IEEE INFOCOM 2022 - IEEE Conference on Computer Communications | 978-1-6654-5822-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/INFOCOM48880.2022.9796876

Physical-Level Parallel Inclusive Communication for Heterogeneous IoT Devices

Sihan Yu*, Xiaonan Zhang†, Pei Huang*, Linke Guo*
*Clemson University, Clemson, SC, USA
†Florida State University, Tallahassee, FL, USA
{sihany, peih, linkeg}@clemson.edu, xzhang@cs.fsu.edu

***Abstract*—The proliferation of Internet of Things (IoT) has transformed the way people interact with the world. Various kinds of wireless protocols have been developed to support diverse types of IoT communications. Unfortunately, the lack of spectrum resources puts a hard limit on managing the large-scale heterogeneous IoT system. Although previous works alleviate this strain by coordinating transmission power, time slots, and sub-channels, they may not be feasible in future IoT applications with dense deployments. In this paper, we explore a physical-level parallel inclusive communication paradigm for the coexistence of Wi-Fi and ZigBee, which leverages novel bits embedding approaches on the OQPSK protocol to enable both Wi-Fi and ZigBee IoT devices to decode the same inclusive signals at the same time but with each one's different data. By carefully crafting the inclusive signals using legacy Wi-Fi protocol, the overlapping spectrum can be simultaneously re-used by both protocols, expecting a maximum data rate (250kbps) for ZigBee devices and up to 3.75Mbps for a Wi-Fi pair over only a 2MHz bandwidth. The achieved spectrum efficiency outperforms a majority of CTC schemes and parallel communication designs. Compared with existing works on parallel communication, our proposed system is the first one that achieves an entire software-level design, which can be readily implemented on Commercial Off-The-Shelf (COTS) devices without any hardware modification. Based on extensive real-world experiments on both USRP and COTS device platforms, we demonstrate the feasibility, generality, and efficiency of the proposed new paradigm.**

***Index Terms*—Parallel Inclusive Communication; Cross-Technology Communication; Wireless Coexistence; Spectrum Efficiency**

I. Introduction

Widely deployed IoT devices have resulted in serious shortages of spectrum resources. According to a recent report in [1], up to 50 billion IoT devices will be used around the world by 2030 in various scenarios, such as home automation, smart city, and industrial IoT. In order to adapt to different application scenarios, both industrial and academia have proposed different wireless network protocols, such as Wi-Fi, Bluetooth, ZigBee, and LoRa. However, many of these protocols operate on the same frequency band, i.e., the 2.4GHz industrial, scientific and medical (ISM) band, which inevitably causes severe spectrum scarcity issues as the number of devices rockets up. Recent advances in the Cross-Technology Communication (CTC) [2]–[11] can partially alleviate this problem by enabling direct communication among heterogeneous wireless devices. However, CTC may also exacerbate the spectrum scarcity issue. Taking Wi-Fi and ZigBee coexistence as an example, many of existing CTC solutions only take advantage of the overlapping spectrum (i.e., 2MHz) to enable the Wi-Fi to ZigBee communication, while leaving the rest of Wi-Fi spectrum (i.e., 20MHz-2MHz=18MHz) occupied but unused at the time of transmission. Moreover, the spectrum scarcity issue will be magnified when a large number of coexisting IoT devices are being densely deployed, e.g., in buildings, warehouses, and hospitals.

The work of L. Guo was supported in part by the National Science Foundation (NSF) under Grant CNS-2008049 and Grant IIS-1949640.

Facing these challenges, how to optimally leverage the overlapping spectrum for enhancing the communication between coexisting protocols becomes an urgent issue. Inspired by [12], we propose a novel communication paradigm to achieve the physical-level parallel inclusive communication between Wi-Fi and ZigBee IoT devices. As shown in Fig. 1, this new design not only enables direct communication from a Wi-Fi device to a ZigBee IoT device, but also provides an additional link for another Wi-Fi receiver. Different from traditional CTC, our inclusive signals can be recognized as legitimate signals by both Wi-Fi and ZigBee receivers. Therefore, the overlapping spectrum can be re-used by both protocols while the non-overlapping spectrum can still be used to carry common Wi-Fi data as needed. It is anticipated this new paradigm can support more coexisting IoT devices without sacrificing spectrum resources.

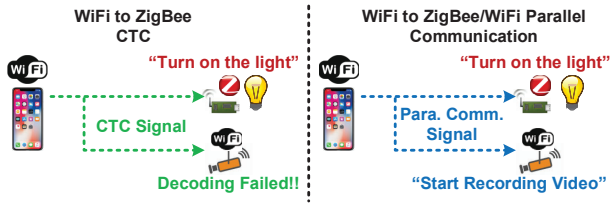


Fig. 1. Parallel Inclusive Communication Paradigm

The basic idea of the proposed design is to leverage the default redundancy of physical-level protocol to embed extra bits via modifying the original signal waveform. Although being modified, the generated inclusive signals can still be successfully decoded by error-correction schemes deployed at the receiver side, while the extra bits can be extracted via a pre-shared software-level mapper. The entirely software-level design has made the proposed paradigm to be readily implemented on a majority of Commercial Off-The-Shelf (COTS) devices, including smartphones, laptops, etc. We list

our main contributions as follows,

- We propose the first physical-level parallel inclusive communication mechanism that enables the transmission of both Wi-Fi and ZigBee data simultaneously on the overlapping channel. The entirely software-level design can be directly implemented on COTS devices.
- We develop three approaches to embed extra bits into OQPSK signals, and also design a decoding mechanism to extract them at the Wi-Fi receiver side.
- We implement our design on both the USRP N210 platform and COTS devices. Extensive experiment results show that compared with existing solutions, a maximum of 3.75Mbps additional Wi-Fi throughput can be achieved via a 2MHz overlapping bandwidth without impacting the ZigBee throughput (i.e., 250kbps).

