c-Chirp: Towards Symmetric Cross-technology Communication over Asymmetric Channels

Dan Xia, Xiaolong Zheng, Liang Liu, Chaoyu Wang, Huadong Ma
Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia
Beijing University of Posts and Telecommunications, Beijing, China
{xiadan, zhengxiaolong, liangliu, matthew, mhd}@bupt.edu.cn

***Abstract*—Cross-Technology Communication (CTC) is an emerging technique that enables direct interconnection among incompatible wireless technologies. However, CTC channels established by existing methods are inherently asymmetric because of either the one-way nature of emulation in physical-level CTC or the asymmetric communication range caused by the asymmetric transmission power. In this paper, we focus on establishing symmetric CTC over asymmetric CTC channels. The bottleneck is the short communication range from the low-power and narrow-band technology to the high-power and wide-band technology because the asymmetric bandwidth and transmission power lead to serious symbol distortions. To compensate the inevitable distortions, we take advantage of the channel asymmetry and construct chirps in WiFi Channel State Information (CSI) to enhance the patterns used for conveying data. In this way, we can extend the communication range from ZigBee to WiFi. We theoretically build the model of CSI chirp based CTC and design *c-Chirp*, a novel CTC from ZigBee to WiFi. Due to channel asymmetry and discreteness, the WiFi receiver can only observe partial and distorted CSI chirps. To cope with this issue, we design a matching based chirp decoding method as well as an adaptation algorithm to reliably decode the symbols. We conduct extensive experiments to evaluate *c-Chirp*. The results show that *c-Chirp* can achieve a $60m$ communication range from ZigBee to WiFi, which is $6\times$ longer than ZigFi, an existing representative CTC from ZigBee to WiFi.**

I. Introduction

Internet of Things (IoT) envisions ubiquitous connections among all "things". However, the rich diversity of wireless technologies raises challenges of interconnecting heterogeneous devices. The emerging Cross-Technology Communication (CTC) technique is proposed to enable the direct communication between the incompatible radios without any extra hardware. Existing CTC methods can be categorized into packet-level CTC and physical-level CTC. Packet-level CTC establishes mutually accessible side channels by different packet patterns in terms of transmission timing [1]–[4], signal strength [5]–[7], and channel state [8]–[10]. Instead of using side channels, physical-level CTC directly intrudes into the channel of another technology by emulating other technology's signal [11] or generating recognizable signal patterns [12] at the sender or using distinguishable results when processing the standard signal from another technology [13] by the heterogeneous receiver.

Though promising, most of the existing CTC methods fail to achieve bi-directional symmetric communication due to the inherent asymmetry of CTC channels. The physical-level CTC channel usually is one-way because the emulation and desired signal patterns highly depend on the targeted receiving technology. The channels established by packet-level CTC are built on top of legacy packets, which can be bi-directional. However, the asymmetric Tx power leads to extremely asymmetric communication ranges. The WiFi Tx power can be $20dBm$. But the maximum ZigBee Tx power is $0dBm$. Hence, transmission patterns desired by CTC are easier to generate and detect from WiFi to ZigBee, but much more unreliable from ZigBee to WiFi. According to the results reported in the literature, the communication range of packet-level CTC from ZigBee to WiFi is only a few meters, which is much shorter than the range from WiFi to ZigBee, which can be dozens of meters. The asymmetric communication limits CTC works within a very limited range when demanding data exchange in both directions.

In this paper, we focus on establishing symmetric CTC over the asymmetric legacy channels to achieve similar communication ranges in both directions. A straightforward idea is combining two state-of-the-art methods that have the longest communication range in each direction. However, as analyzed in detail in Section III, such a naive combination is infeasible. Combining physical-level CTC with other CTC solutions is hard in practice. Even though physical-level CTC methods do not change the radio hardware, they do more or less require changes in standard radio configurations. Combining two packet-level CTC methods is feasible because manipulating packet transmissions doesn't cause any conflict. But unfortunately, the CTC communication range between WiFi and ZigBee is asymmetric due to CTC channel asymmetry. As shown in Fig. 1, according to the reported results in the literature, the longest communication range of packet-level CTC can be near 50 meters from WiFi to ZigBee [14] but only 10 meters from ZigBee to WiFi [9].

From the analysis above, we can find that the short communication range from ZigBee to WiFi is the bottleneck to achieve symmetric CTC. If we have a CTC method from ZigBee to WiFi that has a communication range similar to the state-of-the-art CTC from WiFi to ZigBee, then combining these two methods can achieve symmetric bi-directional CTC. To extend the communication range from ZigBee to WiFi, we are inspired by Chirp Spread Spectrum (CSS) [15] that possesses a high sensitivity. Instead of transmitting in a single

978-1-7281-6630-8/20/$31.00 © 2020 IEEE

ZigBee channel, we utilize multiple ZigBee channels to construct chirps in WiFi Channel State Information (CSI) to get an enhanced CTC coding pattern. In this way, we can improve the sensitivity and therefore extend the communication range.

However, it is non-trivial to achieve CSI chirp based CTC from ZigBee to WiFi. First, the theoretical model of CSI chirp based CTC channel is unexplored. How to enable the CSI chirp with stable features is unknown. Second, due to bandwidth asymmetry, WiFi with a bandwidth of $20MHz$ can only observe part of the CSI chirp that ZigBee constructs on 16 channels, spreading the whole $80MHz$ band. Besides, different from CSS's continuous frequency changing, CSI chirp is discrete because of the discontinuity of ZigBee channels, which is prone to distortions. How to decode the distorted CSI chirps with only partial information is challenging. Third, even though CSI chirp extends the communication range, it also lowers the throughput because of the extended symbol length. Hence, adjusting CSI chirps to extend communication range with minimum throughput degradation is necessary yet challenging, especially when considering channel dynamics.

By addressing these issues, we propose *c-Chirp*, a novel CSI chirp based CTC method from ZigBee to WiFi. By continuously switching channels and sequentially influencing different WiFi subcarriers, the ZigBee sender constructs CSI chirps to increase the sensitivity and enlarge the communication range from ZigBee to WiFi. Chirps with different starting channel indexes are encoded as different symbols. Then the WiFi receiver collects CSI sequences and decodes symbols by inferring the starting channel. The main contribution of this work is summarized as follows.

- To the best of our knowledge, we are the first work towards establishing symmetric CTC over asymmetric CTC channels. We propose a new CTC channel that encodes data by constructing chirps in WiFi CSI. The channel has higher sensitivity and therefore extends the CTC range from ZigBee to WiFi. We also establish the theoretical model of CSI chirp based CTC channel.
- We devise *c-Chirp* that solves practical technical challenges to achieve CSI chirp based CTC. *c-Chirp* uses a dynamic chirp decoding method to reliably decode with only partial and distorted CSI chirps. An adaptation algorithm is also elaborated to cope with the channel dynamics.
- We implement a prototype of *c-Chirp* with commercial WiFi devices and ZigBee motes. We extensively evaluate the performance of *c-Chirp*. The results show that *c-Chirp* can achieve a communication range of $60m$, which is $6\times$ longer than existing CTC from ZigBee to WiFi and comparable to the range of CTC from WiFi to ZigBee.

