sequence. One is the initial phase and the other one is the phase shift between two phases. The initial phase φ can be any value in $[0, 2\pi)$ and the phase shift $\Delta\varphi$ can be any value in $[0, \pi)$. Many sets of $(\varphi, \Delta\varphi)$ will generate different phase sequences. In order to simplify this problem and reduce the computation cost, we discretize the value of φ and $\Delta\varphi$.

We suppose the discretization granularity of initial phase φ is α and the discretization granularity of phase shift $\Delta\varphi$ is β. Discretization granularity will impact the WiFi emulation result and we conduct experiments to choose suitable discretization granularity values. The discretization granularity values from $\frac{\pi}{4}$ to $\frac{\pi}{20}$. Different discretization granularity values generate different phase sequence for WiFi emulation. The experimental results are shown in Fig. 11(a) and Fig. 11(b). We can find that SHD decreases with the increasing of discretization granularity. The commercial ZigBee device sets a threshold to tolerate a certain number of chip errors, which by default is 12. Considering to the computational complexity, we choose the discretization granularity values as $\alpha = \frac{\pi}{12}$ and $\beta = \frac{\pi}{12}$. In this condition, the optional value of φ is $\varphi = m \times \alpha(m = 0, 1, ..., \frac{2\pi}{\alpha} - 1)$, the optional value of $\Delta\varphi$ is $\Delta\varphi = n \times \beta(n = 0, 1, ..., \frac{\pi}{\beta} - 1)$. Next, we adopt a greedy algorithm to get the appropriate set of $(\varphi, \Delta\varphi)$ and generate the initial phase sequence Φ. The optimization function is

$$\min \ SHD \tag{6}$$

$$s.t \begin{cases} \varphi = m * \frac{1}{12}\pi & m = 0, 1, 2, ..., 23 \\ \Delta\varphi = n * \frac{1}{12}\pi & n = 0, 1, 2, ..., 11 \end{cases} \tag{7}$$

C. Phase Sequence Optimization

In this section, we propose phase sequence optimization, which leverages the feasibility in setting phase values to reduce emulation errors and enhance the reliability of CTC. There are two phase sequence optimization methods including CP errors alleviation and SAFE Algorithm.

1) CP errors alleviation: CP errors alleviation aims at reducing the harmful impact of WiFi CP and maximizing the decoding probability of the emulated ZigBee signal. By fully searching the phase optimization space and adjusting the phase values, the method of CP errors alleviation ensures that the

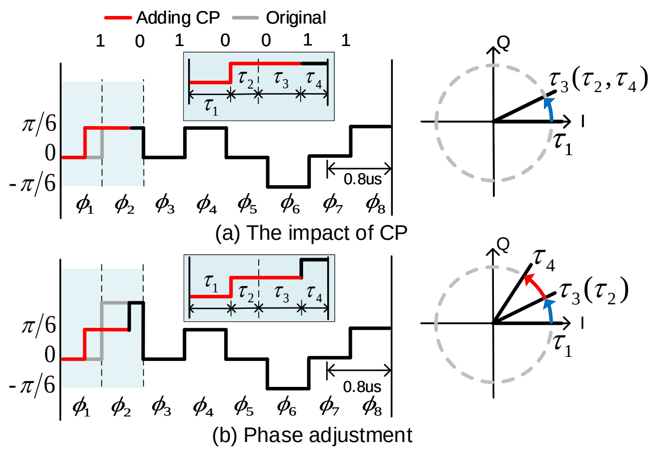


Fig. 12. Forward CP and $\Delta\Phi1 = \Delta\Phi7$

$\Delta\Phi_1$	$\Delta\Phi_7$	Φ_2	$P(A)$	$P(W\|A)$	$P(B)$	$P(W\|B)$	EP
1	1	$\Phi_8 + \Delta\Phi$	0.6	0	0.4	0	0
1	0	$\Phi_8 + \Delta\Phi$	0.6	1	0.4	0	0.6
0	0	$\Phi_8 - \Delta\Phi$	0.6	0	0.4	0	0
0	1	$\Phi_8 - \Delta\Phi$	0.6	1	0.4	0	0.6

TABLE I
THE OPTIMIZATION OF CP ERRORS

decoding result of phase sequence after WiFi CP is same with that before WiFi CP.

First, we introduce the harmful impact of WiFi CP on the emulated ZigBee signals. CP is copied from the right of the WiFi symbol and pasted into (overwrite) the left of the symbol. Enforced by WiFi CP, the front 0.8μs of WiFi symbol is same with the end 0.8μs of WiFi symbol. One ZigBee segment with eight phase values is emulated by a WiFi symbol. Each phase value lasts 0.5μs. As shown in Fig. 10, phase $\Phi1$ and the first 0.3μs of phase $\Phi2$ are affected by WiFi CP. We suppose $P(A)$ is the probability of the sampling position within the first 0.3μs of $\Phi1$ (τ_1), $P(B)$ is the probability of the sampling position within the last 0.2μs of $\Phi1$ (τ_2). $P(A)$ and $P(B)$ satisfy that $P(A) = \frac{0.3}{0.5} = 0.6$ and $P(A) = \frac{0.2}{0.5} = 0.4$. We suppose the error probability when the sampling position within the first 0.3μs of $\Phi1$ (τ_1) is $P(W|A)$. The error probability when the sampling position within the last 0.2μs of $\Phi1$ (τ_2) is $P(W|B)$. The probability of correct decoding P at the ZigBee receiver is:

$$P = 1 - EP = 1 - (P(A)P(W|A) + P(B)P(W|B)) \quad (8)$$

For maximizing the correct decoding probability, we search the phase selection space and adjust the specific phase values to minimize the value of $P(W|A)$ and $P(W|B)$ by adjusting the specific phase values.

