WiRa: Enabling Cross-Technology Communication from WiFi to LoRa with IEEE 802.11ax

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

***Abstract*—Cross-Technology Communication (CTC) is an emerging technique that enables direct interconnection among incompatible wireless technologies. Recent work proposes CTC from IEEE 802.11b to LoRa but has a low efficiency due to their extremely asymmetric data rates. In this paper, we propose *WiRa* that emulates LoRa waveforms with IEEE 802.11ax to achieve an efficient CTC from WiFi to LoRa. By taking advantage of the OFDMA in 802.11ax, *WiRa* can use only a small Resource Unit (RU) to emulate LoRa chirps and set other RUs free for high-rate WiFi users. *WiRa* carefully selects the RU to avoid emulation failures and adopts WiFi frame aggregation to emulate the long LoRa frame. We propose a subframe header mapping method to identify and remove invalid symbols caused by irremovable subframe headers in the aggregated frame. We also propose a mode flipping method to solve Cyclic Prefix errors, based on our finding that different CP modes have different and even opposite impacts on the emulation of a specific LoRa symbol. We implement a prototype of *WiRa* on the USRP platform and commodity LoRa device. The extensive experiments demonstrate *WiRa* can efficiently transmit complete LoRa frames with the throughput of 40.037kbps and the symbol error rate (SER) lower than 0.1.**

I. Introduction

Internet of Things (IoT) promotes the rich diversity of wireless technologies such as WiFi, ZigBee, LoRa, and etc. To achieve the seamless IoT connectivity, interconnecting devices using different wireless technologies is necessary yet challenging. The emerging Cross-technology Communication (CTC) technique is a promising solution. CTC enables the direct transmissions among incompatible wireless technologies by establishing side channels with distinguishable transmission patterns or directly emulating the waveform of the target signal. With the ability of direct interconnection, CTC can open up new application scenarios and network architectures. For example, a commodity WiFi AP with CTC can act as the multi-technology gateway to provide more convenient Internet access for various IoT devices [1].

Early CTC methods on 2.4GHz band mainly focus on interconnection among WiFi, Bluetooth, and ZigBee [2]–[11]. But recently, LoRa, which usually operates on sub-GHz bands, becomes available on the globally available 2.4GHz ISM band [12]. Enabling CTC between LoRa and other technologies attracts increasing research interests [1], [13]–[17]. Symphony [13] achieves CTC from BLE/ZigBee to LoRa by generating continuous single-tone sinusoidal waves with special frame payloads. BLE2LoRa [16] leverages the BLE frequency shifting to emulate LoRa signal. XFi [1] enables the uplink reception of LoRa data using the WiFi radio and uses the WiFi AP as a LoRa gateway. When using WiFi as the LoRa gateway, establishing the downlink from WiFi back to LoRa is also necessary. Wi-Lo [17] is a CTC from WiFi to LoRa that emulates LoRa Chirp Spreading Spectrum (CSS) waveform by the single carrier Complementary Code Keying (CCK) waveform of IEEE 802.11b.

Though effective, such an emulation method suffers serious spectrum inefficiency problem due to the extremely asymmetric data rate. To achieve LoRa with a few hundreds of Hertz bandwidth and tens of kbps data rate, stopping the outgoing downlink traffic of all the high-rate WiFi users will seriously decrease the spectrum utilization. It is worth noting that the data rate asymmetry problem also exists in CTC methods from WiFi to other low-rate technologies such as Zigbee. But a ZigBee packet takes only less than 4.25*ms*, which is an acceptable overhead to obtain CTC benefits. However, an emulated LoRa frame can occupy a channel for tens and even hundreds of milliseconds, resulting in a heavy overhead that counteracts the CTC benefits. As analyzed in Section III, when WiFi traffic is 30Mbps, WiFi's throughput degradation caused by LoRa emulation can be up to 23.81 Mbps. In a nutshell, how to enable CTC from WiFi to LoRa without affecting other WiFi downlink users is still an open problem.

With the development of WiFi, IEEE 802.11ax [18] is coming into effect as the standard of WiFi 6 [19]. IEEE 802.11ax adopts orthogonal frequency domain multiple access (OFDMA) that divides the available spectrum into resource units (RUs) of various sizes to enable multi-user transmissions at the same time. Such an usage inspires us that the WiFi sender can allocate a RU overlapping with the LoRa channel and emulate LoRa waveform only on the small RU rather than the whole spectrum. Then WiFi users can still use other RUs to keep high-rate WiFi transmissions.

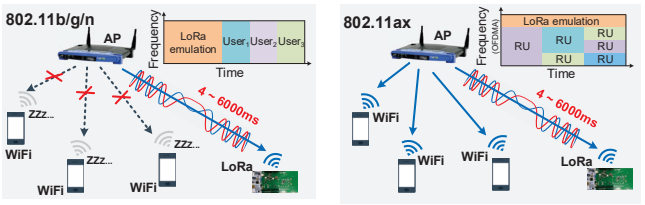
However, it is non-trivial to emulate a complete LoRa frame with 802.11ax. First, due to the diversity of RU choices, how to choose a RU suitable for emulation is unknown. Using the RU with minimum bandwidth covering the LoRa channel can release more spectrum but may lead to emulation failures. Second, the higher data rate of 802.11ax aggravates the asymmetry of data rates between WiFi and LoRa. A single 802.11ax frame cannot emulate an entire LoRa frame and even not a complete symbol in many cases. Only when

the LoRa symbol duration is shorter than 0.16*ms*, a single 802.11ax frame can barely emulate the header of a LoRa frame, which significantly limits the practical applications. A possible solution is using frame aggregation that combines multiple frames into a single transmission unit to extend the WiFi frame length for emulation of more LoRa symbols. But the subframe headers in an aggregated frame inevitably incur invalid symbols during emulation. How to identify and remove these invalid symbols to recover the LoRa frame is challenging. Third, the LoRa waveform emulated by 802.11ax inherently suffers from Cyclic Prefix (CP) errors, which are hard to cope with. Existing CP error alleviation methods for other technologies such as WEBee [9] and WIDE [11] usually group CP segments to reduce chip errors and then rely on the error tolerance of ZigBee to recover the symbols. However, these methods do not work for LoRa because the LoRa symbol doesn't have such high redundancy to tolerate errors. The discontinuity of LoRa chirps caused by CP will result in confusing frequency components when decoding. Since a LoRa symbol is emulated by multiple WiFi symbols, multiple CP distortions within one symbol will aggravate symbol errors.

By addressing the challenges, we propose *WiRa*, a novel WiFi to LoRa CTC approach that uses a small IEEE 802.11ax RU to emulate the complete LoRa frame that can be received by commodity LoRa devices. We analyze the LoRa chirps emulation solving process and find the necessary condition of a successful emulation. Then we select the RU that maximizes the spectrum efficiency with emulation reliability guarantee. By elaborately selecting the content of subframe headers, we construct a distinguishable symbol error pattern to remove invalid symbols caused by subframe headers and reconstruct a complete LoRa frame. To cope with CP errors, we extensively analyze the influence of CP errors on LoRa decoding and find that different CP modes lead to different and even oppositive destructive effects on LoRa symbols. Based on our findings, we design a mode flipping method to reduce CP errors. The main contributions of this work are summarized as follows.