The rest of this paper is organized as follows. We discuss the related work in Section II and analyze existing methods in detail to motivate our work in Section III. We then introduce the *c-Chirp* design in Section V. We present the evaluation of *c-Chirp* in Section VI and conclude our work in Section VII.

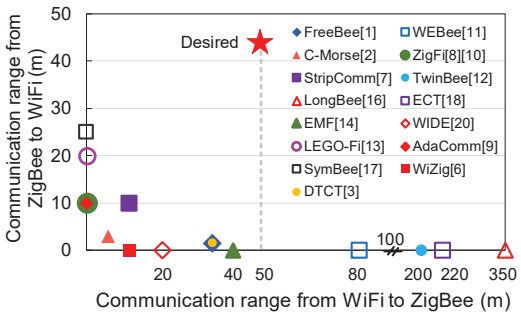


Fig. 1. The communication ranges between ZigBee and WiFi reported in the existing literatures.

II. Related Work

Physical-level CTC emulates the simple or similar signals directly to generate a heterogeneous receiver compliant packet [11], [12], [16]–[19]. WEBee [11] is the first work that emulates the ZigBee signal by specific WiFi payloads. WIDE [20] proposes digital emulation that emulates the binary phase shift sequence of ZigBee's signal but not the exact signal shapes to improve reliability. LongBee [16] concentrates the Tx energy into a narrow band by down-clocking the standard operations of WiFi, to extend the communication range from WiFi to ZigBee. LEGO-Fi [13] reuses WiFi radio modules but in a customized sequence to process the ZigBee signal and then decodes by mapping the characteristic processing results to the ZigBee symbols. Existing physical-level CTC methods are one-way because neither the emulation signal or the desired physical-layer patterns targeted on one technology works for another technology.

Packet-level CTC manipulates packet transmissions such as timing [1]–[3], energy [5]–[7], packet length [21], and channel state variations [8]–[10] to establish mutually accessible side channels. The authors in [22] propose an energy-based CTC channel that encodes symbol 1/0 by the presence/absence of packets. WiZig [6] further uses multiple energy levels to encode multiple bits at the same time. ZigFi [8], [10] leverages the influence of ZigBee transmissions on WiFi CSI to encode symbols from ZigBee to WiFi. AdaComm [9] further proposes a learning-based decoding method to cope with channel dynamics on a single channel. The communication ranges of ZigFi and AdaComm are very limited because they only consider the impacts of one ZigBee channel, which is not stable when the ZigBee sender is far from the WiFi receiver.

Existing CTC usually focuses on enabling one-way communication between incompatible technologies. Establishing symmetric CTC is very important for a practical communication system but has not received enough attention. How to achieve symmetric bi-directional CTC over the inherent asymmetric CTC channels is still an open problem.

III. Motivation

In this section, we first study the limitations of existing CTC methods to achieve symmetric CTC. Then we investigate the reasons why existing methods fail to achieve a satisfying communication range from ZigBee to WiFi.

A. Limitations of the State-of-the-arts

Bi-directional symmetric communication is essential to using CTC for interconnecting heterogeneous IoT devices. However, existing CTC methods either fail to provide comparable communication ranges satisfied in both directions or require high-complexity radio reconfigurations when altering the CTC direction, which is infeasible in practice.

We summarize the reported communication ranges of recent CTC works between ZigBee and WiFi in Fig. 1. From the results, we can observe a gap between the communication ranges in two directions. Using an existing CTC method that works in both directions cannot provide satisfying performance. For example, FreeBee can enable near $40m$ communication from WiFi to ZigBee but only several meters from ZigBee to WiFi.

Combining CTC that has the longest communication range in each direction is a straightforward idea but hard to accomplish by existing CTC methods. Combining physical-level CTC with other CTC solutions is hard in practice. Physical-level CTC methods usually require changes in the standard radio configuration. When altering the communication direction, we have to simultaneously reconfigure the hardware of both the sender and receiver according to the requirements of CTC in the reverse direction, which is too complicated to accomplish in practice. Combing two packet-level CTC methods is feasible but not able to provide satisfying performance. From Fig. 1, we can find that the longest communication range of packet-level CTC from WiFi to ZigBee is near $50m$ [14], while the longest communication range of packet-level CTC is only $10m$ in ZigFi. Then combining ZigFi with other methods can only provide symmetric communication in the $10m$ range, which is much shorter than the achievable range from WiFi to ZigBee.

From the above analysis, we find that the short communication range of CTC from ZigBee to WiFi is the bottleneck to achieve symmetric CTC by combining two packet-level CTC methods. In the following, we study the reason why ZigFi, the packet-level CTC that reports the longest range from ZigBee to WiFi, fails to achieve a satisfying communication range.

B. Reasons for the Short Achievable Range

We conduct experiments to measure the performance of ZigFi, a state-of-the-art packet-level CTC from ZigBee to WiFi, in a hall. We set the distance between the ZigBee sender and WiFi receiver to $50m$, which is comparable to the communication range of packet-level CTC from WiFi to ZigBee. The ZigBee sender transmits ZigFi symbol 0 for $0.5s$ in four different channels (ZigBee channel $21 \sim 24$) in turn. The Tx power of ZigBee is set to $0dBm$. The WiFi receiver operates in WiFi channel 11 and collects CSI by CSITool [23] with a sampling rate of $2KHz$. The presented CSI sequence is the average of four different WiFi subchannels overlapped with corresponding ZigBee channels.

Fig. 2 shows the results during 15 seconds. Surprisingly, even though the distance between the ZigBee sender and WiFi receiver is $50m$ long, all the channels can occasionally detect the influences of ZigBee transmissions. This observation

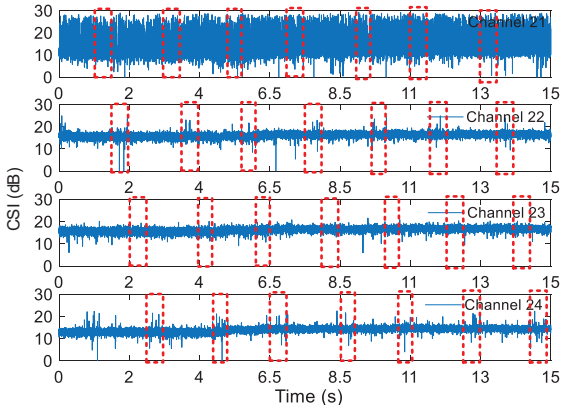


Fig. 2. The CSI in subcarriers overlapped with the ZigBee channel. The red rectangles represent that ZigBee transmits during this time slot.

reveals that ZigBee transmissions still influence WiFi CSI even out of ZigFi's communication range. The reason behind that ZigFi cannot work at the distance where ZigBee still influences WiFi CSI is that the influence is not stable enough to decode. First, the influence is frequency selective. The influence can be too weak to decode such as on channel 23 or the channel is too noisy such as channel 21. Second, the influence is also time varying such as channel 22 and 24. The unstable influence on a single channel causes unstable encoding patterns and therefore a too large symbol error rate to use in practice. Hence, ZigFi is limited to a short range to provide reliable enough communication.