The values of $P(W|A)$ and $P(W|B)$ are related with the sign of phase shifts $\Delta\Phi1$ and $\Delta\Phi7$ and we take the condition $\{\Delta\Phi1 = \Delta\Phi7 = 1\}$ as an example to illustrate how to minimize the value of $P(W|A)$ and $P(W|B)$. We suppose the binary phase shift sequence is $\Delta\Phi = \{\Delta\Phi_1, \Delta\Phi_2, ..., \Delta\Phi_7\} = \{1010011\}$, the original phase sequence $\Phi = \{\Phi_1, \Phi_2, ..., \Phi_8\}$ is shown by the gray/black line in the left of Fig. 12(a). The phase sequence after adding CP is shown by the red/black line in the left of Fig. 12(a). We analyze the demodulation result of the samples affected by the CP as shown in the right of Fig. 12(a). When the first sample is within in τ_1, the second sample is within in τ_3 and the demodulated result is 1. Whereas, if the first sample is within in τ_2, the second sample is within in τ_4 and the demodulated result is wrong. In this condition,

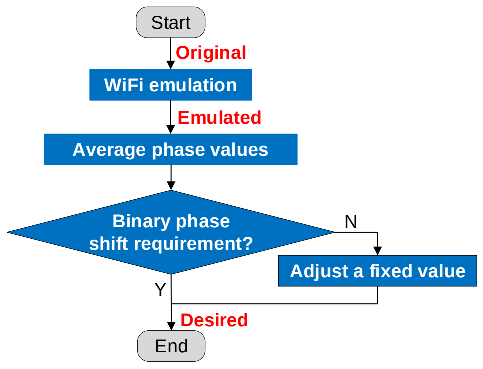


Fig. 13. The workflow of SAFE

$P(W|A) = 0$ and $P(W|B)$=1, respectively. So the correct decoding probability (P) is 0.6 and the error probability (EP) is 0.4. In order to correct the demodulation error, we increase original Φ_2 by $\Delta\Phi$ to make the new Φ_2 larger than original Φ_8. The adjustment result is shown by the gray/black line in the left of Fig. 12(b) and the phase sequence after adding CP is shown by the red/black line in the left of Fig. 12(b). In this condition, the correct decoding probability (P) is 100%.

There are totally four CP optimization cases and the optimization of all cases is shown in Table I. We adjust the value of $\Phi2$ based on the sign of phase shift $\Delta\Phi1$ and $\Delta\Phi7$. As shown in Table I, the error probability (EP) caused by CP can be eliminated when $\Delta\Phi_1$ is equal to $\Delta\Phi_7$. When $\Delta\Phi_1$ is different to $\Delta\Phi_7$, the EP caused by CP can also be reduced to 0.6. In this way, the harmful impact of CP can be effectively alleviated.

2) SAFE Algorithm: We further propose an algorithm named Secondary Adjustment based on FEedback (SAFE) to adjust the phase sequence according to the feedback result of one-time WiFi local emulation to reduce the emulation distortions.

The process of SAFE algorithm is as follows.

(1) The phase sequence generated by the previous steps is fed into the WiFi emulation modules shown in Fig. 1 and we can obtain the emulated phase sequence.

(2) We calculate the average phase value of the emulated phase sequence within a demodulation period as a new phase value to construct an adjusted phase sequence.

(3) We verify whether the adjusted phase sequence meets the binary phase shift requirement of the ZigBee symbol. If the phase shift sequence of the adjusted phase sequence is right, the adjusted phase sequence is the desired phase sequence. If the phase shift sequence of the adjusted phase sequence contradicts the requirement of the ZigBee symbol, we directly further adjust a fixed phase shift based on the previous adjusted phase sequence until zigbee's decoding requirement is met.

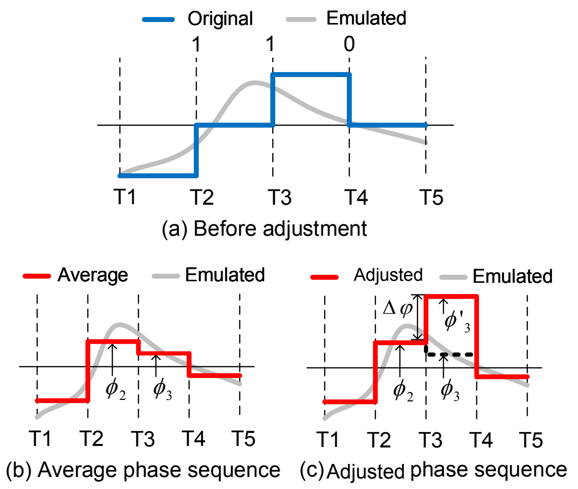


Fig. 14. An example of SAFE

We take Fig. 14(a) as an example to illustrate the process of SAFE algorithm. The original phase sequence generated by previous steps and the emulated phase sequence after WiFi emulation are shown in the blue and gray lines respectively. We calculate the average phase value within each demodulation period of the emulated phase sequence as a new phase value. As shown in Fig. 14(b), the new calculated phase values are $\Phi_1, \Phi_2, \Phi_3, \Phi_4$. Whereas, the new phase value obtained by this method may contradict the requirement of the binary phase shift. The average phase value Φ_3 of the emulated phase sequence within (T_3, T_4) is lower than the average phase value Φ_2 of the emulated phase sequence within (T_2, T_3). Whereas, the required binary phase shift is "1". The phase sequence is contradict with the phase shift requirement of ZigBee decoding. In this condition, we directly adjust an fixed phase shift based on the previous phase value. As shown in Fig. 14(c), we adjust the phase value within (T_3, T_4) as Φ_3', where $\Phi_3' = \Phi_2 + \Delta\varphi$ and $\Delta\varphi$ is selected by the previous greedy algorithm. In this way, we obtain the desired phase sequence for WiFi emulation.