- We identify the low spectrum efficiency problem caused by the extreme asymmetry of data rates between WiFi and LoRa. Then we propose emulating LoRa on a small RU rather than the whole spectrum to improve the efficiency.
- We address several technical challenges of emulating LoRa with 802.11ax and propose the designs of *WiRa*. We find the condition of a successful emulation and accordingly select the most suitable RU for emulation. We design a subframe header mapping method that can identify and remove the invalid symbols to reconstruct a complete LoRa frame. We also reduce CP errors by flipping the CP modes to alleviate the distortions.
- We implement a prototype of *WiRa* with USRP N210 platform and commodity LoRa chips (Semtech SX1280). Our extensive experiments demonstrate that *WiRa* can reliably transmit complete LoRa frames with the throughput of 40.037kbps and the SER lower than 0.1.

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 design of *WiRa* in Section IV. We present the evaluation of *WiRa* in Section V and conclude our work in Section VI.



(a) LoRa emulation with IEEE 802.11b/g/n (b) LoRa emulation with IEEE 802.11ax

Fig. 1. LoRa emulation with different WiFi protocol.

II. Related Work

CTC is a promising technique that enables direct transmissions among incompatible wireless technologies. Many studies have achieved CTC among WiFi, Bluetooth, and ZigBee [2]–[11], [20]–[36]. Packet-level CTC establishes side channels by manipulating packet transmissions to convey data. The incompatible receiver can decode CTC bits by identifying the distinguishable patterns such as transmission timing [2], signal strength [3], [4], channel state information [5], [6], [8], and etc. WEBee [9] first proposes the physical-level CTC that enables WiFi radios to emulate ZigBee signals by specific WiFi payloads. Then researchers further propose emulation based CTC for other technologies, such as BlueBee from BLE to ZigBee [10] and BlueFi from BLE to WiFi [36].

Recently, with the development of 2.4GHz LoRa, CTC between LoRa and other technologies attracts increasing research interests [1], [13]–[17]. Symphony [13] achieves CTC from BLE/ZigBee to LoRa by generating single-tone sinusoidal signals with chosen BLE/ZigBee payloads. LoRaBee [14] realizes CTC from LoRa to ZigBee by leveraging chirp-like patterns in the Received Signal Strength (RSS) sequence on a ZigBee receiver to recognize the transmitted LoRa symbols. BLE2LoRa [16] is a CTC from BLE to LoRa that leverages the frequency shifting of BLE to resemble LoRa chirps with linearly varying frequencies.

Interconnecting WiFi and LoRa also attracts researchers' interests due to the Internet accessibility of WiFi. XFi [1] enables the uplink reception of LoRa data using the WiFi radio. Wi-Lo [17] is the only CTC work that enables the downlink from WiFi to LoRa. Wi-Lo leverages the waveform similarity between LoRa CSS and IEEE 802.11b to emulate LoRa. Though effective, Wi-Lo doesn't fully consider the extremely asymmetric data rates between WiFi and LoRa and therefore suffers serious spectrum inefficiency problem. Different from Wi-Lo, our work emulates LoRa waveforms with 802.11ax on a small RU and releases the rest spectrum for other high-rate WiFi users.

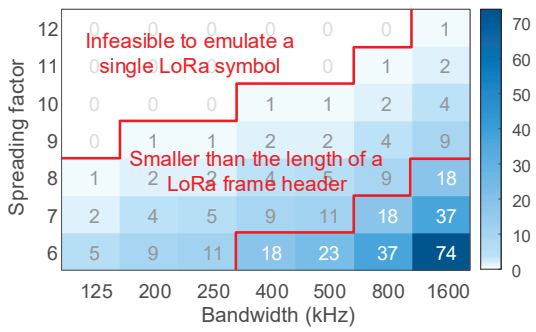


Fig. 2. The number of symbols that a single longest WiFi frame can emulate.

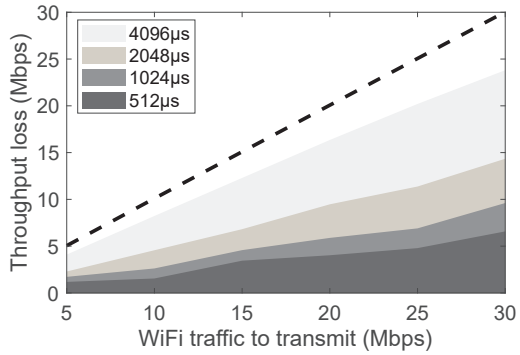


Fig. 3. WiFi performance degradation caused by LoRa frame emulation.

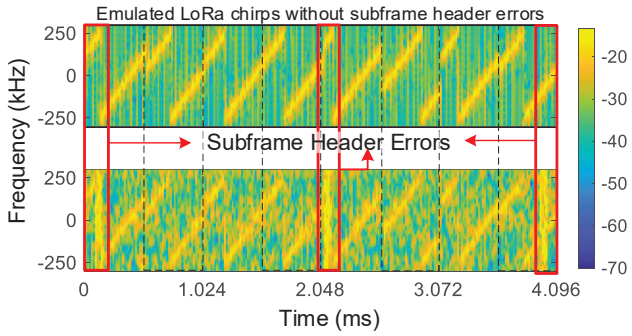


Fig. 4. Illustration of the chirp distortions caused by subframe header errors.

III. Motivation

A. Low Spectrum Efficiency Caused by Asymmetric Data Rate

The asymmetric data rate problem widely exists in CTC from high-rate technologies to low-rate technologies. A high-rate packet can only emulate a limited number of symbols of the low-rate technology. The length of the emulated payload is therefore much shorter than the original, hindering the practical applications of CTC.

This asymmetric data rate problem is especially serious for CTC from WiFi to LoRa due to the huge difference between their data rates. The extreme asymmetric data rate can cause emulation failures in many cases. Fig. 2 presents the number of LoRa symbols that a single longest 802.11b frame can emulate. We can find that for more than 85% of the common LoRa settings, a single 802.11b frame cannot emulate a complete LoRa frame and even not a complete LoRa symbol. When adopting the setting of SF=10 and BW=250kHz, the duration of a single LoRa symbol can be up to 4.096*ms*, which is even longer than the duration of the longest 802.11b frame.

Even if a LoRa symbol can be emulated by a single 802.11b packet, we have to use multiple packets to emulate a complete LoRa frame, which can easily occupy the WiFi channel for tens and even hundreds of milliseconds. To achieve LoRa with a few hundreds of Hertz bandwidth and tens of kbps data rate, stopping the downlink traffic of all coexisting high-rate WiFi users will seriously decrease the spectrum utilization. From the example in Fig. 1(a), we can find that all WiFi users cannot be served until the AP finishes emulating the LoRa frame.