The results in Fig. 2 reveal that the influence of ZigBee on CSI is still effective but very unstable when the range is long. Hence, to extend the communication range, the key is how to construct a stable and distinguishable pattern to encode symbols on top of the unstable CSI influence. Inspired by CSS modulation in LoRa, we try to improve the decoding sensitivity by constructing chirps in the WiFi CSI matrix. We take advantage of the channel asymmetry between WiFi and ZigBee and use transmissions on multiple ZigBee channels to construct the CSI chirp in Fig. 2. However, achieving the CSI chirp based CTC from ZigBee to WiFi still needs elaborate designs to solve several practical challenges.

IV. CSI CHIRP BASED CTC CHANNEL MODEL

In this section, we derive the theoretical channel model of CSI chirp-based CTC to demonstrate that CSI chirps can enhance sensitivity.

Signal through a WiFi channel is expressed as $Y = HX + N$, where X is the transmitted signal, and N is Additive White Gaussian Noise (AWGN). $H = [H(f_1), H(f_2), \ldots, H(f_L)]$ is the CSI vector that describes the properties of L subcarriers. The WiFi receiver can estimate H using pre-defined signal X_w and the corresponding received signal Y_w. $\hat{H}_w$, the estimation of H with minimum mean square errors (MMSE) is:

$$\hat{H}_w = \frac{Y_w X_w^H}{(\sigma_n^2 I + X_w X_w^H)} \tag{1}$$

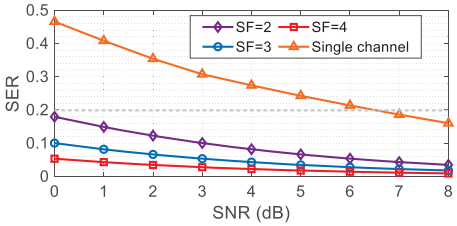


Fig. 3. The theoretical relationship between SER and SNR.

where σ_n^2 is the variance of noise N, I is the identity matrix, and X_w^H is the Hermitian transposition of X_w. $X_w = \sqrt{P_w}S_w$, where P_w is the transmission power and the entries of S_w are random variables with zero-mean and unit variance [24]. Denote the ZigBee signal as $Z = [Z(f_1), Z(f_2), \ldots, Z(f_L)]$, the entries of which are circularly symmetric complex Gaussian random variables with the mean a and the variance σ_z^2. If ZigBee transmissions influence a WiFi link, then the received signal is $Y_w = H_z X_w + N + Z$ and the MMSE estimate of $\hat{H}$ is denoted as $\hat{H}_z$. Then CTC decodes current symbol by calculating the CSI variations $|\triangle H| = |\hat{H} - \hat{H}_w| = [|\hat{H}(f_1) - \hat{H}_w(f_1)|, \ldots, |\hat{H}(f_L) - \hat{H}_w(f_L)|]$. From Eq. (1), we can calculate the probability density function of $|\triangle H(f_l)|$ as follows.

$$f(x) = \begin{cases} (\frac{x}{\sigma_1^2})exp[\frac{-(x^2+A^2)}{2\sigma_1^2}]I_0(\frac{xA}{\sigma_1^2}),\ x \geq 0 & , \quad \hat{H} = \hat{H}_z \\ (\frac{x}{\sigma_0^2})exp[\frac{-x^2}{2\sigma_0^2}],\ x \geq 0 & , \quad \hat{H} = \hat{H}_w \end{cases} \tag{2}$$

where $\sigma_0^2 = E[|\triangle H(f_l))|^2]/2 = 2(\frac{P_w}{P_w+N_0})$ is the variance of $f(x)$ when there is only WiFi, $\sigma_1^2 = E[|\triangle H(f_l))|^2]/2 = \frac{P_w}{N_0+P_w+\sigma_z^2} + \frac{P_w}{N_0+P_w} + A^2$ is the variance when there is ZigBee, where $A = |E(\triangle H(f_l))| = \frac{a\sqrt{P_w}}{N_0+P_w+\sigma_z^2}$ is the mean of $\triangle H(f_l)$. I_0 is the modified 0-th order Bessel function.

1) Single Channel: Existing methods such as ZigFi use a single ZigBee channel to encode symbol "1" and "0" as the presence and absence of ZigBee packets, respectively. To decode, we use a threshold b to judge whether ZigBee is causing large CSI variations. We decode the CTC symbol as "1" if $|\triangle H(f_l)| > b$, or "0" otherwise. Assume that "1" and "0" are sent with the same probability, then the Symbol Error Rate (SER) P_e^s can be calculated as follows.

$$\begin{aligned} P_e^s &= 0.5(P(0|1) + P(1|0)) \\ &= 0.5(\int_{-\infty}^{b} f_1(x)\,dx + \int_{b}^{+\infty} f_0(x)\,dx) \\ &= 0.5(1 - Q_M(\tfrac{A}{\sigma_1}, \tfrac{b}{\sigma_1}) + exp(-\tfrac{b^2}{2\sigma_0^2})) \end{aligned} \tag{3}$$

where $Q_M(a,b) = \int_b^{+\infty} x\ exp(-\frac{a^2+x^2}{2})I_0(ax)\,dx$ is Marcum Q-function. Since we leverage the CSI variation to convey data, the Signal-to-Noise Ratio (SNR) γ should be redefined as the ratio of CSI variation with ZigBee (signal strength) to the variation without ZigBee (noise strength). Namely, $\gamma = \sigma_1^2/\sigma_0^2$.

The threshold b is chosen as $(\sigma_0^2 + \sigma_1^2)/2$. Since $|\triangle H(f_l)|$ follows a Rician distribution, the relationship between SER and SNR can be calculated as

$$\begin{aligned} P_e^s = \ & 0.5(1 - \sqrt{\tfrac{1}{\sqrt{2K}}(1+\tfrac{1}{\gamma})}Q(\tfrac{1}{2}(1+\tfrac{1}{\gamma} - \sqrt{2K})) \\ & + exp(-\tfrac{1}{4}(1+\gamma)^2)),\ \gamma = \frac{\sigma_1^2}{\sigma_0^2} \end{aligned} \tag{4}$$

where $K = A^2/2\sigma_1^2$ is the Rician factor.