II. Motivation and Preliminaries

A. Objectives and Challenges

Recent advances in the IoT protocols intensively exploit available spectrum to increase capacity while neglecting the coexistence of devices in the same wireless environment. Although some of the existing works [2], [13]–[16] can achieve simultaneous transmission, they mainly by coordinating transmission power, time slots, and sub-channels. This high-level coordination always requires precise synchronization among all devices, which is not feasible in heterogeneous systems with dense deployments. Moreover, the spectral is still only used by one device at a time. Recent efforts in CTC [2]–[11] enable direct communication among different protocols. However, the CTC design cannot always efficiently use the dedicated spectrum. As the pioneer work in parallel CTC, PIC [12] deploys a costly software-defined radio (SDR) to let both Wi-Fi and BLE devices decode the same signal. Unfortunately, their design can only achieve low throughput due to the symbol-level modification, and also requires a redesign of hardware at the sender side. Other small-scale techniques, such as interference cancellation [17]–[19] and network coding [20], are less effective when the network size increases, especially in terms of latency.

To sum up, designing signals that can achieve parallel communication for heterogeneous IoT devices can greatly benefit large-scale IoT deployment with limited spectrum resources.

B. Opportunities and Intuition

1) ZigBee Modulation/Demodulation: ZigBee protocol adopts the OQPSK modulation scheme using phase-shift to modulate data. ZigBee signal waveform consists of two parts: in-phase and quadrature, both of which are half-sine waves with an offset of $90°$. The waveform can be regarded as generated by a point (in the later section, we will call it "the rotation point" or simply "the point") rotating around the origin of coordinates, where its x-coordinate represents the in-phase value and y-coordinate represents the quadrature value, as shown in Fig. 2. The point rotates $90°$ per $0.5\mu s$. According to the direction of rotation (counter-clockwise or clockwise), the corresponding waveform can be demodulated as chip "1" or chip "0". After acquiring 32 chips, the 32-bit chip sequence will be decoded into a ZigBee symbol.

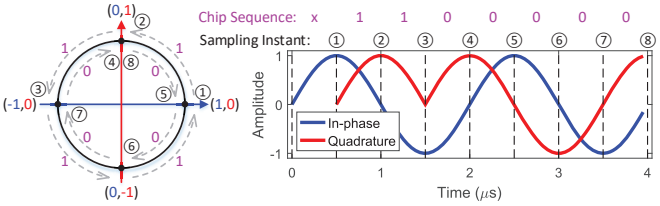


Fig. 2. ZigBee Demodulation Mechanism

2) Design Intuitions: The OQPSK protocol does not particularly emphasize the starting/ending phase during the rotation. For example, if the phase rotates from $45°$ to $135°$, the resulting signal waveform will differ from that rotating from $0°$ to $90°$. But it can still be decoded as chip "1", because the rotation direction is counter-clockwise. This property has been verified and utilized to achieve the digital emulation of ZigBee in the pioneer work [4], whose purpose is to improve the decoding accuracy of emulated signals. Compared with them, our innovation lies in transmitting more information within a single frequency band, which can support the parallel communication of heterogeneous IoT devices. Specifically, as shown in Fig. 3, some protocol-defined redundancy can be explored, such as rotating the point (1) from "a" to "b", indicating that the starting and ending point can be at some places other than the coordinate axes; (2) from "c" to "d", indicating that the rotation angle can be other than $90°$; (3) from "e" to "f", which means the distance between the point and the origin of coordinates can be changed during rotation; and (4) from "f" to "g", which means the rotation path does not necessarily to be an arc. The above changes will result in different signal waveforms for sending, but all of them will be decoded as chip "1". Therefore, the fault tolerance of the demodulation mechanism becomes an ideal carrier for transmitting extra information.

Our basic idea is to embed extra bits into the emulated ZigBee signals by changing the default phase transition patterns in Fig. 2. The extra bits can be recognized by WiFi receivers but do not interfere with the decoding of ZigBee receivers.

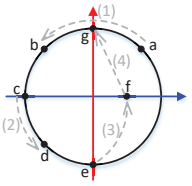
III. Physical-level Inclusive Signal Design

A. Design Overview

Fig. 4 demonstrates our proposed paradigm that mainly consists of two modules, **Extra bits embedding** and **Inclusive signal emulation**. To generate an inclusive signal, the raw ZigBee signal will be first fine-tuned to embed with needed data for parallel transmission, then the resulting ZigBee signals (w/ embedded bits) will be emulated by a Wi-Fi device. At the receiver side, the inclusive signals will be decoded by both ZigBee and Wi-Fi IoT devices in parallel.

For the design of inclusive signal, we plan to use the ladder-shaped waveform (as shown in Fig. 5(a)) [4] instead of using the standard ZigBee waveform (half-sine waveform). Ladder-shaped waveforms have the following advantages:

- The sampling value is stable during each sampling instant, which provides accurate readings for demodulation.



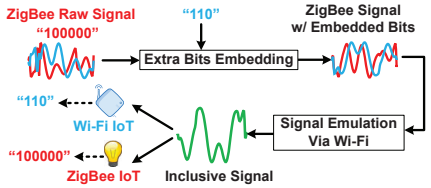
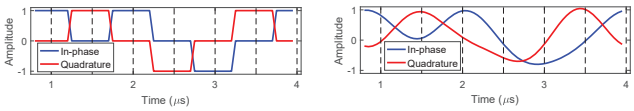


Fig. 3. High-level Freedom in Demod

Fig. 4. Flow Diagram of Parallel Inclusive Communication

- The ladder-shaped signals can be explored to represent diverse levels of amplitude for extra bits embedding.
- It can maintain the stability of chip rate to keep the consistency between standard ZigBee and our design, so that the spectrum of the customized waveforms can be the same as that of standard ZigBee.

Since only 7 subcarriers are used to emulate the inclusive signal, the final generated waveforms are not exactly the same as what we designed (i.e., the ladder-shaped waveforms), which are kind of smooth, as shown in Fig. 5(b). But they have the same tendency with the ladder-shaped waveforms, which can also be recognized by both WiFi and ZigBee receivers.



(a) Ideal Ladder-Shaped Waveform (b) Final Generated Waveform

Fig. 5. Waveform Design Comparison

B. Extra Bits Embedding

We develop the following three embedding methods to carry additional Wi-Fi bits on the overlapping spectrum without sacrificing the reliability of original ZigBee transmission.