To quantitatively show the low efficiency, we measure the WiFi throughput degradation when utilizing the whole spectrum to emulate LoRa frames. As Fig. 3 shows, the LoRa frame emulation does introduce huge overhead and causes the WiFi throughput decline dramatically. For all the symbol duration, the throughput decreases with the increase of WiFi traffic. When WiFi is transmitting traffic with 30Mbps, emulating LoRa with symbol duration of 4096μs can lead to a WiFi throughput loss of 23.81Mbps. Meanwhile, the throughput loss caused by LoRa emulation increases sharply with the increase of LoRa symbol duration. When the LoRa symbol duration increases from 512μs to 4096μs, the throughput loss increases by 261.85%.

Though CTC from WiFi to LoRa is very attractive to expand the ubiquity of wireless connections and open up new IoT application scenarios, such a low spectrum efficiency hinders its real application in practical scenarios.

B. IEEE 802.11ax based LoRa Emulation

With the development of WiFi, the emerging IEEE 802.11ax proposes OFDMA that divides the whole bandwidth into multiple Resource Units (RUs) and enables transmissions of multiple users at the same time. Such a usage inspires us to improve the spectrum efficiency by allocating a RU for emulation rather than the whole spectrum. Then other WiFi users can still use other RUs for their own transmissions, as shown in as shown Fig. 1(b). Besides, we can leverage the frame aggregation of 802.11ax to extend the emulation length instead of using multiple packets that is easily interrupted by other devices. Though promising, using 802.11ax to emulate LoRa faces new challenges.

1) Subframe Header Errors: To extend the WiFi frame length for emulation of a complete LoRa frame, we use 802.11ax frame aggregation that combines multiple frames into a single transmission unit. Frame aggregation avoids the negative impacts of uncontrollable interframe spaces (IFS) on LoRa emulation when using multiple packets. But in an aggregated frame, subframe headers cannot be omitted and therefore inevitably incur emulation distortions, as shown in Fig. 4. Subframe errors can cause broken LoRa symbols,

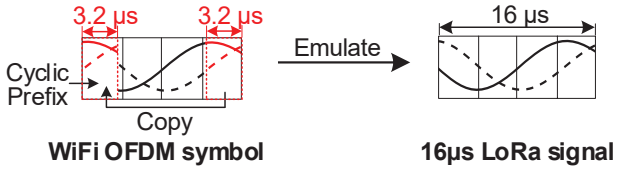
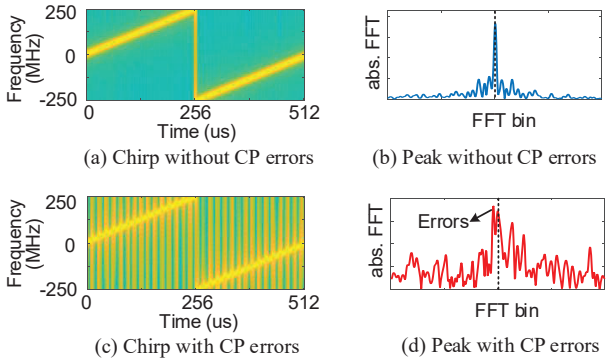


Fig. 5. Illustration of Cyclic Prefixing.



(a) Chirp without CP errors (b) Peak without CP errors

(c) Chirp with CP errors (d) Peak with CP errors

Fig. 6. Illustration of Cyclic Prefix errors.

resulting in consequent packet detection failures and decoding errors. Eliminating subframe errors to recover the LoRa frame is necessary yet challenging.

2) Cyclic Prefix Errors: Due to WiFi hardware restrictions, the LoRa waveform emulated by 802.11ax inherently suffers from Cyclic Prefix (CP) errors, which are hard to recover. As shown in Fig. 5, for each OFDM symbol, CP copies the last $3.2\mu s$ signal of an $16\mu s$ OFDM symbol as the first $3.2\mu s$ signal to avoid inter-symbol interference. Since LoRa symbols do not have such repetitions, CP will lead to the discontinuity of a LoRa chirp. What's worse, a LoRa symbol is emulated by multiple WiFi symbols, multiple CP distortions within one LoRa symbol will lead to serious symbol errors. In the example shown in Fig. 6, a $512\mu s$ LoRa chirp can experience as many as 32 CP distortions and the emulated chirp will be distorted from Fig. 6(a) to Fig. 6(c). Then the frequent discontinuities will result in confusing frequency components that cannot be removed, as shown in Fig. 6(d), leading to decoding errors. The index of the maximal energy peak is incorrectly decoded to chirp symbol, rather than the correct second energy peak. Existing CP error alleviation methods for other technologies do not work for LoRa. For example, WEBee [9] and WIDE [11] usually disperse the impact of CP to the left/right-most or middle boundaries to reduce chip errors caused by CP and then rely on the error tolerance of ZigBee to recover the symbols. But a LoRa symbol doesn't have such high redundancy to tolerate CP errors. How to reduce the CP errors when emulating LoRa with 802.11ax is still an open question.

IV. System Design

WiRa utilizes a small IEEE 802.11ax RU to emulate a complete LoRa frame that can be recognized and decoded by

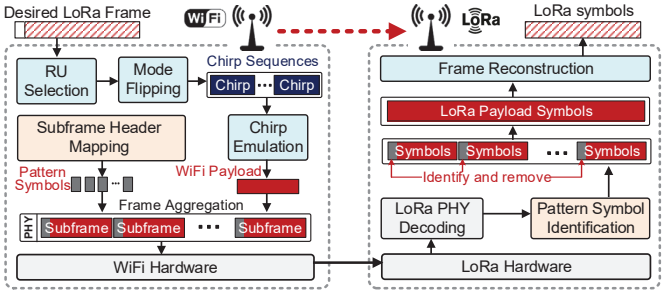


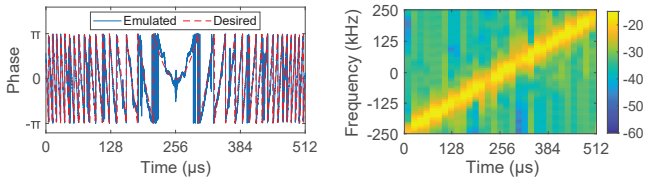
Fig. 7. The framework of *WiRa*.

commodity LoRa devices, as shown in Fig. 7. Given the LoRa frame to emulate, *WiRa* first carefully selects the suitable RU to emulate LoRa chirps by chirp emulation (Section IV-A). Then, *WiRa* leverages frame aggregation to construct a long WiFi frame and uses subframe header mapping to construct a distinguishable error pattern, which can be used to identify and remove invalid symbols to reconstruct the LoRa frame (Section IV-B). *WiRa* also designs a mode flipping based CP error reduction method based on our finding that different CP modes have different and even opposite distortions for a specific symbol (Section IV-C). Finally, we propose a frame reconstruction method that enables commodity LoRa devices recover the LoRa frame from the decoded symbols with errors (Section IV-D).