D. Parallel Communication

As we all know, a WiFi channel overlaps with several ZigBee channels. WIDE can support parallel CTC from WiFi to ZigBee with the channel mapping scheme. The central frequency of a WiFi channel is set as 2440MHz, the two regions of WiFi subcarriers [-13 to -19] and [13 to 19] can be utilized to achieve two parallel CTC with standard ZigBee channel 17 and channel 19 as shown in Fig. 15. We note many commodity WiFi radios (e.g., Atheros AR9485, AR5112, and AR2425) can set their central frequency.

E. Discussions

1) The reliability of digital emulation: The key reason that CTC cannot achieve reliable communication is the incompatibility of physical layer among different wireless technologies. Due to the difference in modulation mechanism (WiFi OFDM vs. ZigBee OQPSK) and modulation speed (WiFi 1symbol/4us

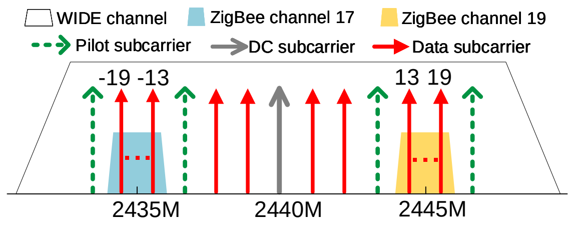


Fig. 15. Channel Mapping for parallel communication

vs. ZigBee 1chip/1us), there are inevitable emulation errors between WiFi emulated signal and ZigBee standard signal. Therefore, minimizing the emulation errors is the key to achieve reliable communication. In WIDE, what the WiFi sender emulates is the desired phase shift sequence rather than the time-domain waveform. Although there are errors between QAM points and ZigBee symbols, digital emulation enables a WiFi sender to select QAM points that best match the desired phase shift sequence. It is worth noticing that there is considerable flexibility here for the sender to select the QAM points. WIDE leverages the flexibility of QAM point selection at the WiFi sender and the error tolerance of DSSS mechanism at the ZigBee receiver to improve the reliability of CTC.

2) The generalizability of digital emulation: The modulation of ZigBee is Offset Quadrature Phase-Shift Keying (OQPSK) with halfsine pulse shaping [33]. The design of WIDE is suitable for the commercial devices whose decoding mechanism is based on phase information. For example, mainstream commercial ZigBee devices CC2530 provided by TI [34], MC1321 provided by FREESCALE [35], AT86RF230 provided by ATMEL [36], and nRF24E1/nRF9E5 provided by Nordic [37] leverage the phase information to decode ZigBee signals. WIDE cannot be implemented on other ZigBee devices which don't rely on phase information to decode.

3) Comparison with TwinBee: TwinBee [29] proposes a specific chip-combining coding to recover the errors in error-prone chips. The basic idea of the chip-combining coding is leveraging the cyclic-shift feature of ZigBee chip sequences to move the error-prone chips. We compare TwinBee with our work in two aspects. First, TwinBee doesn't reduce the emulation errors of a ZigBee symbol, but recovers the chip sequence by combining two ZigBee symbols. In comparison, WIDE can directly minimize the emulation errors of a ZigBee symbol by digital emulation. Two emulated ZigBee symbols based on WIDE can also be further combined like TwinBee to further improve communication reliability. Second, the premise of TwinBee is to obtain the 32 bits of chip data in the physical layer to recover ZigBee symbol. However, most commercial ZigBee devices can only get the decoded ZigBee symbol and don't support getting the decoded raw chip sequence in physical layer. Hence, TwinBee is difficult to run on commercial devices.

V. Evaluation

In this section, we conduct extensive experiments to evaluate the performance of WIDE. We compare WIDE with WEBee,