2) CSI Chirp: We leverage multiple ZigBee channels to construct a CSI chirp. Denote $\mathbb{M} = \{m_1, ..., m_i, ..., m_M\}$ as the symbol set. Symbol m_i corresponds to a CSI chirp $c_{m_i} \in \mathbb{C}$, where $\mathbb{C}$ is the set of CSI chirps with different starting channel indexes. A CSI chirp $c_{m_i} = \{c_{m_i}[0], c_{m_i}[1], ..., c_{m_i}[2^{SF}-1]\}$ is a sequence of 2^{SF} CSI variations. SF is the spreading factor. When receiving a CSI chirp $c_{\hat{m}} = \{c_{\hat{m}}[0], c_{\hat{m}}[1], \ldots, c_{\hat{m}}[2^{SF}-1]\}$, the receiver decodes by mapping $c_{\hat{m}}$ to a symbol $\hat{m}$ by the following rule.

$$\hat{m} = \arg\min_{m_i} \Sigma_{j=0}^{2^{SF}-1}(c_{\hat{m}}[j] - c_{m_i}[j])^2 \tag{5}$$

where $(c_{\hat{m}}[j] - c_{m_i}[j])^2$ is the square of the Euclidean distance between $c_{\hat{m}}[j]$ and $c_{m_i}[j]$ in the CSI matrix collected during a CTC symbol window.

Since $|\triangle H(f_l)|$ follows a Rician distribution, P_e [24] can also be expressed as $P_e = \int_0^{\infty} f(\gamma)P_\gamma(\gamma)\,d\gamma$, where $P_\gamma(\gamma)$ is the probability of symbol error in AWGN channel. We transform Eq. (2) by replacing the amplitude x with the SNR γ into the formation $f(\gamma)$. $P_\gamma(\gamma)$ and $f(\gamma)$ are expressed as follows.

$$\begin{aligned} P_\gamma(\gamma) \quad & \approx S_{d_{min}}Q(\sqrt{\tfrac{d_{min}^2}{2\sigma_0^2}}) = S_{d_{min}}Q(\sqrt{8 \times SF \times \gamma}) \\ f(\gamma) \quad & = \tfrac{(1+K)e^{-K}}{\bar{\gamma}}exp[-\tfrac{(1+K)\gamma}{\bar{\gamma}}]I_0(2\sqrt{\tfrac{K(1+K)\gamma}{\bar{\gamma}}}) \end{aligned} \tag{6}$$

where d_{min} is the minimum distance among symbols, $S_{d_{min}}$ is the number of symbols with minimum distance d_{min} and $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^{\infty} e^{-\frac{u^2}{2}}\,du$. $K = \frac{A^2}{2\sigma_1^2}$ is the Rician factor and $\bar{\gamma}$ is the average SNR.

We plot the theoretical SER of CSI chirps with different SFs and the method using a single channel in Fig. 3. We can find that using CSI chirps can significantly reduce the SNR required for decoding to obtain the same SER. For example, when we require SER is lower than 0.2, then CSI chirps with all SFs theoretically can work even when the SNR is as low as $1dB$. However, the required SNR of using a single channel must be larger than $6dB$. With a lower required SNR for decoding, CSI chirps can bear larger path loss and therefore have a larger communication range. Even though the theoretical results shed the light on using CSI chirps to extend the CTC communication range, applying CSI chirps in practice still needs elaborate designs to solve the challenges such as chirp distortions caused by bandwidth asymmetry and channel discontinuity.

V. System Design

The preliminary studies have demonstrated CSI chirps can lower the required SNR for decoding and therefore have the potential to extend the communication range from ZigBee to WiFi. In this section, we present the designs to achieve CSI chirp based CTC with commercial WiFi and ZigBee devices.

A. System Overview

Fig. 4 shows the framework of *c-Chirp*. Without modifying the WiFi or ZigBee physical layer, a CTC channel of *c-Chirp* is built on the existing WiFi link. The ZigBee sender first

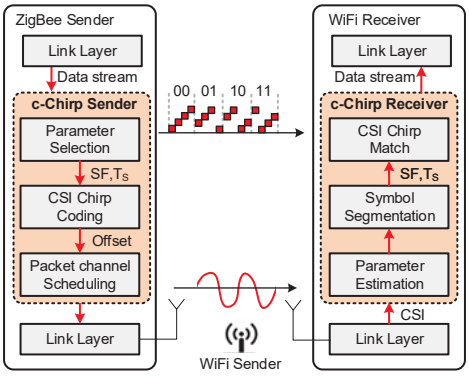


Fig. 4. The framework of *c-Chirp*.

selects the parameter such as SF based on the CTC channel quality and then encodes the data by different chirps with different indexes of the starting channel. Then the ZigBee sender will generate the channel scheduling to transmit packets in channels according to the encoding order. The WiFi receiver will extract the CSI from received packets to monitor whether there is ZigBee. Once detecting, the *c-Chirp* receiver will estimate the parameter and synchronize by the preamble defined as a number of CSI up-chirps. Then based on the symbol window length learned by synchronization, the *c-Chirp* receiver segments the CSI sequences into frames that correspond to symbols. Due to the channel asymmetry, the WiFi receiver can only observe partial information if ZigBee encodes on the whole $80MHz$ band. Besides, due to the discontinuity of ZigBee channels, the CSI chirps are prone to distortions. To settle these issues, we propose a matching based decoding algorithm that can reliably decode distorted CSI chirp with only partial information.

B. CSI Chirp Coding

We keep the CSI chirp coding simply because the coding is on the ZigBee node that has limited resources. The chirp coding process is similar to CSS. *c-Chirp* improves the sensitivity and enhances the encoded CSI patterns by CSI chirps. *c-Chirp* lets the ZigBee sender transmit in multiple ZigBee channels in turn to construct CSI chirps. As shown in Fig. 5, using K ZigBee channels C_0 to C_{K-1} can construct K different chirps to encode K symbols. A symbol consists of K chips that each chip corresponds to the CSI influence on one channel. When transmitting a symbol, the sender broadcasts ZigBee packets in all K channels in turn but starts from different initial channels to encode different symbols. For example, when transmitting symbol i, *c-Chirp* starts the chirp from channel C_i and finishes the broadcasting on channel $C_{(i-1+K) \bmod K}$.

We will introduce the parameter selection in Section V-D together with the parameter adaptation. Denote the chip window length as T_c. After deciding the spreading factor SF, we can get the symbol window length should be $T_s = 2^{SF} \cdot T_c$. Then a schedule of ZigBee transmissions is determined according to the encoded symbols. In our current design, similar to ZigFi, to reliably obtain the CSI samples influenced by ZigBee transmissions, we set T_c as $5ms$.