A. Chirp Emulation

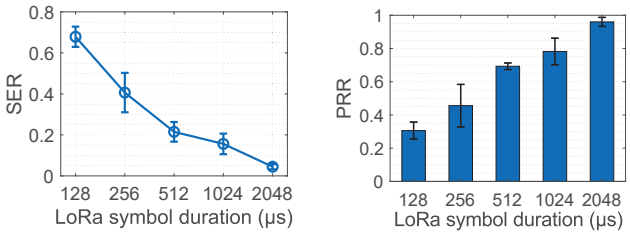
LoRa adopts the Chirp Spreading Spectrum (CSS) modulation that modulates data into chirp symbols whose frequency changes linearly over time. Since LoRa encodes data by shifting the initial frequency, the conversion between instantaneous frequency and instantaneous phase provides an opportunity for emulating chirps by 802.11ax Quadrature Amplitude Modulation (QAM) signal. We can utilize the phase sequence to resemble the spectrum of LoRa chirp. The waveform with the desired phase sequence is fed into FFT (Fast Fourier Transform) module and then we select the nearest QAM constellation points to construct the WiFi payload. Fig. 8(a) shows the desired and emulated phase sequence of LoRa symbol "0". It is easy to see that the desired phase sequence is approximated well by the emulated sequence. The spectrum of emulated chirp is shown in Fig. 8(b). The results show that the spectrum of a LoRa chirp can be resembled by the phase sequence.

Given the desired phase sequence constructed by a sequence of QAM points, we need to translate the QAM sequence into coded bits and then obtain the source bits in WiFi payload. To be resilient to noise, WiFi uses interleaving and convolutional coding that maps arbitrary source bits into a constrained set of coded bits. Such randomness makes it difficult to restore the source bits from desired coded bits when emulating. Let Galois Finite field matrices (GF(2)) of I and G represent the interleaving and convolutional coding matrices, respectively. Then, the WiFi encoding can be formulated as $Y = (IG)X$,



(a) Phase of emulated LoRa chirp (b) Spectrum of emulated LoRa chirp

Fig. 8. An example of the emulated LoRa chirp.



(a) SER (b) PRR

Fig. 9. Emulation performance degradation caused by subframe headers.

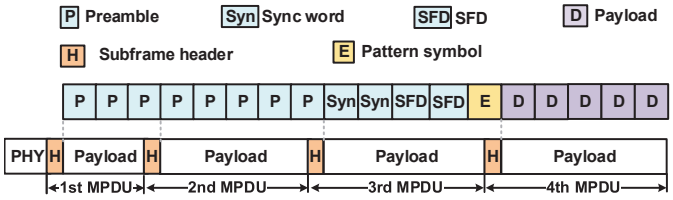


Fig. 10. Case 1: Single subframe payload is enough to emulate 2 sync words and 2.25 SFD symbols.

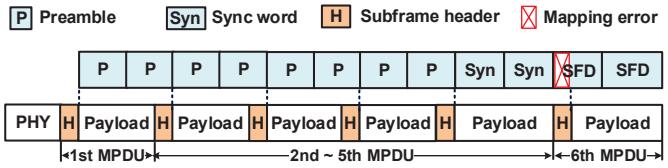


Fig. 11. Case 2: Single subframe payload only can emulate 2 sync words.

where X and Y are the source and coded bits. Suppose a RU has m data subcarriers and uses K-QAM, then X can be translated into $(m \cdot \log_2 K)$ coded bits and $\log_2 K$ coded bits correspond to a QAM point.

To emulate a LoRa chirp with bandwidth BW, a WiFi RU needs at least $s = \lceil \frac{BW}{78.125} \rceil$ subcarriers with the bandwidth of 78.125kHz for each. Though using the RU with minimum bandwidth covering the LoRa channel can release more spectrum, such a greedy selection can lead to emulation failures. Since LoRa bandwidth is not aligned to the WiFi RU bandwidth, the emulation actually manipulates source bits X but can only control a subvector of the coded bits Y, denoted as $\hat{Y}$. Then, the emulation can be formulated as $(\hat{I}\hat{G})X = \hat{Y}$, where $\hat{I}$ and $\hat{G}$ are submatrices of I and G. Given a code rate of c, $\hat{I}\hat{G}$ is a $(s \cdot log_2 K) \times (m \cdot log_2 K \cdot c)$ matrix. We can find X by calculating the inverse of $\hat{I}\hat{G}$ under GF(2). However, the inverse of $\hat{I}\hat{G}$ can be calculated by Gaussian Elimination only when $\hat{I}\hat{G}$ is full row-rank or square. Using a RU with the minimum number of subcarriers may not meet this requirement, leading to emulation failures.

To ensure that $\hat{I}\hat{G}$ is full row-rank or square, the condition $(s \cdot \log_2 K) \leq (m \cdot log_2 K \cdot c)$ should be satisfied. Given the fixed c and s, we can select a minimal WiFi RU satisfying $m \geq \frac{s}{c}$ to guarantee the success of emulation. In another word, *WiRa* can emulate any desired phase sequence with an arbitrary combination of s QAM points, when $m \geq \frac{s}{c}$.

In WiFi OFDMA, without hardware modification, the pilot subcarriers in each RU are uncontrollable, which may incur emulation errors. Therefore, similar to WEBee [9], *WiRa* also avoids the overlapping between WiFi pilot subcarriers and LoRa frequency band by channel mapping.

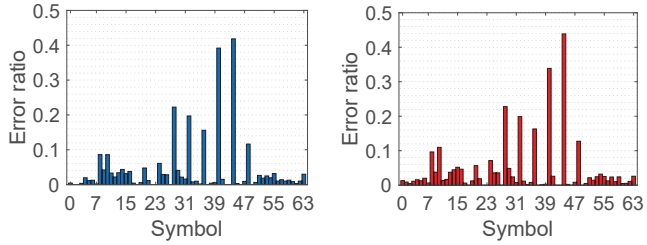
B. LoRa Frame Emulation

WiRa constructs a WiFi Aggregate MAC Protocol Data Unit (A-MPDU) to emulate a complete LoRa frame. Since each WiFi subframe has a header consisted of 4bytes Delimiter and a 30bytes MAC header, the irremovable subframe headers will inevitably incur emulation errors.

We first investigate the impact of subframe headers on LoRa frame emulation. We plot the SER and PRR under subframe header errors for different LoRa symbol duration in Fig. 9. We can observe a clear destructive effect of subframe headers, especially for the LoRa symbols with shorter duration. When the LoRa symbol duration is $128\mu s$, SER can be up to 0.679 and the PRR is only 0.306. But for the longer symbol duration such as $2048\mu s$, SER caused by subframe header errors is only 0.044. This is because a subframe header lasts $40.3\mu s$, which mainly causes serious waveform distortions for short symbol durations. From the results, we learn that we can ignore the influence of subframe headers when the LoRa symbol duration is larger than $1024\mu s$ and concentrate on solving the subframe header errors for symbols with short LoRa symbol duration. Since the subframe header cannot be omitted, we cannot avoid the distortions and can only identify and remove those wrong symbols after demodulation to recover the LoRa frame.