1) Starting Phase Embedding: The OQPSK protocol demodulates the received signals by checking the rotation direction of the phase transition diagram. It does not particularly emphasize the starting/ending phase of the rotation. Motivated by this, we plan to apply different starting phases to represent extra information.

Suppose the chip sequence to be sent to the ZigBee IoT device is "100000", the point on the phase transition diagram will start from $0°$ and counter-clockwisely rotate once, then clockwisely rotate 5 times. Due to the above-mentioned demodulation method, we can define as many starting phases as possible to generate diverse waveforms, all of which will have the same decoding result due to the identical rotation mode. For example, we can define 8 (=2^3) different starting phases (each of them has an angle difference of $45°$) to represent 3-bit extra information. Fig. 6 shows two cases starting from $0°$ and $270°$ (representing extra bits "000" and "110"), respectively, which have totally different waveforms, but both of them can be decoded by ZigBee receiver as chip sequence "100000". Specifically, a Wi-Fi symbol lasts for $4\mu s$, containing 8 phase transitions. To generate a ZigBee signal waveform having extra bits, the phases of each sampling instant can be calculated as

$$\theta_j = \theta_0 + \sum_{i=1}^{8} \omega_i q_i \quad (1)$$

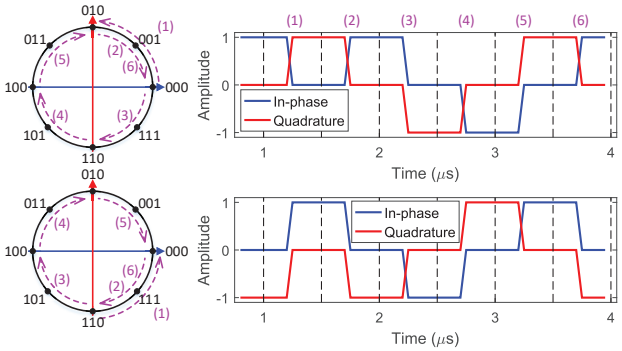


Fig. 6. Principle of Starting Phase Embedding

where θ_0 is the starting phase that can be assigned by user (e.g., $0°$), ω_i is the rotation velocity of chip "i" (e.g., $90°/0.5\mu s$) and $q_i \in \{-1, 1\}$ represents the rotation direction of the i-th element of chip sequence. After getting all the phases of the sampling instants, a ladder-shaped waveform can be generated. Meanwhile, the extra bits can be embedded.

2) Rotation Radius Embedding: The dynamically changed radius of the rotation point can also be used to embed extra bits for Wi-Fi receivers. In particular, the rotation radius ρ can be expressed as the amplitude of waveform, i.e., $\rho[x]=\sqrt{(I[x])^2+(Q[x])^2}$, where $x \in \mathbb{N}$ is the sampling instant. We can define 2^m levels of radius for choosing. Meanwhile, we can divide a waveform into n parts, each part has an individual rotation radius, so that the total number of embedded bits is mn. As shown in Fig. 7, we use 2^1 different amplitudes (i.e. $\{1, 0.5\}$) to represent 1-bit information. Then, the waveform can be divided into two parts, each of which has an individual amplitude, so that a 2-bit extra information "10" is embedded. Therefore, the changes of rotation radius can be leveraged to carry additional information.

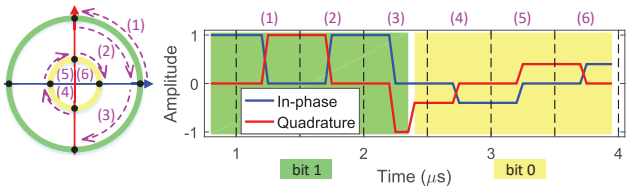


Fig. 7. Principle of Rotation Radius Embedding

3) Rotation Velocity Embedding: Instead of the default rotation velocity ($90°/0.5\mu s$), diverse rotation velocities (whose upper limit is $180°/0.5\mu s$) can be used to embed extra information. We can define 2^m levels of rotation velocities to represent m bits information. Taking Fig. 8 as an example, we provide two rotation velocities (i.e. $90°/0.5\mu s$ and $40°/0.5\mu s$) to represent bit "1" and "0", respectively. The rotation point can rotate among 4 (upper figure) and 9 (lower figure) available phases, respectively. Therefore, 1-bit extra information can be embedded by choosing different rotation velocities. Note that the rotation velocity cannot vary during the period of a WiFi symbol ($4\mu s$), otherwise the resulted spectrum will be wider and cannot be emulated with the overlapping channels.

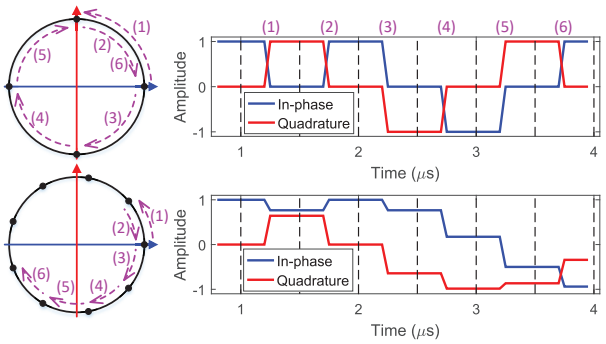


Fig. 8. Principle of Rotation Velocity Embedding

C. Embedding Strategy Trade-off

To maximally utilize the overlapping spectrum, we should embed as many extra bits as possible. However, this will also reduce the decoding accuracy due to the similarity of the generated signals. In addition, using one of the embedding approaches may also impact the amount of bits that can be embedded by other approaches. Therefore, how to properly choose the embedding approaches becomes a critical issue.

Based on our experiments and analysis, starting phase embedding is relatively independent, which doesn't impact the use of the other two approaches. This is because changing the starting phase does not impact the spectrum width. The independence of rotation radius embedding is weaker than that of starting phase embedding, because the changing of radius may slightly impact the rotation velocity, and finally degrade the accuracy of the inclusive signal. Rotation velocity embedding has the worst effect among these three due to the fact that it may alter the spectral width of the generated signals.

To adapt to different application scenarios, we define two modes: (1) **High Throughput-Driven**, to embed as many extra bits as possible to the packets while tolerating a few bit errors; (2) **Low Error Rate-Driven**, for applications that taking decoding accuracy as the first priority. Given different decoding performances, we will fully evaluate the combination of embedding methods and the number of embedded extra bits.