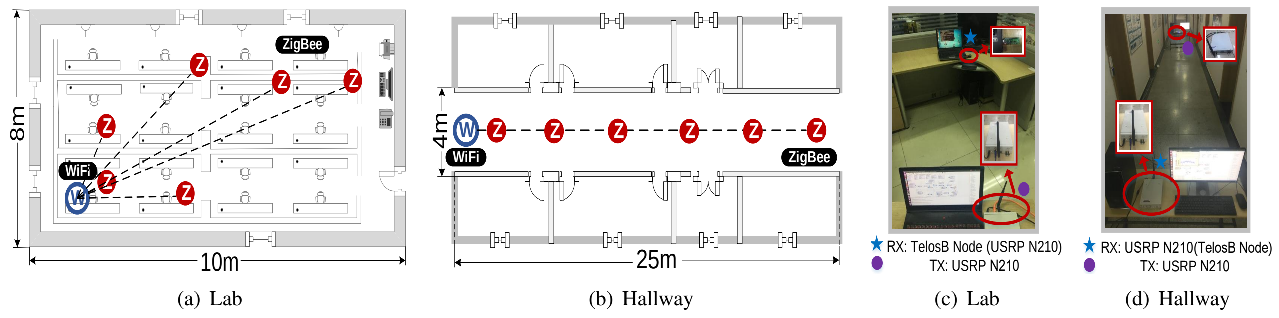


Fig. 16. Experiment settings in the lab and the hallway

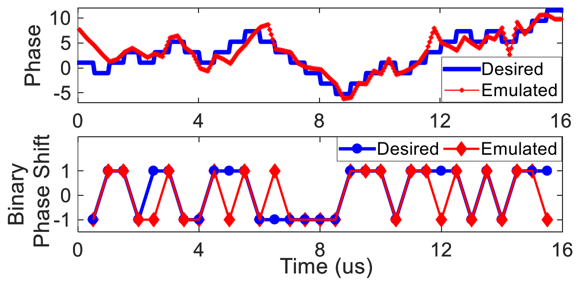


Fig. 17. Emulated phase sequence

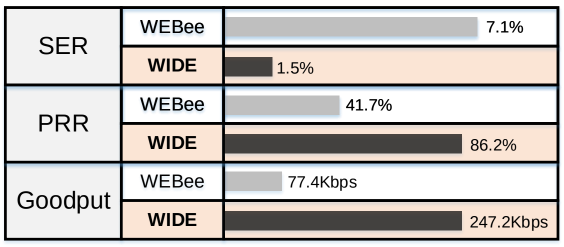
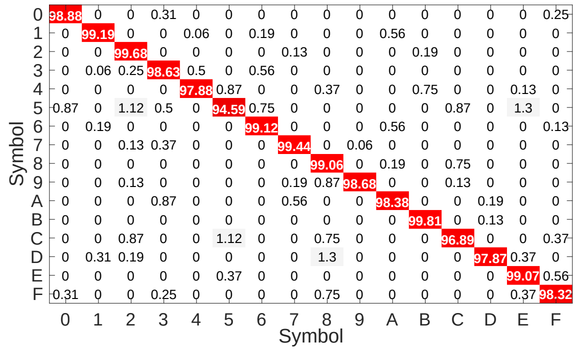


Fig. 18. Overall performance comparison



Symbol	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	98.88	0	0	0.31	0	0	0	0	0	0	0	0	0	0	0	0.25
1	0	99.19	0	0	0.06	0	0.19	0	0	0	0.56	0	0	0	0	0
2	0	0	99.68	0	0	0	0	0.13	0	0	0	0.19	0	0	0	0
3	0	0.06	0.25	98.63	0.5	0	0.56	0	0	0	0	0	0	0	0	0
4	0	0	0	0	97.88	0.87	0	0	0.37	0	0	0.75	0	0	0.13	0
5	0.87	0	1.12	0.5	0	94.59	0.75	0	0	0	0	0	0.87	0	1.3	0
6	0	0.19	0	0	0	0	99.12	0	0	0	0.56	0	0	0	0	0.13
7	0	0	0.13	0.37	0	0	0	99.44	0	0.06	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	99.06	0	0.19	0	0.75	0	0	0
9	0	0	0.13	0	0	0	0	0.19	0.87	98.68	0	0	0.13	0	0	0
A	0	0	0	0.87	0	0	0	0.56	0	0	98.38	0	0	0.19	0	0
B	0	0	0	0	0	0	0	0	0	0	0	99.81	0	0.13	0	0
C	0	0	0.87	0	0	1.12	0	0	0.75	0	0	0	96.89	0	0	0.37
D	0	0.31	0.19	0	0	0	0	0	1.3	0	0	0	0	97.87	0.37	0
E	0	0	0	0	0	0.37	0	0	0	0	0	0	0	0	99.07	0.56
F	0.31	0	0	0.25	0	0	0	0	0.75	0	0	0	0	0	0.37	98.32

Fig. 19. Decoding accuracy for different symbols

the representative physical-level CTC from WiFi to ZigBee. WIDE can be implemented directly with commodity devices. The USRP N210 devices are used only for evaluation purpose to measure low-level PHY information, such as Hamming distance and symbol error rate (SER).

A. Experiment Setup

The WIDE transmitter is a USRP N210 device with 802.11 b/g PHY. The WIDE receiver is a USRP N210 device with 802.15.4 PHY. During experiments,each emulated ZigBee packet consists of four bytes of preamble (0x00000000), a byte of start of frame delimiter (SFD) (0xA7), two bytes of packet length, variable bytes payload. For fair comparison, the composition of WEBee packet is the same as that of WIDE. We set the ZigBee channel at 19 and set the central frequency of the WiFi channel at 2440MHz, which can be realized by many commodity WiFi devices (e.g. Atheros AR9485, AR5112, and AR2425). Our evaluation include symbol error rate (SER), packet reception ratio (PRR), and goodput. To ensure statistical validity, we obtain the average result of 10 experiments, each of which sends 1,000 WIDE packets under a wide range of settings including indoor/hallway, short/long distance, and mobile scenarios.

B. Emulated Phase Sequence

First, we observe the desired phase sequence and emulated phase sequence to verify the feasibility of digital emulation. As shown in Fig. 17, we can find that the emulated phase sequence resembles the desired phase sequence with limited distortions. The decoded binary phase shift of the emulated phase sequence only has four wrong bits (except the first and the last bits). The limited number of error bits can be tolerated by the mechanism of ZigBee DSSS decoding. This emulated phase sequence can be decoded successfully. Therefore, the digital emulation method is feasible for the physical-level CTC.

C. Overall Performance Comparison

We conduct experiments to compare the overall performance of WIDE and WEBee in practice. The distance between the WiFi sender and the ZigBee receiver is 4m. The ZigBee payload is 16 bytes and includes all 16 different symbols. The experiments are conducted in our lab as shown in Fig. 16(c), with a consistent ambient environment and a similar network interference condition.