For the LoRa header part, we control positions of subframe headers to avoid corruption of the important fields. Compared with the preamble, LoRa Sync words and SFD are more vulnerable. Therefore, we consciously avoid mapping subframe headers to Sync and SFD symbols. When a single subframe payload is enough to emulate 2 Sync and 2.25 SFD symbols, we first map the Sync and SFD symbols into a single subframe payload, as shown in Fig. 10. And then the subframe header is mapped to the tail of the last preamble symbol. For example, in Fig. 10, the subframe header of 2^{nd} and 3^{rd} MPDU is mapped to the tail of 3^{rd} and 8^{th} preamble symbol.

For the LoRa symbol duration of $1024\mu s$, a single subframe payload is not enough to emulate both Sync and SFD symbols. In this case, we first map two Sync symbols into a single subframe payload. Then the first emulated SFD symbol will encounter distortions due to the following subframe header, as shown in Fig. 11. But this distortion can be tolerated by SFD



(a) 1^{st} symbol sequence (b) 2^{nd} symbol sequences

Fig. 12. Error ratios of 64 symbols in frames with different symbol sequences.

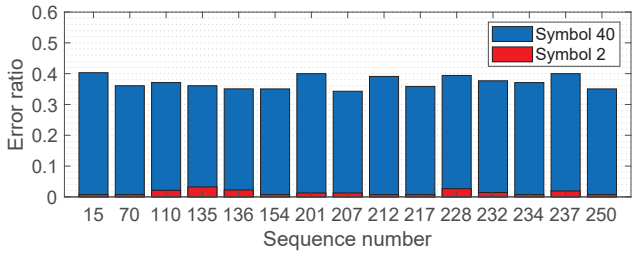


Fig. 13. Error ratios of symbol 2 and 40 at different positions.

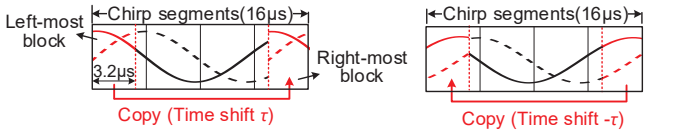
detection in this case because the SFD duration ($2.25 \times 1024 = 2304\mu s$) is much longer than a WiFi subframe header duration ($40.3\mu s$). Hence, the distortion causes negligible influence.

For the LoRa payload part, we introduce an extra specific error symbol such as base up-chirp to intentionally construct a distinguishable symbol pattern at the subframe error position. The LoRa symbols used for constructing the pattern are called pattern symbols in short. Then, we can eliminate the subframe header errors by identifying and removing the pattern symbols. When the normal LoRa payload has symbols same to the pattern symbols, we will add pattern symbols before the payload symbol to avoid confusion. Then, we can still hold this payload symbol when removing the pattern symbols during decoding. For example, if we use "00" as the pattern symbols to indicate the subframe errors, then we will change the normal symbol "0..." to "000...". Note that for the long symbol duration ($\geq 1024\mu s$), the subframe header distortions have negligible impact and we will not add any pattern symbols to the normal chirp emulation process.

C. CP Error Alleviation based on Mode Flipping

The Cyclic Prefixing (CP) will cause confusing frequency components when decoding a symbol, which is hard to remove and leads to decoding errors. To alleviate the CP errors, we conduct several experiments on USRP N210 with LoRa PHY [37] to investigate the characteristics of CP errors. We first construct a frame (96bytes) including 64[1] different emulated LoRa symbols to measure the error ratio of different symbols. Fig. 12(a) plots the results during 300 LoRa frames. From the results, we find that:

[1]Given $BW = 500kHz$ and $SF = 8$, a LoRa frame includes $2^{(SF-2)} = 64$ different emulated LoRa chirps.



(a) Mode 1: flipping right-most block (b) Mode 2: flipping left-most block

Fig. 14. Illustration of two flipping modes.

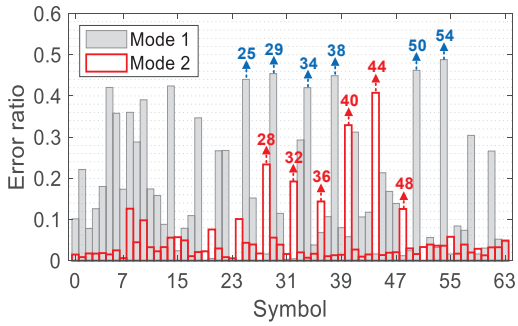


Fig. 15. Error ratio of 64 LoRa chirps with two flipping modes.

Observation 1: *For the chirp emulation of different LoRa symbols, CP causes varying degrees of distortion.*

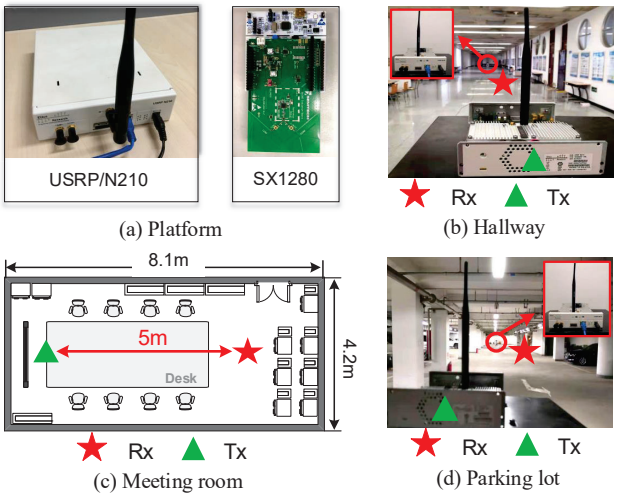
We can find the error ratios of 64 symbols have significant diversities in Fig. 12(a). Some symbols experience clearly higher error ratios. The difference of error ratios among different symbols can be as large as 0.4. To understand the diversity is caused by the position or symbol itself, we reorder the symbols in the frame and repeat the experiments. The results are shown in Fig. 12(b). We observe a very similar result. Namely, the error ratio of a symbol is independent of its position in the frame. To further verify our idea, we compare the error ratios of symbol 2 and 40 at the same frame positions[2]. The results are shown in Fig. 13, we can clearly find that the error ratio of two symbols are significantly different but consistent at different positions. That is:

Observation 2: *The destruction of WiFi CP on LoRa chirp emulation is only related to the LoRa symbol itself, regardless of its position in the frame.*

We notice that WiFi supports two CP flipping modes, flipping only left-most or right-most blocks of chirp segments before the emulation, as shown in Fig. 14. We then further study the impact of CP on emulation when using two modes. We plot the error ratios of the symbols emulated by two modes in Fig. 15. We surprisingly find that the error ratios of a symbol in two modes are so different and even opposite. Symbols with higher error ratio in Mode 1 have lower error ratio in Mode 2, such as symbol 28, 40 and 44. Conversely, the symbols with higher error ratio in Mode 2 have lower error ratio in Mode 1, such as symbol 38, 50 and 54.