D. Inclusive Signal Emulation

In this subsection, we mainly focus on the signal emulation on the 2MHz overlapping spectrum, while the non-overlapping spectrum can still be used to carry common Wi-Fi data. As the pioneer works, [3], [21] show the feasibility of using Wi-Fi for achieving physical-level emulation, we will develop an optimization approach to emulate the inclusive signals that can achieve high accuracy for parallel decoding.

1) Signal Emulation via Wi-Fi: Wi-Fi and ZigBee have different constellation diagrams. The FFT points of the inclusive signals need to be quantized to the predefined 64-QAM constellation points, which brings distortion inevitably. To generate perfect inclusive signals and maximally utilize the 64-QAM constellation points, our idea is to: (1) adjust the size of 64-QAM constellation diagram to adapt it to send inclusive signals; (2) minimize the distortion in the global perspective by choosing the nearest 64-QAM constellation points for signal emulation.

We formulate the process of resizing 64-QAM as an optimization problem, as shown in (2) and (3). Eq. (2) depicts the overall quantization error, in which α is a scalar to scale the size of 64-QAM constellation diagram; P_i is a complex number that denotes the i-th constellation point among 64 predefined constellation points; P_j is a complex number that denotes the constellation point which is used to emulate inclusive signals. M is the total number of constellation points that are used to emulate inclusive signals. Eq. (3) depicts the objective of our optimization, which is to find a proper α that can minimize (2). Due to that the second derivative $E(\alpha)'' = \sum_{j=1}^{M} 2P_i^2 > 0$, (2) is a convex function and has the global minimum.

$$E(\alpha) = \sum_{j=1}^{M} min\{(\alpha P_i - P_j)^2 | i \in [1, 64]\} \quad (2)$$

$$\alpha = \underset{\alpha}{argmin} E(\alpha) \quad (3)$$

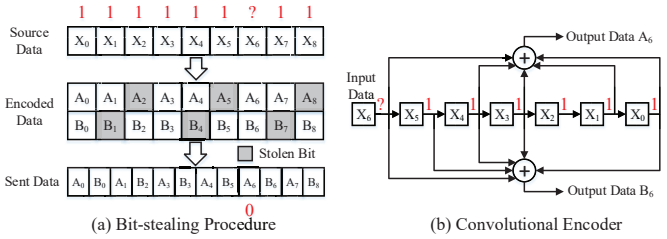
The global minimum can be acquired in $\mathcal{O}(M \log M)$ for M FFT points if using the binary search algorithm.

After finding the optimal scalar α, the 64-QAM constellation diagram can be maximally utilized to generate inclusive signals. Since only 7 overlapping Wi-Fi subcarriers are leveraged for emulation, a total number of 42 ($= 7 \times 6$ bits/symbol) bits/symbol will be used to generate the final inclusive signal.

2) Channel Coding: To ensure that the above 42 bits can correctly appear on the 7 subcarriers, it is pivotal to consider the channel coding. For a standard Wi-Fi signal generation process, each OFDM symbol contains a 216-bit sequence that will be coded to a 288-bit sequence if applying the $3/4$ convolutional coding at the physical layer. The crafted 42 bits need to be assigned to the correct position of the 288-bit sequence, which means the 216-bit sequence also needs to be well-designed. Given the positions and corresponding values of 42 out of 288 bits, we need to use "reverse engineering" to determine the sequence of 216 bits to be sent into the channel coding process.

Assume bit "0" is expected to be assigned at position A_6 (out of the 42 bits) in the sent data as shown in Fig. 9(a). The corresponding input X_6 (out of 216 bits) should be well-designed to have "0" at A_6. Suppose the rest of source data are padded with bit "1" (actually, they can be padded with any content, which do not affect the calculation process of X_6). According to the convolutional encoding process in Fig. 9 (b), we can get $A_6 = X_0 \oplus X_1 \oplus X_3 \oplus X_4 \oplus X_6$. Since we know $A_6 = 0$, we can infer that bit "0" should be assigned to X_6 in the source data.

Similarly, the source data of all the 42 bits used in generating the inclusive signals can be determined. Then, the final inclusive signals can be obtained after going through the standard Wi-Fi protocol (including scrambling, convolutional coding, interleaving, etc.). Note that our scheme is not limited to only transmitting inclusive signals on the overlapping spectrum. Instead of padding with all "0" or "1", we can also put normal Wi-Fi data on the rest 174(=216-42) bits



(a) Bit-stealing Procedure (b) Convolutional Encoder

Fig. 9. Channel Coding

before determining the values of the 42 bits, and thus the non-overlapping subcarriers can also be used for original Wi-Fi-to-Wi-Fi communication.

IV. Inclusive Signal Demodulation

The inclusive signal demodulation happens in parallel on both ZigBee and Wi-Fi receivers as shown in Fig. 10. Due to the purely software-level design, neither receiver needs to make any hardware modification for demodulating.

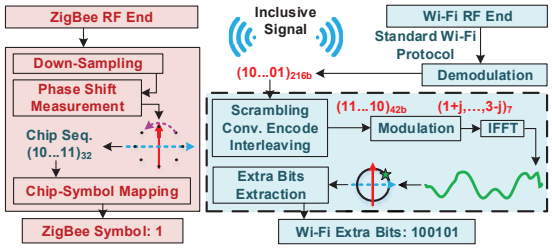


Fig. 10. Inclusive Signal Demodulation

A. ZigBee End Demodulation

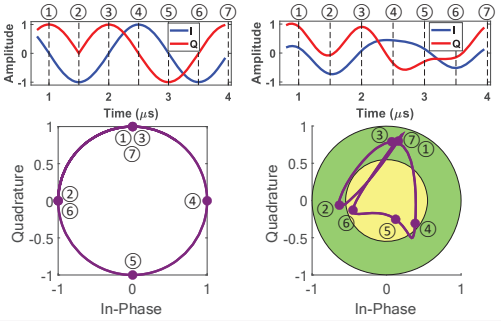
Upon receiving the inclusive signal, the ZigBee demodulation module will first apply down-sampling and then, measure the phase shift. The derived phase transition diagram, although different from the standard diagram, can still be used to output the chip sequence based on the rotation direction. Finally, the ZigBee receiver calculates the Hamming distance between the received 32-chip sequence and all the 16 predefined ZigBee symbols. The 32-chip sequence will be decoded into the ZigBee symbol with minimum Hamming distance.