The comparison results are shown in Fig. 18. First, the SER of WEBee is 7.1% and the SER of WIDE is 1.5%. The reason is that the WiFi emulation result of the ladder shaped waveform is better than the half sine waveform, which reduces

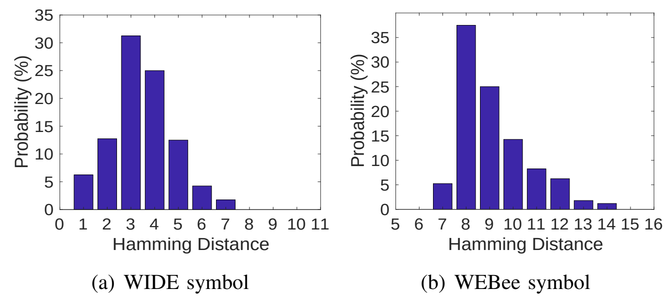


Fig. 20. Hamming Distance

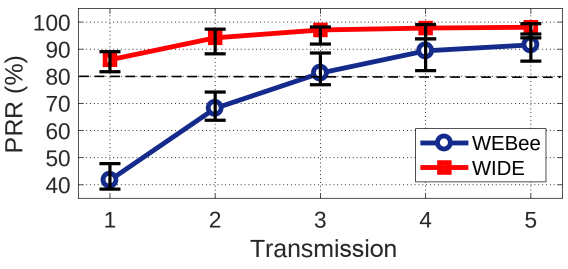


Fig. 21. PRR with the times of transmission

the symbol decoding errors. We further evaluate the decoding accuracy for all different ZigBee symbols and the evaluation results are shown in Fig. 19. The average decoding accuracy of different symbols varies from 94.59% to 99.81%. Because the phase sequence of each ZigBee symbol is different, the emulation result of WiFi is also different. Furthermore, Fig. 20(a) shows the Hamming Distance between the decoded binary phase shift sequence and the predefined binary phase shift sequence when we adopt WEBee and WIDE. For all ZigBee symbols, the Hamming Distance of WIDE is much lower than WEBee. The commercial ZigBee device sets a threshold to tolerate a certain number of chip errors, which by default is 12. This threshold is relaxed to 20 in WEBee, while WIDE has no need to modify this threshold.

At the same time, the PRR of WIDE can be up to 86.1%, while the PRR of WEBee is 41.7%. This is also because that the better emulation result improves the possibility of preamble detection and header synchronization for the ZigBee signals. In addition, the PRR of both WEBee and WIDE is also related to the number of packet transmissions, a parameter to trade off between throughput and reliability. Fig. 21 illustrates the PRR under different transmission numbers. The PRR of WIDE exceeds 80% when the CTC packets are sent at a rate of 250Kbps (only 1 transmission), while WEBee achieves the 80% PRR when the CTC packets are sent at a rate of 83.3Kbps (3 transmissions).

WIDE and WEBee are both the physical-level CTC methods, so the theoretical throughput of WIDE and WEBee can both be the ceiling speed of standard ZigBee communication. Due to the different SER and PRR of ZigBee decoding, the goodputs of WEBee and WIDE are 77.4Kbps and 247.2Kbps, respectively. It is worth noting that the performance of WEBee realized by our evaluation is worse than [23], this is because we don't adopt repeated preamble protection and link coding.

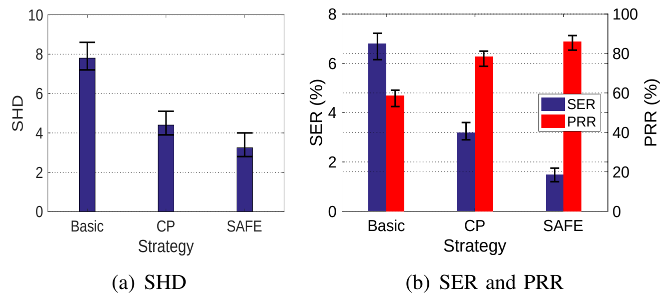


Fig. 22. WIDE performance with different optimization methods

Whereas, it doesn't affect the comparison of WIDE and WEBee when we conduct all the experiments under the same settings.

D. WIDE Performance with Different Optimization Methods

In order to reduce the WiFi emulation errors, WIDE applies CP alleviation and SAFE algorithm in the Phase Sequence Optimization. In this subsection, we conduct experiments to evaluate the performance of SER and PRR improvement with these two optimization methods. The evaluation results are shown in Fig. 22(a) and Fig. 22(b). We can find that average Hamming distance between all the desired ZigBee symbols and the emulated ZigBee symbols decreases with the optimization methods. For example, the average Hamming distance decreases from 7.8 to 4.4 with the CP error alleviation. Further, the average Hamming distance decreases to 3.25 by using the SAFE algorithm. SER decreases from 6.8% to 3.2% and 1.5% with the CP error alleviation and the SAFE algorithm. Similarly, PRR increases from 58.6% to 78.4% and 86.2% with the CP error alleviation and the SAFE algorithm. Therefore, our optimization methods are effective to reduce the WiFi emulation errors and improve the reliability of CTC.

E. WIDE Performance under Different Settings

In this subsection, we vary the ZigBee payload length, the distance between the sender and the receiver, and operating environments to study their impacts on WIDE in terms of SER and PRR. We also evaluate WIDE in mobile scenarios.

1) Impact of ZigBee payload length: We study the impact of payload length on WIDE and WEBee. We change the payload length of ZigBee from 8 bytes to 24 bytes. The distance between the WiFi sender and the ZigBee receiver is 4m. Fig. 23(a) and Fig. 23(b) show the evaluation results of SER and PRR respectively. We can find that the SER increases with the increase of payload length since that the longer payload brings more accumulated errors. When the payload length is 8 bytes, the SER of WIDE and WEBee is 1.21% and 5.41% respectively. When the payload length increases to 24 bytes, the SER of WIDE and WEBee increases to 2.17% and 9.14% respectively. Whereas, the SER of WIDE is still much lower than that of WEBee.