By analyzing the results, we find that the time offsets in two flipping modes, τ ($\tau = 12.8\mu s$) and $-\tau$, can be translated into different phase rotations in the frequency domain. Then

[2]Due to the hamming coding, symbol 2 and 40 can only appear in certain positions at the same time, not arbitrary positions.



(a) Platform (b) Hallway (c) Meeting room (d) Parking lot

Fig. 16. The hardware platforms and three experiment environments.

the peaks in the frequency domain have opposite phase shifts, $e^{j2\pi f\tau}$ and $e^{-j2\pi f\tau}$, leading to opposite destructive effects on FFT peaks. Our key insight from the in-depth investigation is:

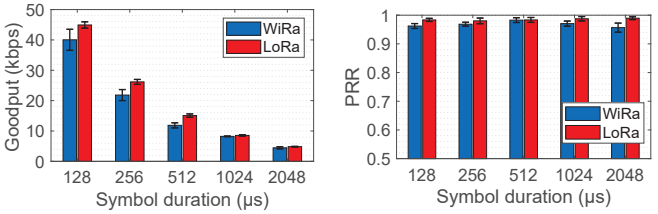
Observation 3: *Before feeding chirp segments to FFT, flipping the left-most or right-most blocks of the segment results in opposite CP errors on the same LoRa symbol.*

Based on our key insight, we propose a mode flipping based CP error alleviation method that selects the specific flipping mode for different symbols to significantly alleviate the distortions. According to Observation 2, we can estimate the CP errors of two flipping modes for each symbol in advance and then select the corresponding mode for the symbols to emulate. To estimate the best flipping mode for a symbol, *WiRa* first multiples the emulated chirp with a base down-chirp and applies FFT on the multiplication result, translating the time-domain chirp into a peak in the frequency domain. Then *WiRa* calculates the difference between the highest peak and the second peak of the FFT result. The flipping mode with a larger difference is selected for the LoRa symbol. Note that *WiRa* only needs to construct the mapping between the symbol and flipping mode once and then directly uses the mapping in online emulation.

D. LoRa Frame Reconstruction

A *WiRa* sender first selects an optimal 802.11ax RU to emulate all desired LoRa symbols with the phase sequences. When dealing with subframe header errors, we intentionally insert the pattern symbols to identify the unavoidable subframe header errors. Then the receiver has to identify and remove the wrong symbols to reconstruct the correct LoRa frame.

Upon passing the LoRa preamble detection, the emulated LoRa frame will be demodulated and decoded into a symbol sequence at LoRa receiver. To identify the pattern symbols, we first restore the demodulated symbol sequence by reversing the LoRa decoding process. Then we search all demodulated symbols that have the same value to the pattern symbols, which are suspected as the invalid symbols caused by subframe



(a) Goodput (b) PRR

Fig. 17. Performance of *WiRa* compared to the commodity LoRa with different symbol duration.

header errors. Since the subframe headers appear with equal time intervals due to the A-MPDU structure, we can leverage its periodicity to further improve the accuracy of identifying pattern symbols. We divide the demodulated symbols into groups with the length equals to the duration of a WiFi subframe. Then we use a folding method to identify the positions of pattern symbols. After identifying the pattern symbols, we can remove them to recover the LoRa frame.

V. Evaluation

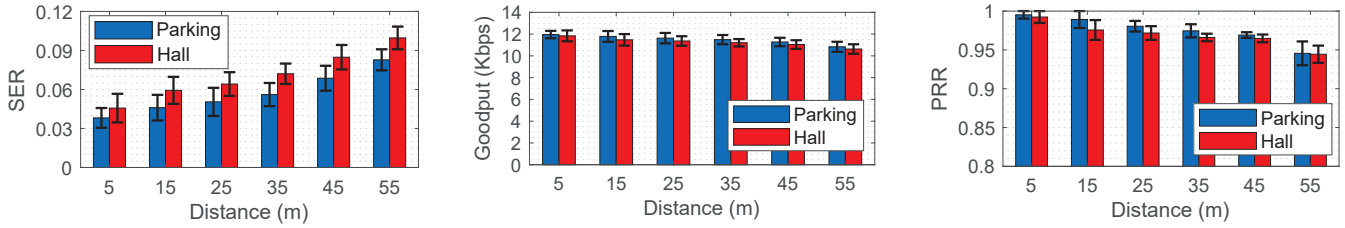
We implement a prototype of *WiRa* using USRP N210 software radios and commodity LoRa devices. The *WiRa* LoRa receiver is implemented on both USRP N210 with LoRa PHY and the commercial LoRa platform equipped with Semtech SX1280 chip [38], as shown in Fig. 16(a). We implement IEEE 802.11ax OFDMA on USRP N210 as the *WiRa* sender just for the convenience of performance evaluation. *WiRa* can be implemented on commodity WiFi 6 devices because we do not change any hardware or firmware.

In the experiments, each LoRa frame consists of 8 up-chirps as preamble, 2-symbol sync word (0x18), 2.25 down-chirps as SFD, and a variable number of payloads. We set the central frequency of WiFi channel at 2472MHz and the central frequency of LoRa channel at 2469.734MHz. By default, we set the spreading factor (SF), coding rate (CR), and bandwidth (BW) of LoRa as 8, 4/5, and 500kHz. We conduct experiments in three environments shown in Fig. 16. We first present the overall performance of *WiRa* in various scenarios and then evaluate the performance of *WiRa* on addressing the CP and subframe header errors.

A. Performance under Different Settings

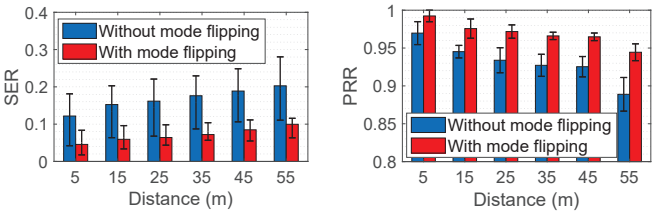
We first study the overall performance of *WiRa* under different settings, including symbol duration, payload length, transmission distance, and environments.

1) Impact of LoRa symbol duration: We first study the performance of *WiRa* with different LoRa symbol duration. We conduct experiments in the parking lot and set the distance between the *WiRa* sender and receiver to $5m$. We vary SF to obtain the LoRa symbol duration from $128\mu s$ to $2048\mu s$. The payload length of a LoRa frame is set to 160bytes. We compare *WiRa* with commodity LoRa in terms of goodput and Packet Reception Ratio (PRR).