Fig. 11 compares the standard ZigBee signal with an inclusive signal which is received by a ZigBee device. Although the inclusive signal has been customized to carry extra bits, the decoding process will still output the same chip sequences. Our evaluation results demonstrate that our emulation approach can generate perfect inclusive signals to maintain the ZigBee demodulation accuracy at a high level (detailed in Sec. V-C).

B. Wi-Fi End Demodulation

Different from the ZigBee demodulation process, the demodulation of embedded extra bits brings significant challenges at the Wi-Fi receiver side due to the newly designed embedding approaches. As shown in Fig. 10, the demodulation process follows the standard Wi-Fi protocol until the 216-bit bitstream has been obtained. Then, we will "reverse" the demodulation process to discover the embedded information.

Specifically, we design a component to "reconstruct" the time-domain waveform from the decoded information, as



	Standard ZigBee Waveform						Customized ZigBee Waveform					
Time Interval	①→②	②→③	③→④	④→⑤	⑤→⑥	⑥→⑦	①→②	②→③	③→④	④→⑤	⑤→⑥	⑥→⑦
Rotation Direction	CCW	CW	CW	CW	CW	CW	CCW	CW	CW	CW	CW	CW
Decoding Results	1	0	0	0	0	0	1	0	0	0	0	0

Fig. 11. Comparison of ZigBee Waveform

shown in the dashed line of Fig. 10. It is implemented at the application layer because the bitstream has been acquired from the "Demodulation" block, which represents the physical layer. First, the channel coding process is performed to convert the 216-bit information into 288-bit. The main benefit of this process is that we can leverage the convolutional coding process to obtain the reconstructed waveforms with less error. Then, the 42-bit useful information will be extracted and fed into the 64-QAM and IFFT blocks to get the reconstructed waveforms. Finally, the Wi-Fi receiver can decode the extra bits by checking the positions of the rotation point on the phase transition diagram of the reconstructed waveforms. In this step, some predefined information (e.g., the embedding approaches and the numbers of embedded bits) is shared between the sender and receivers to offer help for decoding. In the following parts, we mainly explain how the "Extra Bits Extraction" block works.

1) Extra Bits Extraction via Starting Phase Embedding: Although the starting phase can be directly acquired from the phase transition diagram, the accuracy is not good enough. We increase the sampling rate to improve the accuracy. First, we detect the chip sequence $Q = \{q_1, q_2, ..., q_6\}$ $(q_i \in \{1, -1\})$ that the received inclusive signals want to depict. In this step, the received signals may have some distortion so that we should compare the detected chip sequence with the standard chip sequence to get the correct Q. Then, since the receiver also knows the predefined optional starting phases, the receiver can list all the possible rotation modes which start from different phases. Finally, the receiver can compare the received rotation mode with the listed rotation modes to find the most similar one.

The above process can be formulated as follows. ϕ_i denotes the optional starting phase. Since the optional rotation velocity ω is also predefined, we can derive the theoretical phase sequence (Φ_i) when the starting phase is ϕ_i. After getting all possible Φ_i, we can compare the received rotation mode with all Φ_i and pick out the most similar one, so that the starting phase can be acquired.

$$\phi_i = 2\pi \frac{i}{2^n} (i \in [1, 2^n], n \in \mathbb{N})$$

$$\Phi_i = \left\{\phi_i, \phi_i + \sum_{i=1}^{1} \omega q_i, \phi_i + \sum_{i=1}^{2} \omega q_i, ..., \phi_i + \sum_{i=1}^{6} \omega q_i\right\}$$

2) Extra Bits Extraction via Rotation Radius Embedding: The rotation radius can be acquired by analyzing the distribution of the sampling points. Taking Fig. 11 as an example, suppose the receiver already knows that two extra bits are embedded into the waveform, where the first half of the waveform represents one bit and the second half of the waveform represents another bit. From the obtained phase transition diagram, it can be seen that at the first three sampling instants (①,②,③), the sampling points are located in the green area with large radius whereas at the other three instants (④,⑤,⑥), the sampling points are located in the yellow area with small radius. Thus, the first half of the waveform will be decoded as bit `1` and the second half of the waveform will be decoded as bit `0`.

3) Extra Bits Extraction via Rotation Velocity Embedding: Since the optional rotation velocity is limited and predefined, the Wi-Fi receiver can compare the derived rotation velocity with the predefined optional values to get the embedded information. First, we can acquire 63 small rotation angles (δ_i) from a Wi-Fi symbol since it contains 64 sampling points (θ_i). Then, several (L) adjacent δ_i can form a large rotation angle. We can pick these large rotation angles at a certain interval (λ) to form a sample set (Δ). The following equations describe the selection process of sample data, in which $s \geq 1$ is the starting point that can be set by user.

$$\delta_i = |\theta_{i+1} - \theta_i|, \quad 1 \leq i \leq 63$$

$$\Delta = \left\{\sum_{i=s}^{s+L-1} \delta_i, \sum_{i=s+\lambda}^{s+\lambda+L-1} \delta_i, ..., \sum_{i=s+n\lambda}^{s+n\lambda+L-1} \delta_i\right\}, \quad s + n\lambda + L - 1 \leq 63$$

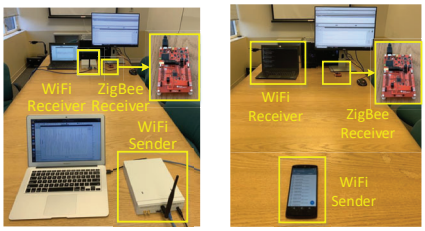
Finally, we can use the decision boundary to classify it. We continue to use the case in Fig. 11 as an example, where 2 different rotation velocity (e.g. $90°/0.5\mu s$ and $45°/0.5\mu s$) are used to represent 1 bit extra information (i.e. bit `1` and bit `0`). So, $(90° + 45°)/2 = 67.5°$ is set to be the decision boundary. After obtaining the phase transition diagram, we can find that from key point ① to key point ⑦ (in this case, $\lambda = 10, L = 10, s = 5$), the rotation point rotates $\Delta = \{107°, 101°, 124°, 26°, 100°, 114°\}$, respectively, most of which are greater than the decision boundary and closer to $90°$ instead of $45°$. Thus, the embedded information will be decoded as bit `1`.