The PRR of ZigBee packets relies on the preamble detection, header synchronization, and the CRC result. So a single symbol error may lead to a packet loss. When a packet has

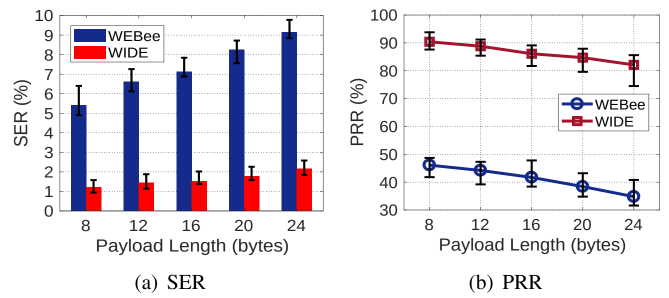


Fig. 23. WIDE performance with different ZigBee payload lengths

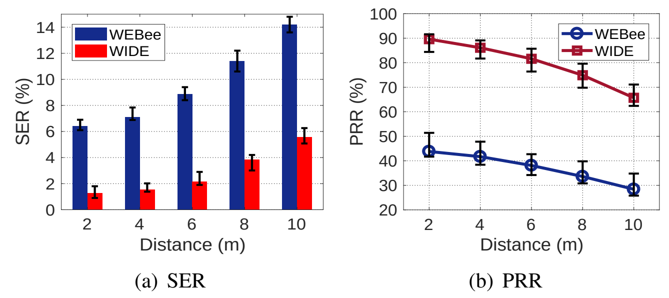


Fig. 24. WIDE performance with different distances

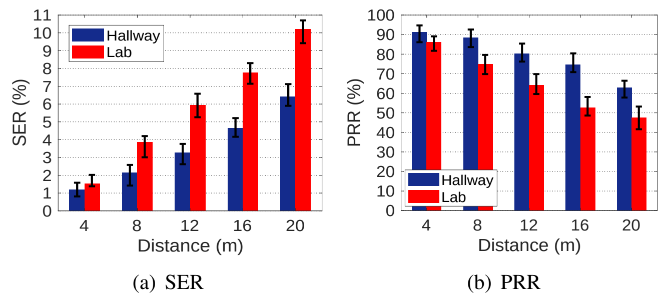


Fig. 25. WIDE performance in hallway with different distances

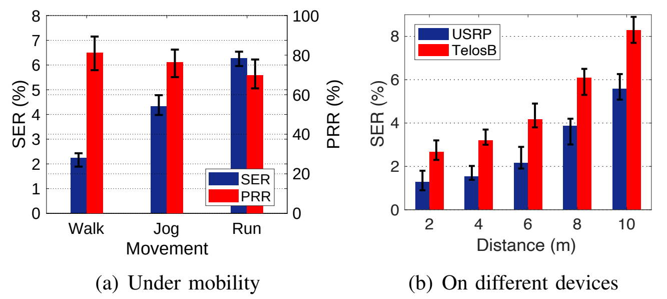


Fig. 26. WIDE performance under mobility and different devices

variable length payload, the longer payload it has, the easier to lose packets. Fig. 23(b) shows the PRR with the variation of payload length. We can find that the PRR of WEBee decreases sharply with the increase of payload length. When the payload length is 24 bytes, the PRR of WEBee decreases to 34.1%. Whereas, the PRR of WIDE only decreases to 82.1%.

The results reveal that the payload length has an influence on the performance of WIDE and WEBee, but WIDE can still achieve relatively reliable performance when increasing the payload length.

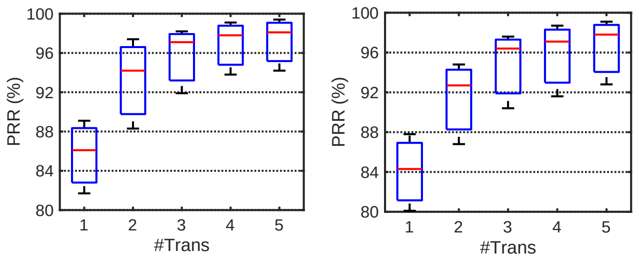
2) Impact of distance: We then study the impact of distance between the WiFi sender and the ZigBee receiver. We conduct this experiment in the lab and vary the distance from 1m to 10m, as shown in Fig.16(a). The ZigBee payload length is 16 bytes.

Fig. 24(a) shows the SER with the variation of distance. We can find that the SER increases with the increase of distance. When the distance is 2m, the SER of WIDE and WEBee is 1.28% and 6.41% respectively. When the distance increases to 10m, the SER of WIDE and WEBee increases to 5.58% and 15.24% respectively. Because the longer the distance, the lower the SNR. This results in the worse attenuation of the signal amplitude and the phase distortion. Whereas, the SER of WIDE is still more stable and lower than that of WEBee, which is due to the good emulation for the ladder shaped waveform.

The PRR with the variation of distance is shown in Fig. 24(b). We can find that the PRR of WEBee decreases sharply with the increase of distance. When the distance is 10m, the PRR of WEBee decreases to 28.4%. The PRR of WIDE decreases to 65.7%. With the increase of distance, the success rate of ZigBee preamble detection and header synchronization will decrease. In addition, the increase of the SER also makes CRC easier to fail.

The results reveal that the distance has an influence on the performance of WIDE and WEBee, but WIDE can still achieve relatively reliable performance when increasing the distance. In addition, we want to clarify that WIDE is not restricted in use at short range. Indeed, WIDE doesn't affect the communication distance of the WiFi device and the ZigBee device. The limited communication range in the experiments is just because of the limited space of the real lab, where we carried out the experiments.