(a) SER (b) Goodput (c) PRR

Fig. 18. Performance of *WiRa* with different transmission distance in two environments.



(a) SER (b) PRR

Fig. 19. Performance of CP mode flipping with different distance.

The experimental results are shown in Fig. 17. When increasing the symbol duration from $128\mu s$ to $2048\mu s$, the goodput of *WiRa* and the commodity LoRa decrease from 40.037kbps and 44.929kbps to 4.462kbps and 4.832kbps, respectively. Due to the inevitable invalid symbols caused by subframe headers, when the symbol duration is short ($[128\mu s, 512\mu s]$), the goodput of *WiRa* is 14.4% lower than the goodput of the commodity LoRa. But when the symbol duration is $1024\mu s$ and $2048\mu s$, *WiRa* achieves a similar goodput as the commodity LoRa because we don't introduce any overhead in these two settings. Fig. 17(b) shows that *WiRa* achieves a comparable PRR with the commodity LoRa, indicating that *WiRa* successfully conquers emulation errors.

2) Impact of transmission distance: We also evaluate *WiRa* at different transmission distance. We conduct the experiments in two environments, the hallway and parking lot. The payload length of a LoRa frame is 64bytes. The transmission power of the *WiRa* sender is $20dBm$. Due to the limited indoor space, we vary the distance between the sender and receiver from $5m$ to $55m$. Performance with longer distance is presented in the outdoor experiments in Section V-D.

We measure the SER, goodput, and PRR during experiments and plot the results in Fig. 18. Fig. 18(a) shows that when the distance increases from $5m$ to $55m$, the SER of *WiRa* in the hallway and parking lot increase from 0.046 and 0.038 to 0.099 and 0.083, respectively. Despite of the increase, the SER is still lower than 0.1. Due to the low SER, *WiRa* achieves high goodput and PRR even when the distance is large. As shown in Fig. 18(b), at distance of $55m$, the average goodput of *WiRa* are 10.834kbps and 10.630kbps in the parking lot and hallway respectively. Fig. 18(c) show that when the distance increases from $5m$ to $55m$, PRR has a slight degradation, which is within 0.1 in both environments.

B. Performance on Solving CP Errors

WiRa adopts CP mode flipping to avoid the serious distortions of CP on specific symbols. In this subsection, we evaluate its performance on reducing CP errors. We study the performance of our mode flipping based method at different transmission distance. The experiments are conducted in the hallway. We set the LoRa payload to 64bytes and the transmission power of *WiRa* sender to $20dBm$. We vary the distance from $5m$ to $55m$ and measure the SER and PRR.

The experimental results are shown in Fig. 19. We can clearly find that *WiRa* with mode flipping can achieve a much lower SER and consequently a higher PRR. When the distance increase from $5m$ to $55m$, SER of *WiRa* without mode flipping increase from 0.122 to 0.203, while the SER of *WiRa* with mode flipping increases from 0.046 to 0.099. Thanks to the error reduction by mode flipping, the SER at $55m$ is even smaller than the SER of *WiRa* without mode flipping at $5m$. Consequently, mode flipping helps *WiRa* improve the PRR from 0.889 to 0.944 when the transmission distance is $55m$.

C. Performance on Solving Subframe Header Errors

To identify and remove the invalid symbols caused by subframe headers, we intentionally construct the pattern symbols. Then the LoRa receiver can identify and remove the pattern symbols to reconstruct the correct LoRa frame. To accurately recover the frame, the key is identifying the pattern symbols. Hence, we evaluate the identification accuracy to study the effectiveness of *WiRa* on solving the subframe header errors.

The identification accuracy is related to the number of pattern symbols. Hence, we first vary the number of pattern symbols and measure the accuracy in the office environment where the *WiRa* sender and receiver are $5m$ apart. The LoRa payload length is 32bytes. The identification accuracy results are shown in Fig. 20. We can find that when the symbol duration is no less than $256\mu s$, the identification accuracy is larger than 0.93 even using only one pattern symbol. For the shortest LoRa duration, i.e. $128\mu s$, four pattern symbols can provide a nearly reliable identification result, which has an acceptable overhead of $512\mu s$. In our current implementation, *WiRa* adopts 4, 2, and 1 pattern symbols for symbol duration $128\mu s$, $256\mu s$, and $512\mu s$.

However, the more pattern symbols are used, the less effective payload symbols can be emulated by a WiFi subframe. To measure the overhead, we calculate the emulation

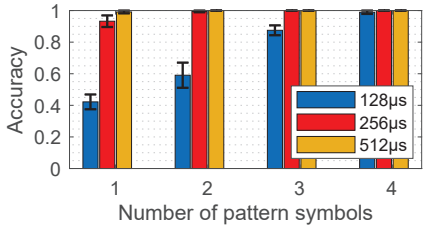


Fig. 20. Identification accuracy with different number of pattern symbols.

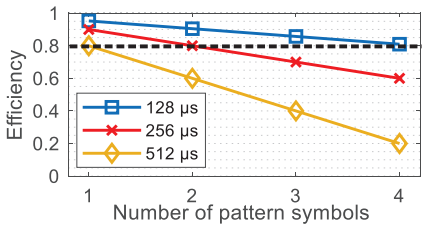


Fig. 21. Emulation efficiency using different number of pattern symbols.

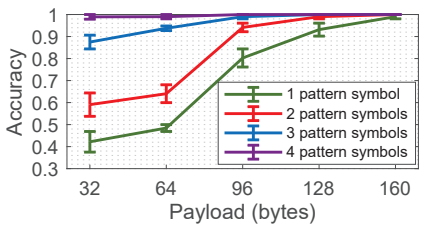
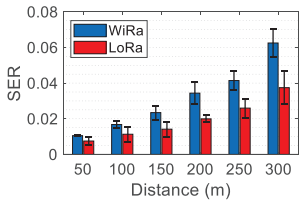
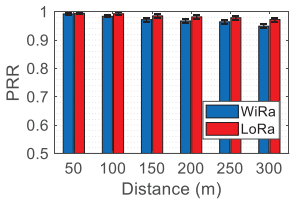


Fig. 22. Identification accuracy with different payload length (symbol duration is $128\mu s$).

(a) Setting



(b) SER



(c) PRR

Fig. 23. Long-range real-world performance of *WiRa*.

efficiency as the ratio of the number of effective payload symbols to the total number of emulated symbols within a WiFi subframe. Fig. 21 shows the emulation efficiency of using different number of pattern symbols. As expected, the emulation efficiency decreases with the increase of the number of pattern symbols. But for the number of pattern symbols adopted by *WiRa*, the efficiency is larger than 0.8 for all symbol durations. This result is consistent to the goodput of $512\mu s$ in Fig. 17, which is about 80% of the goodput of the commodity LoRa. It is worth noting that for LoRa symbols with duration $\geq 1024\mu s$, *WiRa* doesn't add pattern symbols and there will be no efficiency loss caused by our emulation.