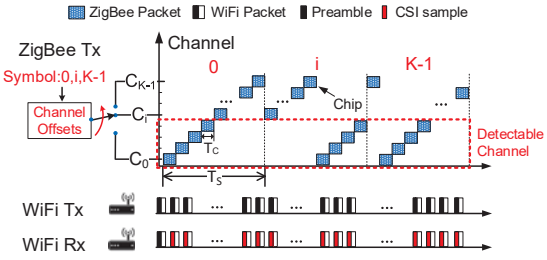


Fig. 5. Illustration of c-Chirp coding.

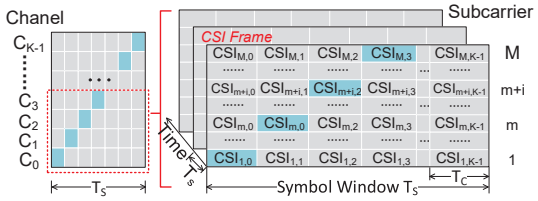


Fig. 6. CSI segmentation.

C. CSI Chirp Decoding

Though resilient to frequency selective and time-varying influences, a CSI chirp is discontinuous in the subcarriers of CSI because the ZigBee channels are discontinuous. Hence, it is prone to distortions. To tackle this problem, we devise a novel decoding mechanism to recover the distorted CSI chirp by exploiting the linear feature of CSI chirp.

1) Parameter Estimation: To avoid the packet exchange overhead, we use the preamble in each data packet to estimate parameters SF and T_s, instead of explicit coordination between the sender and receiver. The preamble is a series of CSI upchirps. Since SF can be calculated by T_s, we first estimate T_s and then calculate SF. A WiFi receiver can observe four chips corresponding to four overlapped ZigBee channels. Then we calculate the correlation between the preamble templates and the received CSI sequences. If the correlation is larger than a pre-defined threshold, a *c-Chirp* packet is detected. Then we use autocorrelation on the preamble to estimate T_s.

A WiFi channel contains 64 subcarriers. The bandwidth of each subcarrier is $312.5KHz$. When receiving a packet, the WiFi receiver can calculate the CSI values of all subcarriers and obtain a CSI vector. According to SF and T_s obtained from the parameter estimation component, we can segment the CSI sequences into CSI frames that each corresponds to a CSI chirp, as shown in Fig. 6. A blue part represents that ZigBee transmits in this slot. Each symbol window contains $M \times K$ CSI sequences, where M is the number of subcarriers. In each symbol window, we can obtain a matrix $\mathbb{CSI}_{M \times K}$, where $CSI_{m,k}$ is the CSI amplitude sequence of subcarrier m collected within the k-th chip window.

Since we focus on the variation in each chip to judge whether there is ZigBee influence, we further extract the CSI variation information in terms of the amplitude difference from the segmented CSI matrixes. We first calculate the mean CSI amplitude of each subcarrier by all the collected CSI samples in a symbol window. Then we calculate the difference between the amplitude of each CSI sample and the mean CSI

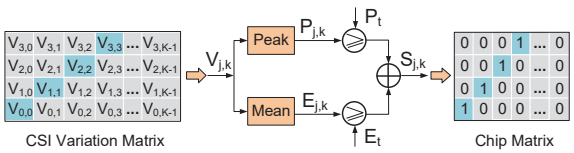


Fig. 7. Mapping the CSI differences into chips.

amplitude. Hence, in each chip, we have a CSI difference sequence $\triangle CSI_{m,k}$. Since a ZigBee channel C_j overlaps c subcarriers from m_1 to m_c, then we group the CSI differences of correlated subcarriers in a chip window according to the following equation.

$$V_{j,k} = \sum_{i=m_1}^{m_c} \alpha_i \triangle CSI_{i,k} \tag{7}$$

where α_i is the weight of subcarrier m_i. Similarly, we can get the grouped CSI variations of the other three ZigBee channels overlapped with the WiFi channel, and obtain the extracted CSI variation matrix $\mathbb{V}_{4\times K}$.

2) CSI Chirp Match: The segmentation component provides CSI difference matrix, $\mathbb{S}_{4\times K}$, to describe the CSI amplitude changes. Similar to ZigFi, we also use the peak and mean of the CSI difference to judge whether there is ZigBee influence during a chip window. What's different is *c-Chirp* uses multiple chips to overcome the instability of ZigBee's influence in a single channel. As Fig. 7 shown, given $\mathbb{V}_{4\times K}$, we can calculate the peak $P_{j,k}$ and mean $E_{j,k}$ of the CSI difference sequence $V_{j,k}$ in each chip window, and obtain the peak matrix $\mathbb{P}$ and mean matrix $\mathbb{E}$. Then we can use these two features, $P_{j,k}$ and $E_{j,k}$, to whether the chip $S_{j,k}$ is "1" or "0". We decide $S_{j,k}$ to 1 if $E_{j,k}$ is larger than the threshold E_t and $P_{j,k}$ is larger than the threshold P_t, and set $S_{j,k}$ to 0 otherwise. Then we can obtain a chip matrix $\mathbb{S}_{4\times K}$.

Due to the channel asymmetry, WiFi with a channel of $20MHz$ can only observe a partial CSI chirp when ZigBee constructs on 16 channels, spreading the whole $80MHz$ band. Then the starting channel observed by a WiFi *c-Chirp* receiver can be different from the one that ZigBee starts. For example, Symbol $K-1$ in Fig. 5, the starting channel is C_{K-1} but the WiFi receiver detects it as channel C_0, leading to decoding errors. Notice that besides the channel index, the time of the observed channel is influenced by ZigBee in the symbol window also gives the information about the starting channel. Therefore, to reliably decode symbols with only partial and distorted information, we define a *CSI chirp template* to identify the channel offset (τ) and the initial channel (C_I). The channel offset indicates the offset of starting channel from the initial channel, which is the channel index to encode symbols. The CSI chirp template G with $SF = \log_2 K$ is defined as

$$\begin{aligned} G(C_I,\tau,K) &= F_{4\times K}((C_I+\tau)\ mod\ K,\ k), k = 0,...,K-1 \\ F_{4\times K}(i,j) &= \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \end{aligned} \tag{8}$$

Since the CSI chirp is discontinuous, *c-Chirp* uses Dynamic Hamming Distance (DHD) to quantify the distance between the template $G(C_I,\tau,K)$ and extracted chip matrix $\mathbb{S}$, which

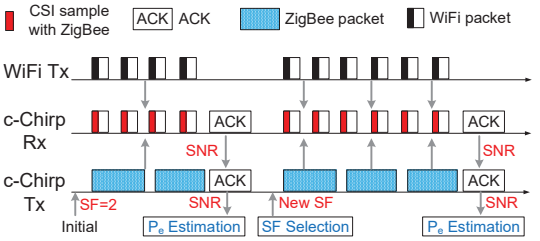


Fig. 8. Illustration of the adaptation process.

is calculated as $DHD_\tau = \sum_{i=0}^{K-1} D_i(\tau)$, where $D_i(\tau)$ is the hamming distance of chip i between the template $G(C_I,\tau,K)$ and the received symbol. To decode the received symbol, we adopt the following function to estimate τ and C_I.

$$(\tau, C_I) = \arg \min_{(\tau,C_I)=(0,11)}^{(K-1,26)} DHD_\tau \tag{9}$$

after learning (τ, C_I), *c-Chirp* can obtain the mapped symbol.