3) Impact of environment: We evaluate WIDE in different environments. We conduct the experiments to compare the SER and PRR of WIDE in the lab and the hallway as shown in Fig. 16(c) and Fig. 16(d). Fig. 25(a) and Fig. 25(b) present the SER and PRR of WIDE in two environments. We can find that the SER of WIDE in the hallway is lower than the SER of WIDE in the lab. This is because the environment in the lab is more complicated than that in the hallway, which leads to more serious multipath influences on the received signals and results in higher decoding errors. We can also find that the SER increases with the increase of distance. For example, when the distance is 10m, the SER of WIDE in the lab and hallway is 10.24% and 6.42% respectively. In addition, the PRR of WIDE in the hallway is higher than the PRR of WIDE in the lab. This is because the environment of hallway is cleaner and the SNR is higher, which improves the possibility of preamble detection and header synchronization for the ZigBee signals. The PRR decreases with the increase of distance. For example, when the distance is 20m, the PRR of WIDE in the lab and hallway is 47.4% and 62.8% respectively.



(a) PRR of ZigBee channel 19 (b) PRR of ZigBee channel 17

Fig. 27. WIDE performance of parallel communication

F. Impact of mobility

In the experiments, the WIDE sender transmits packets to emulate ZigBee signals. The ZigBee payload length is 16 bytes. A volunteer carrying the ZigBee receiver walks, jogs, and runs at a speed of 1 m/s, 2 m/s, and 4 m/s, respectively. Fig. 26(a) shows the SER and PRR of WIDE with varying speeds. The SER increases and the PRR decreases with the increase of the speed. When the moving speed is 4m/s, the SER and the PRR of WIDE is 6.28% and 68.9%, which is still acceptable.

G. Evaluation on the commercial ZigBee device.

In this subsection, we evaluate the performance of WIDE on the commodity device. A TelosB node CC2530 [34] is used as the WIDE receiver. The WiFi sender transmits emulated packets with the payload length of 16 bytes. The distance between the WiFi sender and the TelosB node varies from 2m to 10m. We measure the SER of the emulated ZigBee symbols as shown in Fig. 26(b). When the distance increases from 2m to 10m, the SER of WIDE on TelosB node increases from 2.6% to 8.2%. We find the performance of WIDE on TelosB node is slightly worse than that on USRP due to the difference of receiving sensitivity between USRP and commercial TelosB node.

H. Parallel communication

We also conduct experiments to evaluate the performance of WIDE for parallel communication. We set the frequency of a WiFi channel at 2440MHz, the two regions of WiFi subcarriers [-13 to -19] and [13 to 19] are utilized to achieve two parallel CTC with standard ZigBee channel 17 and channel 19. The distance between the WIDE sender and the ZigBee receiver is 4m and the payload length of the ZigBee is 16 bytes. Fig. 27(a) and Fig. 27(b) compare the performance of WIDE between two different ZigBee channels when packet retransmission is utilized. The PRR of WIDE increases with the increase of retransmissions under different parallel ZigBee channels. At the same time, the performance of WIDE at ZigBee channel 17 is slightly worse than the performance at ZigBee channel 19 overall. The reason for this phenomenon is the diversity of the channel quality under different channels.

VI. Conclusion

In this paper, we propose WIDE, a physical-level CTC from WiFi to ZigBee via digital emulation. Instead of emulating the standard ZigBee half sine waveform, WIDE selects an appropriate non-standard waveform to emulate ZigBee binary phase shift sequence, which is the essence of ZigBee decoding. Compared with analog emulation, digital emulation can adjust the phase sequence at the same time of satisfying the requirement of the binary phase shift sequence, which offers flexibility to achieve a more reliable CTC. We conduct extensive experiments to evaluate the performance of WIDE. The results show that WIDE significantly improves the Packet Reception Ratio (PRR) from 41.7% to 86.2%, which is 2× of WEBee's, an existing representative physical-level CTC. To the best of our knowledge, WIDE is the first work that leverages digital emulation to achieve physical-level CTC. Without loss of generality, the method of digital emulation is generic and applicable to a set of CTCs, where the transmitter has a wider bandwidth for emulation and the receiver decoding is based on the phase shift.

Acknowledgment

This work is supported in part by National Key R&D Program of China No. 2017YFB1003000, National Science Fund of China under grant No. 61772306, and the R&D Project of Key Core Technology and Generic Technology in Shanxi Province (2020XXX007).

References

[1] X. Zheng, Z. Cao, J. Wang, Y. He, and Y. Liu, "Zisense: towards interference resilient duty cycling in wireless sensor networks," in *Procs. of ACM SenSys*, 2014.

[2] L. Shangguan, Z. Yang, A. X. Liu, Z. Zhou, T. Gu, and Y. Liu, "Stpp: Spatial-temporal phase profiling-based method for relative rfid tag localization," *IEEE/ACM Transactions on Networking*, vol. 25, no. 1, pp. 596–609, 2017.

[3] R. K. Sheshadri, K. Sundaresan, E. Chai, A. Khojastepour, S. Rangarajan, and D. Koutsonikolas, "Blu: Blue-printing interference for robust lte access in unlicensed spectrum," in *Procs. of ACM CoNEXT*, 2017.

[4] Y. Yan, P. Yang, X. Li, T. Yue, L. Zhang, and L. You, "Zimo: building cross-technology mimo to harmonize zigbee smog with wifi flash without intervention," in *Procs. of ACM MobiCom*, 2013.

[5] Y. Li, Z. Chi, X. Liu, and T. Zhu, "Passivezigbee: Enabling zigbee transmissions using wifi," in *Procs. of ACM Sensys*, 2018.

[6] Y. Chae, S. Wang, and K. Song Min, "Exploiting wifi guard band for safeguarded zigbee," in *Procs. of ACM Sensys*, 2018.

[7] K. Sundaresan, S. V. Krishnamurthy, X. Zhang, A. Khojastepour, and S. Rangarajan, "Trinity: A practical transmitter cooperation framework to handle heterogeneous user profiles in wireless networks," in *Procs. of ACM MobiHoc*, 2015.

[8] Z. An, L. Yang, and Q. Lin, "Cross-frequency communication: Near-field identification of uhf rfids with wifi," in *Procs. of ACM MobiCom*, 2018.