To sum up, the aforementioned three methods are independent to each other, but they can also cooperate with each other to improve the decoding efficiency. For example, if the rotation velocity is acquired firstly, it can be used to derive the starting phase, which can greatly reduce the number of comparisons.

V. Performance Evaluation

A. Experiment Settings

We fully evaluate our proposed design on both simulation platform and COTS devices as shown in Fig. 12. The simulation platform consists of USRP along with the GNU radio and TI CC26X2R1 launchpad. Due to its flexibility in terms of modifying system parameters, the simulation platform will be used to thoroughly investigate the performance under different environmental constraints, e.g., SNRs, distances, RSS. To further demonstrate the feasibility in real application scenarios, we also evaluate the performance on COTS devices including an LG Nexus 5 smartphone as the sender and a ThinkPad X1 Carbon laptop as the Wi-Fi receiver.



(a) USRP Platform (b) COTS Platform

Fig. 12. Experiment Settings

B. Overall Performance

In this subsection, we jointly evaluate the performance in terms of decoding accuracy, time consumption, and extra throughput when adopting different extra bits embedding approaches. As shown in Fig. 12(a), a USRP N210 will send the inclusive signals while another USRP N210 will act as the Wi-Fi IoT device for decoding the extra bits.

1) Decoding Accuracy-Single Embedding Approach: We first conduct a thorough experiment to analyze the performance of each extra bits embedding approach. The experiment will focus on the decoding accuracy at the USRP receiver in terms of the correct number of extra bits extracted from the received signals. Each of the embedding approaches has been tested 50 times to minimize the environmental impact.

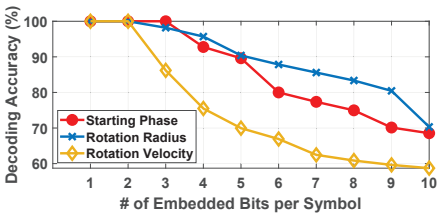


Fig. 13. Decoding Accuracy w/ Single Embedding Scheme

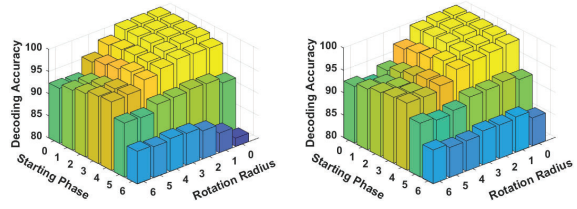
To be consistent with the following Wi-Fi throughput analysis, we evaluate decoding performance based on the total number of embedded bits on a Wi-Fi symbol. As shown in Fig. 13, the decoding accuracy drops when the number of embedded bits is increasing, where all of the three approaches follow a similar trend. In accordance with the discussion in Sec. III-C, the starting phase embedding has the best performance when the number of embedded bits is less than 4. Even if the number of embedded bits is up to 10 bits/symbol, the starting phase or rotation radius approach can still maintain an accuracy of nearly 70%. Due to the change of spectrum width, the rotation velocity approach has the worst performance.

2) Decoding Accuracy-Mixed Embedding Approach: The above results motivate us to further increase the embedded bits by combining two or three approaches. We continue to evaluate the decoding accuracy under mixed embedding strategies

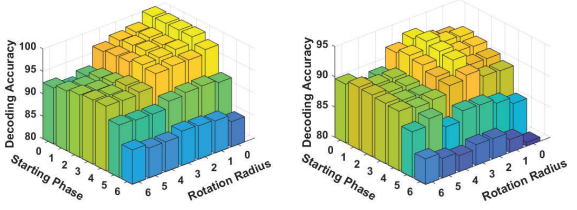
(i.e. rotation velocity (0-3 bits), rotation radius (0-6 bits), and starting phase (0-6 bits)), as shown in Fig. 14(a)-14(d). It can be seen that the starting phase/rotation radius/rotation velocity embedding has a relatively good decoding accuracy when the number of embedded bits is 0-4 bits, 0-3 bits, and 0-2 bits, respectively. One may notice that the decoding accuracy has a slight increase when the number of bits embedded by starting phase increases from 0 to 4. Eq. (4) explains the reason. When the number of embedded bits increases, both the denominator and the numerator increase and it is uncertain who grows faster. So the result is non-monotonic.

$$acc = \#\ of\ correctly\ decoded\ bits / \#\ of\ embedded\ bits \quad (4)$$

The above experimental results verify our discussions and further pave a way for choosing adequate approaches for embedding extra bits.



(a) Rotation Velocity #bit = 0 (b) Rotation Velocity #bit = 1



(c) Rotation Velocity #bit = 2 (d) Rotation Velocity #bit = 3

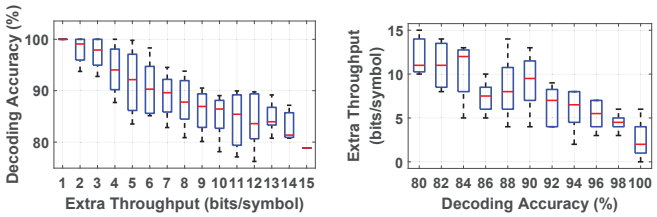
Fig. 14. Decoding Accuracy w/ Mixed Embedding Schemes

To further demonstrate the generality of the proposed scheme, we design two modes for better adapting to different real-world application scenarios.

High Throughput Mode. The objective of this mode is to embed as many bits as possible while maintaining the decoding accuracy at a reasonable level. As shown in Fig. 15(a), for each Wi-Fi symbol, the embedded bits can be up to 15, contributing a total of 3.75 Mbps (=15 bits/symbol $\times$ 250K symbols/s) extra throughput under 802.11g protocol while maintaining an average of 80% decoding accuracy.