D. Adaptation Mechanism

Even though the major design goal of *c-Chirp* is extending the communication range with satisfying reliability. Practical applications desire a high throughput along with the satisfying reliability. Note that given the fixed chip window length T_c, a high SF can reduce the SER but lead to throughput decrease because of the extended symbol window length. Hence, we propose an adaption mechanism to maximize the goodput with the required SER. The optimization goal obj of adaptation is

$$\begin{aligned} obj = &\quad \frac{1}{2^{SF}T_c} \cdot (1-P_e) \\ s.t.: &\quad P_e \leq \tilde{P}_e \end{aligned} \tag{10}$$

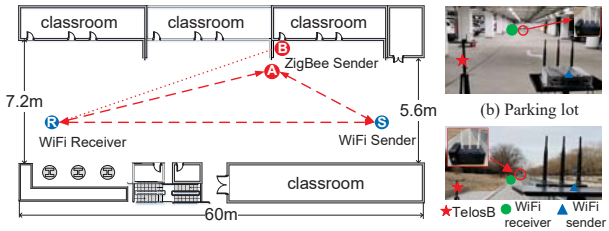
where $\tilde{P}_e$ is the required SER.

The procedure of adaptation is shown in Fig. 8. Initially, *c-Chirp* set $SF = 2$. The *c-Chirp* receiver keeps estimating the SNR of the CSI chirp channel by measuring the ratio of the CSI variance when there is ZigBee to the CSI variance when there is no ZigBee. Then the SNR is piggyback in the ACK message and sent back to the *c-Chirp* sender. The *c-Chirp* sender estimates the SER based on current SNR, according to the channel model we propose in Section IV. The *c-Chirp* receiver will select the smallest SF that can satisfy the SER requirement to maximize the goodput. Then the *c-Chirp* sender will use the new SF for the next CTC packet transmission. The receiver can estimate the adopted SF by the method we introduced in the previous subsection.

VI. Evaluation

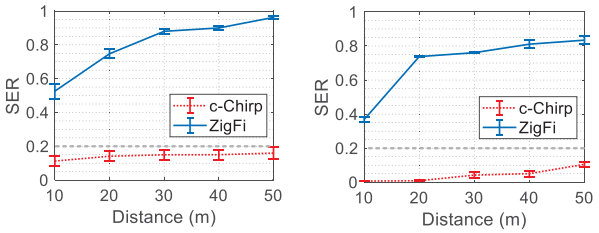
A. Experiment Setting

We implement *c-Chirp* on the off-the-shelf devices. We use TelosB, a commercial ZigBee platform to implement *c-Chirp* sender and a commercial WiFi computer equipped with Intel 5300 NIC as the *c-Chirp* receiver. CSITool [23] is installed in the *c-Chirp* receiver to collect CSI with a sampling frequency of $2KHz$. Unless otherwise specified, WiFi is set to channel 11 and the Tx power of ZigBee is set to $0dBm$. We use ZigFi for comparison. We conduct extensive experiments



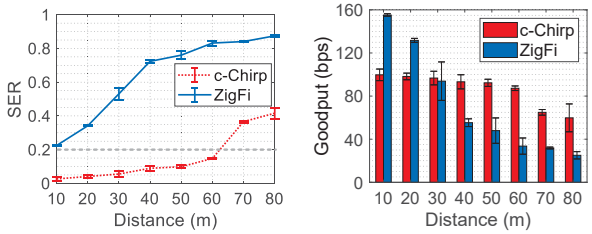
(a) The floor plan of the hall (b) Parking lot (c) Outdoor

Fig. 9. Experiment settings in three environments.



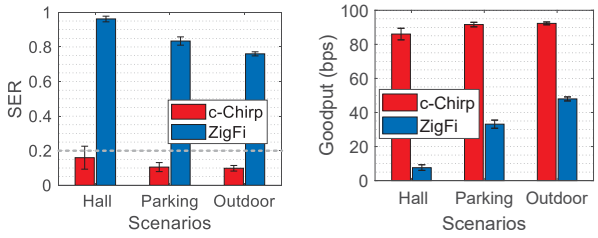
(a) SER in the indoor hall (b) SER in the indoor parking lot

Fig. 10. SER of c-Chirp in the indoor environments.



(a) SER (b) Goodput

Fig. 11. Performance of c-Chirp in the outdoor environment.



(a) SER (b) Goodput

Fig. 12. Performance of c-Chirp in different scenarios.

under wide settings including different scenarios, distance and the transmission power of *c-Chirp* sender.

B. Overall Performance Comparison

We first compare the performance of *c-Chirp* with ZigFi in terms of the communication range. We study the SER of *c-Chirp* and ZigFi when varying the distance between the ZigBee sender and WiFi receiver. We first conduct experiments in the indoor environments, including the hall (Fig. 9(a)) and the parking lot (Fig. 9(b)). The positions of the ZigBee sender and WiFi sender are 5m apart, which is not changed during the experiments. Then we vary the position of the WiFi receiver to get different communication distances. Due to the limited indoor space, the distance is varied from 10m to 50m.

The results are shown in Fig. 10. As expected, SER increases with the increase of distance. But we can clearly find *c-Chirp* can achieve a much lower SER, compared to ZigFi. When the distance increases from 10m to 50m, the SER of ZigFi increases from 0.526 to 0.962, while the SER of *c-Chirp* increases only 0.047, from 0.113 to 0.160. Similar results are observed in the parking lot. When varying the distance from 10m to 50m in the parking lot, the SER of *c-Chirp* and ZigFi increase from 0.007 and 0.368 to 0.105 and 0.834, respectively. Both methods in the parking lot have better performance because the interference is less than the hall of our teaching building.

To evaluate the performance in longer range, we conduct experiments in the outdoor environment shown in Fig. 9(c). The settings are same to the indoor environments. The experimental results are shown in Fig. 11. From Fig. 11(a), we can find that the SER of *c-Chirp* is 0.148 when the distance is 60m, where ZigFi has a too high SER to work. Actually, if the required SER is lower than 0.2, ZigFi cannot work well when the distance is larger than 10m. We also measure the goodput of *c-Chirp* and ZigFi and present the results in Fig. 11(b). We can find that the goodput of *c-Chirp* is lower than ZigFi when the distance is shorter than 20m because *c-Chirp* sacrifices the throughput for reliability. But when the distance is larger than 30m, the goodput of *c-Chirp* is becoming better than ZigFi. This is because ZigFi has too many symbol errors caused by the unstable influence in a single channel.