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

[10] G. Chen and W. Dong, "Jamcloak: Reactive jamming attack over cross-technology communication links," in *Procs. of IEEE ICNP*, 2018.

[11] Y. Li, Z. Chi, X. Liu, and T. Zhu, "Chiron: Concurrent high throughput communication for iot devices," in *Procs. of ACM MobiSys*, 2018.

[12] Z. Yin, Z. Li, K. Song Min, and T. He, "Explicit channel coordination via cross-technology communication," in *Procs. of ACM MobiSys*, 2018.

[13] H. Iqbal, M. H. Alizai, I. A. Qazi, O. Landsiedel, and Z. A. Uzmi, "Scylla: interleaving multiple iot stacks on a single radio," in *Procs. of ACM CoNext*, 2018.

[14] D. Liu, Z. Cao, J. Wang, Y. He, and Y. Liu, "Duplicate detectable opportunistic forwarding in duty-cycled wireless sensor networks," in *Procs. of IEEE/ACM ToN*, 2016.
[15] W. Jiang, K. Song Min, Z. Li, and T. He, "Achieving receiver-side cross-technology communication with cross-decoding," in *Procs. of ACM MobiCom*, 2018.
[16] Y. Zhang and Q. Li, "Howies: A holistic approach to zigbee assisted wifi energy savings in mobile devices," in *Procs. of IEEE INFOCOM*, 2013.
[17] C. Kameswari and D. Ashutosh, "Esense: communication through energy sensing," in *Procs. of ACM MobiCom*, 2009.
[18] X. Guo, X. Zheng, and Y. He, "Wizig: Cross-technology energy communication over a noisy channel," in *Procs. of IEEE INFOCOM*, 2017.
[19] X. Zheng, Y. He, and X. Guo, "Stripcomm: Interference-resilient cross-technology communication in coexisting environments," in *Procs. of IEEE INFOCOM*, 2018.
[20] Z. Yin, W. Jiang, K. Song Min, and T. He, "C-morse: Cross-technology communication with transparent morse coding," in *Procs. of IEEE INFOCOM*, 2017.
[21] K. Song Min and T. He, "Freebee: Cross-technology communication via free side-channel," in *Procs. of ACM MobiCom*, 2015.
[22] W. Jiang, Z. Yin, K. Song Min, and T. He, "Transparent cross-technology communication over data traffic," in *Procs. of IEEE INFOCOM*, 2017.
[23] Z. Li and T. He, "Webee: Physical-layer cross-technology communication via emulation," in *Procs. of ACM MobiCom*, 2017.
[24] Z. Li, Y. Li, W. Jiang, and T. He, "Bluebee: Physical-layer cross-technology communication via emulation," in *Procs. of ACM SenSys*, 2017.
[25] Z. Chi, Y. Li, H. Sun, Y. Yao, Z. Lu, and T. Zhu, "B2w2: N-way concurrent communication for iot devices," in *Procs. of ACM SenSys*, 2016.
[26] X. Guo, Y. He, X. Zheng, L. Yu, and O. Gnawali, "Zigfi: Harnessing channel state information for cross-technology communication," in *Procs. of IEEE INFOCOM*, 2018.
[27] 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 *Procs. of IEEE INFOCOM*, 2017.
[28] S. Wang, Z. Yin, Z. Li, and T. He, "Networking support for physical-layer cross-technology communication," in *Procs. of IEEE ICNP*, 2018.
[29] Y. Chen, Z. Li, and T. He, "Twinbee: Relable physical-layer cross-technology communication with symbol-level coding," in *Procs. of IEEE INFOCOM*, 2018.
[30] Z. Li and T. He, "Longbee: Enabling long-range cross-technology communication," in *Procs. of IEEE INFOCOM*, 2018.
[31] X. Guo, Y. He, X. Zheng, Z. Yu, and Y. Liu, "Lego-fi: Transmitter-transparent ctc with cross-demapping," in *Procs. of IEEE INFOCOM*, 2019.
[32] Z. Chi, Y. Li, Y. Yao, and T. Zhu, "Pmc: Parallel multi-protocol communication to heterogeneous iot radios within a single wifi channel," in *Procs. of IEEE ICNP*, 2017.
[33] "Zigbee." https://standards.ieee.org/standard/802_15_4-2020.html.
[34] "Cc2530." https://www.ti.com.cn/product/cn/CC2530.
[35] "Mc1321." https://www.datasheetarchive.com/MC1321-datasheet.html.
[36] "At86rf230." https://www.alldatasheet.com/datasheet-pdf/pdf/313502/ATMEL/AT86RF230.html.
[37] "nrf24e1." https://www.keil.com/dd/docs/datashts/nordic/nrf24e1.pdf.

Yuan He is an associate professor in the School of Software and BNRist of Tsinghua University. He received his B.E. degree in the University of Science and Technology of China, his M.E. degree in the Institute of Software, Chinese Academy of Sciences, and his PhD degree in Hong Kong University of Science and Technology. His research interests include wireless networks, Internet of Things, pervasive and mobile computing. He is a member of the IEEE and ACM.

Xiuzhen Guo received the B.E. degree in the School of Electronic and Information Engineering from Southwest University in 2016. She is currently a PhD student in Tsinghua University. Her research interests include Internet of Things, wireless networks, and ubiquitous computing.

Jia Zhang received the B.E. degree in the School of Software from Tsinghua University in 2019. He is currently a PhD student in Tsinghua University. His research interests include wireless sensor networks and cross-technology communication.

Haotian Jiang received the B.E. degree in the School of Software from Tsinghua University in 2019. He is currently a master student in Tsinghua University. His research interests include wireless network co-existence and cross-technology communication.