Low Error Rate Mode. This mode is helping applications with high accuracy requirements and a relatively low throughput constraint. As shown in Fig. 15(b), by choosing proper embedding approaches, the accuracy can maintain above 80% when more than 10 bits are embedded into each symbol. Even for some extreme cases requiring higher than 95% decoding accuracy, our design can still provide an extra throughput of 5 bits/symbol.

Repeated Transmission for Reliability. In order to achieve higher reliability, we can adopt the retransmission scheme as needed. We continue to use the extra bit embedding strategies above, and evaluate the performance with multiple retransmissions. The result in Fig. 16 clearly shows the reliability



(a) High Throughput Mode (b) Low Error Rate Mode

Fig. 15. Performance under Different Operating Modes

improvement. Even for the worst case (6 bits from starting phase + 6 bits from rotation radius + 3 bits from rotation velocity), the decoding accuracy reaches 95.5% and 99.1% after 1 and 2 times of retransmission, respectively.

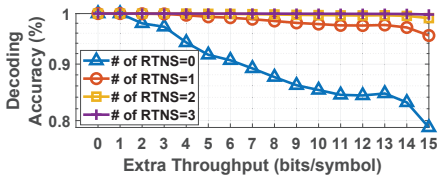
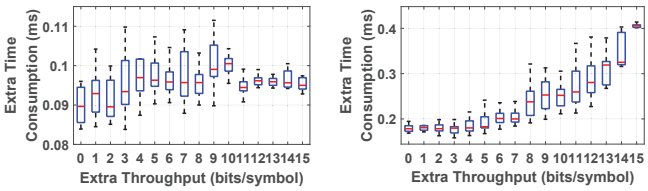


Fig. 16. Decoding Accuracy w/ Retransmissions

3) Time Consumption: The time consumption is of great importance in dedicated application scenarios. We use MacBook Air 2015 (equipped with Intel Core i7-5650U CPU @ 2.2GHz) to measure the time consumption of signal generation and decoding of extra Wi-Fi bits. We generate more than 10,000 inclusive signals with different numbers of embedded bits. Fig. 17(a) shows that the time consumption of inclusive signal generation keeps stable (usually less than 0.1ms) with the increase of embedded bits. On the receiver side, as shown in Fig. 17(b), the decoding process requires more time when the number of embedded bits is increasing, reaching a maximum of 0.4ms for 15 extra Wi-Fi bits through a combination of all embedding approaches. Note that the signal generation/decoding process can be further sped up by using advanced laptops or smartphones.

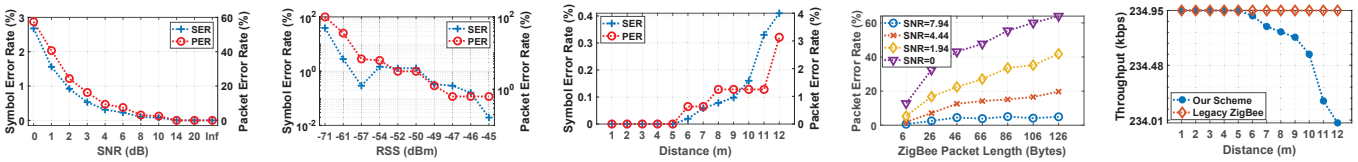


(a) Time Consumption at Tx Side (b) Time Consumption at Rx Side

Fig. 17. Time Consumption of Processing Inclusive Signal

C. ZigBee Downlink Performance

1) USRP Platform: We first evaluate how the environmental factors impact the ZigBee downlink performance. In particular, a TI CC26X2R1 launchpad is used as the receiver, while a USRP N210 performs as the sender. In each case, the USRP sends 100 packets, each packet includes 64 ZigBee symbols, each ZigBee symbol is emulated by 4 WiFi symbols. According to the above evaluation results, we choose to embed 3 bits via starting phase, 2 bits via rotation radius, and 1 bit via rotation velocity approach into each Wi-Fi symbol, resulting in a total of 24 extra bits in the to-be-emulated ZigBee signal.



(a) SER & PER v.s. SNR (b) SER & PER v.s. RSS (c) SER & PER v.s. Distance (d) PER v.s. Packet Length (e) Throughput v.s. Distance

Fig. 18. Wi-Fi-to-ZigBee Performance (USRP)

Impact of SNR. We deploy another USRP N210 to generate additive white Gaussian noise (AWGN). As shown in Fig. 18(a), while keeping the distance stable, both the symbol error rate (SER) and packet error rate (PER) at the ZigBee receiver side decrease with the increase of SNR. When the channel condition becomes better, e.g., SNR$>$ 7.96dB, the SER and PER can be lower than 0.2% and 5%, respectively. This high level of decoding accuracy is mainly due to the deployed DSSS scheme at the ZigBee receiver side, demonstrating our design will not impact the original ZigBee communication.

Impact of RSS. We also measure the SER and PER under different conditions of received signal strengths (RSS). From Fig. 18(b), we can find that both SER and PER decrease with the increase of RSS. The SER and PER can maintain at 1% and 3%, respectively, even under the condition that the signal strength is very low (approximately -50dBm).

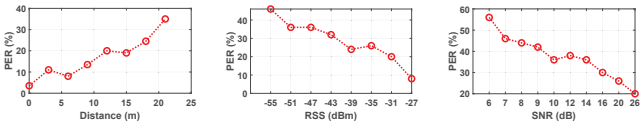
Error Rate over Distance. In this experiment, we compare the error rate over the distance in an indoor environment. As shown in Fig. 18(c), from 1 meter to 5 meters, both the SER and PER keep at 0. From 5-10 meters, the error rate increases with the increase of distance, but keeps at a low level, i.e., PER is less than 1.5% and SER is less than 0.2%, which proves the proposed design has a good effect in the indoor environment.

Packet Length Vs. PER. In practical applications, the length of packets can greatly impact the PER. When the SER is a certain value, the more symbols a packet contains, the larger possibility that the packet contains errors. This experiment specifically evaluates how the packet length impacts the PER at the ZigBee receiver side. As shown in Fig.18(d), the PER increases with the increase of ZigBee packet length, but the PER remains below 41.8% as long as the $SNR > 0$ (even if the packet length approaches the maximum size (128B)). Note that when the packet length=126B, PER=41.8% means the SER is only $1 - \sqrt[126]{41.8\%} = 0.69\%$.