Our motivation is to achieve a 50m communication range which is comparable to the longest range from WiFi to ZigBee. Hence, in Fig. 12, we present the experimental results when the distance between the sender and receiver is 50m in three environments. We can find that the SER of *c-Chirp* is lower than the required 0.2 SER and its performance in both SER and goodput is much better than ZigFi. The results demonstrate that *c-Chirp* can achieve a long enough communication range from ZigBee to WiFi for building the symmetric CTC.

We also conduct experiments in the hall to show that simple retransmission cannot help ZigFi to obtain a satisfying communication range. The distance between *c-Chirp* sender and receiver is 40m. We plot the SER with different numbers of retransmissions in Fig. 13. We can find that even retransmitting 10 times can only reduce the SER to 0.671, which is still too high to use in practice. We also find that retransmissions help *c-Chirp* achieve better reliability. The SER of *c-Chirp* can decrease by 49.17% when retransmitting 3 times. Hence, if a lower SER is desired, extending the current implementation of *c-Chirp* with retransmission mechanism will work.

C. Performance of c-Chirp's Components

1) Parameter Estimation: We use the preamble to estimate *SF* and synchronize for the following decoding. Hence, the accuracy of parameter estimation is highly related to the overall performance. We vary the length of preamble and study the estimation accuracy in the hall environment where the

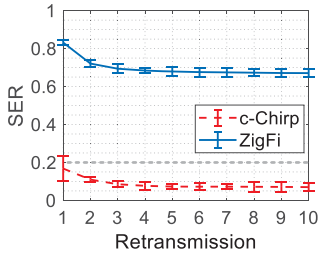


Fig. 13. SER with retransmissions.

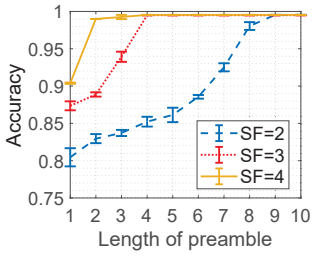


Fig. 14. Parameter estimation.

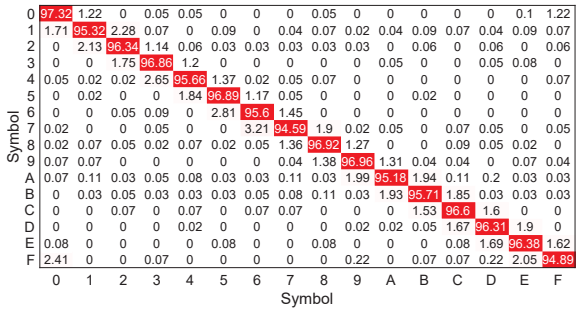
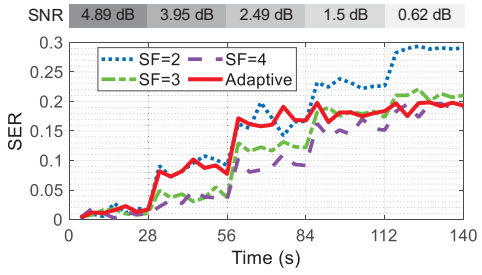
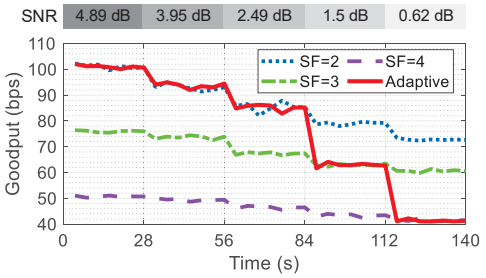


Fig. 15. Decoding accuracy for different symbols.



(a) SER



(b) Goodput

Fig. 16. Performance of the adaptation mechanism.

c-Chirp sender and receiver are 40m apart. The results are shown in Fig. 14. When increasing the length of preamble, the estimation accuracy increase because the CSI variation information can be observed. We also find that the convergence rate of estimation accuracy increases with the increase of SF. The accuracy converges to the accuracy larger than 0.98 when the length increases to 2, 4, and 8 for $SF = 2, 3, 4$ respectively. Hence, in our current implementation, we use different preamble lengths for different SF. Given the fixed chip window length, using different preamble lengths just lead to the same time duration of the preamble. Hence, the *c-Chirp* receiver can just calculate the correlation of the received CSI sequences with a fixed length, reducing the complexity.

2) Decoding Accuracy: We study the decoding accuracy for different symbols when $SF = 4$, to study whether there is bias on the decoding accuracy. The experiment is conducted in the parking lot and the distance between sender and receiver is 40m. Each symbol is transmitted for 3000 times. The results are shown in Fig. 15. The average decoding accuracy of different symbols varies from 0.9459 to 0.9732. There is no specific symbol having obviously lower accuracy than other symbols. The decoding errors usually occur among the nearby symbols because we use the channel offset to encode symbols.

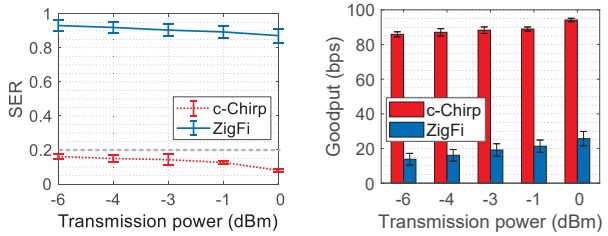
3) Adaptation Mechanism: To evaluate the performance of adaption mechanism, we study the performance of *c-Chirp* with fixed SF and adaptive SF. We conduct the experiments in the parking lot and vary the transmission power from 0dBm to $-10dBm$ to obtain different SNR. The distance between *c-Chirp* sender and receiver is 20m. The required SER is 0.2. The SER and goodput during the 140s experiment are shown in Fig. 16. We can find that for all the methods, with the decrease of SNR, the SER increases and the goodput decreases. During $[0, 84s]$, *c-Chirp* with any SF can achieve an SER lower than 0.2. *c-Chirp* with adaptation mechanism will select the smallest SF for a better goodput. During $[84, 140s]$, SNR is not good enough to provide a satisfied SER. Then *c-Chirp* with adaptation mechanism will increase SF accordingly to obtain the SER satisfied the requirement. The results demonstrate that the adaption mechanism can efficiently adjust the parameters based on the quality of the CSI chirp channel.