We also study the identification accuracy with different payload length. We set the symbol duration to $128\mu s$ and the results are shown in Fig. 22. We can find that with the increase of payload length, the identification accuracy also increases. When the payload length is larger than 128bytes, using only one pattern symbol can achieve an accuracy larger than 0.9. This is because a longer payload provides more folding chances to identify the periodical pattern symbols.

D. Outdoor Scenario

We also evaluate *WiRa* in the real outdoor environments. We conduct real-world experiments along a campus road and vary the distance between sender and receiver from $50m$ to $300m$. The experimental settings are shown in Fig. 23(a). We set the spreading factor (SF) and bandwidth (BW) as 8 and $400kHz$. The LoRa payload is 64bytes and the transmission power is $20dBm$. We compare the SER and PRR of *WiRa* and the commodity LoRa. The experimental results are shown in Fig. 23(b) and Fig. 23(c). When increasing the distance from $50m$ to $300m$, the SER of *WiRa* increases from 0.011 to 0.063, while the SER of commodity LoRa increases from 0.008 to 0.037. Though it has a higher SER due to the imperfection of emulated signal, *WiRa* has already achieved a comparable performance to the commodity LoRa in terms of both SER and PRR.

VI. Conclusion

In this paper, we propose *WiRa*, a novel CTC from WiFi to LoRa that can emulate complete LoRa frames with IEEE 802.11ax. Instead of using the whole spectrum, *WiRa* only uses a carefully selected small 802.11ax RU to emulate LoRa chirps. To extend the emulation length, WiFi frame aggregation is used but brings emulation errors caused by unremovable subframe headers. We propose a subframe header mapping method to build a distinguishable symbol pattern at the subframe headers. By identifying and removing the incorrect symbols, *WiRa* can reconstruct a complete LoRa frame. To cope with CP errors, we propose a mode flipping method. We implement a prototype of *WiRa* with USRP N210 platform and commodity LoRa chips. We conduct extensive experiments to evaluate the performance of *WiRa* in various settings. The results show that *WiRa* can achieve efficient CTC from 802.11ax to LoRa. The throughput of *WiRa* can be 40.037kbps and the SER is lower than 0.1.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China under Grant 62072050, the A3 Foresight Program of NSFC under Grant 62061146002, the Funds for Creative Research Groups of China under Grant 61921003, the Open Funding of Key Laboratory of Dependable Service Computing in Cyber Physical Society (Chongqing University), Ministry of Education of China, under Grant CPS-DSC2020003, and BUPT Excellent Ph.D. Students Foundation under Grant CX2021218. Xiaolong Zheng is the corresponding author.

REFERENCES

[1] R. Liu, Z. Yin, W. Jiang, and T. He, "Xfi: Cross-technology iot data collection via commodity wifi," in *Proceedings of IEEE ICNP*, 2020.

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

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

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

[5] 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.

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

[7] 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.

[8] D. Xia, X. Zheng, L. Liu, C. Wang, and H. Ma, "c-chirp: Towards symmetric cross-technology communication over asymmetric channels," in *Proceedings of IEEE SECON*, 2020.

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

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

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

[12] G. Chen, W. Dong, and J. Lv, "Lofi: Enabling 2.4ghz lora and wifi coexistence by detecting extremely weak signals," in *Proceedings of IEEE INFOCOM*, 2020.

[13] Z. Li and Y. Chen, "Achieving universal low-power wide-area networks on existing wireless devices," in *Proceedings of IEEE ICNP*, 2019.

[14] J. Shi, D. Mu, and M. Sha, "Lorabee: Cross-technology communication from lora to zigbee via payload encoding," in *Proceedings of IEEE ICNP*, 2019.

[15] J. Shi, X. Chen, and M. Sha, "Enabling direct messaging from lora to zigbee in the 2.4 ghz band for industrial wireless networks," in *Proceedings of IEEE ICII*, 2019.

[16] Z. Li and Y. Chen, "Ble2lora: cross-technology communication from bluetooth to lora via chirp emulation," in *Proceedings of IEEE SECON*, 2020.

[17] P. Gawlowicz, A. Zubow, and F. Dressler, "Wi-lo: Emulating lora using cots wifi," *arXiv preprint arXiv:2105.04998*, 2021.

[18] W. L. W. Group, "Ieee draft standard for information technology – telecommunications and information exchange between systems local and metropolitan area networks – specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment enhancements for high efficiency wlan," *IEEE P802.11ax/D6.0*, 2019.

[24] J. Zhang, X. Guo, H. Jiang, X. Zheng, and Y. He, "Link quality estimation of cross-technology communication: The case with physical-level emulation," *ACM Transactions on Sensor Networks*, vol. 18, no. 1, pp. 1–20, 2021.

[19] "Wifi 6," Available: https://www.wi-fi.org/news-events/newsroom/wi-fi-alliance-introduces-wi-fi-6.

[20] 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.

[21] D. Xia, X. Zheng, L. Liu, C. Wang, and H. Ma, "c-chirp: Towards symmetric cross-technology communication over asymmetric channels," *IEEE/ACM Transactions on Networking*, vol. 29, no. 3, pp. 1169–1182, 2021.

[22] J. Zhang, X. Guo, H. Jiang, X. Zheng, and Y. He, "Link quality estimation of cross-technology communication," in *Proceedings of IEEE INFOCOM*, 2020.

[23] F. Yu, X. Zheng, L. Liu, and H. Ma, "Rctc: Rateless cross-technology communication," in *Proceedings of IEEE GLOBECOM*, 2020.

[25] X. Guo, Y. He, and X. Zheng, "Wizig: Cross-technology energy communication over a noisy channel," *IEEE/ACM Transactions on Networking*, vol. 28, no. 6, pp. 2449–2460, 2020.

[26] 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.

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

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

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

[30] 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.

[31] W. Jeong, J. Jung, Y. Wang, S. Wang, S. Yang, Q. Yan, Y. Yi, and S. M. Kim, "Sdr receiver using commodity wifi via physical-layer signal reconstruction," in *Proceedings of ACM MobiCom*, 2020.

[32] 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.

[33] Z. Chi, Y. Li, H. Sun, Y. Yao, Z. Lu, and T. Zhu, "B2w2: N-way concurrent communication for iot devices," in *Proceedings of ACM SenSys*, 2016.

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

[35] 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.

[36] Z. Li and Y. Chen, "Bluefi: Physical-layer cross-technology communication from bluetooth to wifi," in *Proceedings of IEEE ICDCS*, 2020.

[37] B. Seeber and the Bastille Threat Research Team, "Lora phy for gnu radio," Available: https://github.com/BastilleResearch/gr-lora.

[38] Smetech, "Semtech sx1280: Long range, low power 2.4 ghz wireless rf transceiver with ranging capability," Available: https://www.semtech.com/products/wireless-rf/lora-24ghz/sx1280.