D. Impact of Transmission Power

We rely on the influence of ZigBee transmissions on CSI to convey data. Hence, we study the performance of *c-Chirp* when using different ZigBee transmission power. We conduct the experiments in the parking lot and set the distance between the *c-Chirp* sender and receiver to 40m. We vary ZigBee transmission power from 0dBm to $-6dBm$. Fig. 17(a) and Fig. 17(b) presents the results in terms of SER and goodput. We can find the transmission power indeed has an influence on the performance but within an acceptable range. When the transmission power decreases from 0dBm to $-6dBm$, the SER increases from 0.081 to 0.161 and the goodput decreases from 90.12bps to 85.87bps. The results show that at 40m distance, even with a relatively low transmission power, *c-Chirp* still has a satisfying performance while ZigFi has a too high SER to work in practice.

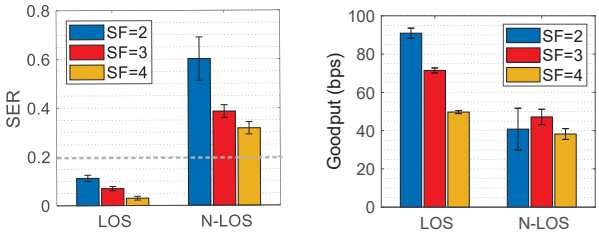
E. Impact of Non-Line-of-Sight

We also evaluate the performance of *c-Chirp* in Line-of-Sight (LOS) and Non-LOS (NLOS) scenarios. We conduct the experiments in the hall and deploy the *c-Chirp* sender at positions A and B as shown in Fig. 9(c), to obtain the LOS and NLOS (through a wall) links. The distance between *c-Chirp* sender and receiver is 55m. Fig. 18(a) presents the SER of *c-Chirp* in two scenarios. As expected, the SER in LOS scenario is lower than SER in NLOS scenario. When



(a) SER (b) Goodput

Fig. 17. Performance with different transmission power.



(a) SER (b) Goodput

Fig. 18. Performance in LOS and NLOS scenarios.

$SF = 4$, the SER of *c-Chirp* in LOS and NLOS is 0.0301 and 0.3169 respectively. This is because NLOS propagation of WiFi signals experience more serious multipath influence on CSI, leading to a lower SNR at the same distance. To improve the reliability in NLOS, we can leverage the retransmission scheme as shown by the evaluation results in Section VI-B. Fig. 18(b) presents the goodput of *c-Chirp* in two scenarios. When $SF = 4$, the goodput of *c-Chirp* in LOS and NLOS is 49.69bps and 38.23bps respectively. The goodput also experiences performance degradation due to multipath influence.

VII. Conclusion

Towards symmetric CTC over the inherent asymmetric CTC channels, we propose *c-Chirp*, a novel CSI chirp based CTC method that enlarges the communication range from ZigBee to WiFi. We propose a new CTC channel and theoretically model the channel to guide practical designs. By continuously switching channels and sequentially influencing different WiFi subcarriers, *c-Chirp* constructs stable CSI chirps to encode symbols on the unstable CSI influences. We design a dynamic chirp decoding method to reliably decode with only partial and distorted CSI chirps. We also propose an adaptation mechanism to maximum the goodput with a satisfying SER in the dynamic environments. We implement *c-Chirp* on commercial WiFi and ZigBee devices. We conduct extensive experiments to evaluate the performance of *c-Chirp* in various settings. The results show that *c-Chirp* can extend the communication range from ZigBee to WiFi to 60m, which is 6$\times$ longer than the existing CTC from ZigBee to WiFi and comparable to the range of existing CTC from WiFi to ZigBee.

Acknowledgment

This work is supported in part by NSFC under No. 61722201, the CCF-Tencent Open Fund under Grant IAGR20190115, the Funds for Creative Research Groups of China under No. 61921003, the Fundamental Research Funds for the Central Universities No. 2019RC40, and the 111 Project (B18008).

References

[1] S. M. Kim and T. He, "Freebee: Cross-technology communication via free side-channel," in *Proceedings of ACM MobiCom*, 2015.

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

[3] W. Jiang, Z. Yin, S. M. Kim, and T. He, "Transparent cross-technology communication over data traffic," in *Proceedings of IEEE INFOCOM*, 2017.

[4] X. Zhang and K. G. Shin, "Gap sense: Lightweight coordination of heterogeneous wireless devices," in *Proceedings of IEEE INFOCOM*, 2013.

[5] K. Chebrolu and A. Dhekne, "Esense: Communication through energy sensing," in *Proceedings of ACM MobiCom*, 2009.

[6] X. Guo, X. Zheng, and Y. He, "Wizig: Cross-technology energy communication over a noisy channel," in *Proceedings of IEEE INFOCOM*, 2017.

[7] X. Zheng, Y. He, and X. Guo, "Stripcomm: Interference-resilient cross-technology communication in coexisting environments," in *Proceedings of IEEE INFOCOM*, 2018.

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

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

[10] X. Guo, Y. He, X. Zheng, L. Yu, and O. Gnawali, "Zigfi: Harnessing channel state information for cross-technology communication," *IEEE/ACM Transactions on Networking*, vol. 28, no. 1, pp. 301–311, 2020.

[11] Z. Li and T. He, "Webee: Physical-layer cross-technology communication via emulation," in *Proceedings of ACM MobiCom*, 2017.

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

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

[14] 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 *Proceedings of IEEE INFOCOM*, 2017.

[15] IEEE Computer Society, "IEEE Stansard 802.15.4a," 2007, https://standards.ieee.org/standard/802_15_4a-2007.html.

[16] Z. Li and T. He, "Longbee: Enabling long-range cross-technology communication," in *Proceedings of IEEE INFOCOM*, 2018.

[17] S. Wang, S. M. Kim, and T. He, "Symbol-level cross-technology communication via payload encoding," in *Proceedings of IEEE ICDCS*, 2018.

[18] W. Wang, T. Xie, X. Liu, and T. Zhu, "Ect: Exploiting cross-technology concurrent transmission for reducing packet delivery delay in iot networks," in *Proceedings of IEEE INFOCOM*, 2018.

[19] J. Yao, X. Zheng, J. Xu, and K. Wu, "Cross-technology communication through symbol-level energy modulation for commercial wireless networks," in *Proceedings of IEEE PerCom*, 2020.

[20] X. Guo, Y. He, J. Zhang, and H. Jiang, "Wide: physical-level ctc via digital emulation," in *Proceedings of ACM/IEEE IPSN*, 2019.

[21] Y. Zhang and Q. Li, "Howies: A holistic approach to zigbee assisted wifi energy savings in mobile devices," in *Proceedings of IEEE INFOCOM*, 2013.

[22] S. Yin, Q. Li, and O. Gnawali, "Interconnecting wifi devices with ieee 802.15. 4 devices without using a gateway," in *Proceedings of IEEE DCOSS*, 2015.

[23] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Predictable 802.11 packet delivery from wireless channel measurements," in *Proceedings of ACM SIGCOMM*, 2011.

[24] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.