Throughput Vs. Distance. We also compare the throughput of our scheme with legacy ZigBee throughput at different distances. As shown in Fig. 18(e), the legacy ZigBee can always achieve the maximum throughput, approximately 234.95 kbps (here, we define the throughput as the ZigBee payload without the PHY header/preamble). On the other hand, the throughput of our scheme will decrease when the distance is beyond 6m, but will still maintain a high level at 99.6% of the theoretical upper bound (250kbps) of legacy ZigBee.

The above results demonstrate that although our scheme adopts customized waveforms for carrying extra bits, the waveforms can still be well decoded by ZigBee receivers.

2) COTS Device Platform: In addition to the experiments carried out on the USRP platform, we also implement it on the COTS devices as shown in Fig. 12(b). Under the same experimental setting, we evaluate the performance from the perspective of the ZigBee receiver, which mainly focus on the PER instead of SER.



(a) PER v.s. Distance (b) PER v.s. RSS (c) PER v.s. SNR

Fig. 19. Wi-Fi-to-ZigBee Performance (COTS Devices)

As shown in Fig. 19(a), the PER increases when the distance between the smartphone and ZigBee receiver increases. In particular, the PER can be maintained below 20% when the distance is within 15m in the indoor environment. Compared with the result in Fig. 18(c), the increase of error rate is mainly due to the hardware limitation of the smartphone. Similarly, Fig. 19(b) shows the impact of RSS on PER. The PER can be lower than 20% when the RSS is higher than -31dBm. We also evaluate the impacts of SNRs as given in Fig. 19(c). The PER can be less than 40% when the SNR is greater than 10dB.

D. Wi-Fi Extra Bit Decoding Performance

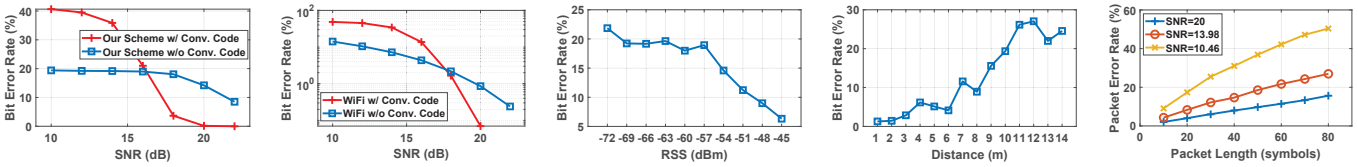
1) USRP Platform: We continue to use the USRP platform to evaluate the decoding performance of extra bits. The identical inclusive signals (as in Sec. V-C1) are used for evaluation.

Impact of Convolutional Coding. When decoding the received inclusive signal, the built-in convolutional coding mechanism can provide an error correction function. As shown in Fig. 20(a), (1) the BER drops with the increase of SNR; (2) in good channel conditions (i.e. $SNR > 17dB$), the error-correction effect is better. To further validate this observation, we let the USRP send normal Wi-Fi signals without embedding extra bits. The results in Fig. 20(b) demonstrate a similar finding, i.e., the convolutional code has a better error-correcting capability under good channel conditions. Many existing works also shows the same result [22]–[24]. Therefore, we can decide whether to use the convolutional code mechanism according to the channel conditions.

Impact of RSS. The BER decreases with the increase of RSS, as shown in Fig. 20(c). When the RSS is larger than -48dBm, the BER can be lower than 10%.

BER over Distance. The experimental result in Fig. 20(d) shows that the Wi-Fi downlink can achieve less than 20% BER when the distance is smaller than 10m.

Note that the above relatively high BER is because 6-bit extra information is embedded into each symbol. If we reduce the extra information, BER can also be reduced significantly.

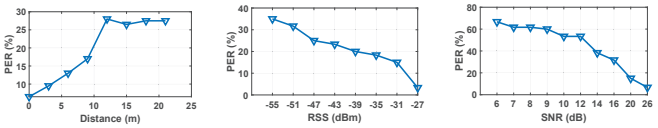


(a) Effect of Conv. Code on Our Scheme (b) Effect of Conv. Code on Normal Wi-Fi (c) BER v.s. RSS (d) BER v.s. Distance (e) PER v.s. Packet Length

Fig. 20. Wi-Fi-to-Wi-Fi Performance (USRP)

Packet Length Vs. PER. Similar to the ZigBee downlink performance, the increase of packet length will result in the increase of PER as shown in Fig. 20(e), especially for poor SNR environment. The reason is that longer packets are more likely to have bit errors, causing the failure of packet transmission.

2) COTS Device Platform: Continuing to use our COTS devices to evaluate the Wi-Fi-to-Wi-Fi performance, we use a ThinkPad X1 laptop as the receiver to demodulate the extra bits. Again, all of the inclusive signals are exactly the same as used in the Sec. V-C. The PER on COTS laptop (i.e. Fig. 21) has a similar trend as on the USRP platform (i.e. Fig. 20). From Fig. 21(a), it can be seen that the PER increases with the increase of distance and there is a sharp surge when the distance is greater than 10m. Fig. 21(b) shows that the PER decreases when the RSS is increasing. In particular, the PER can be lower than 20% when the RSS is greater than -39dBm. The impact of SNR is demonstrated in Fig. 21(c), in which we deploy a USRP N210 to send AWGN between the transmission pair. It can be seen that the PER is approximately lower than 38% when the SNR is greater than 13.98dB.



(a) PER v.s. Distance (b) PER v.s. RSS (c) PER v.s. SNR

Fig. 21. Wi-Fi-to-Wi-Fi Performance (COTS Devices)

In summary, the environmental factors will cause similar trends in terms of error rate for both Wi-Fi and ZigBee downlink decoding, among which the lower SNRs overwhelm other factors to impact the decoding accuracy. The above thorough analysis on both platforms demonstrates the effectiveness of our developed schemes for achieving parallel communication.

VI. Related Work

A. Cross-Technology Communication

Recent advances in CTC can be classified into two categories: packet-level CTC and physical-level CTC.

Packet-level CTC. This line of works use packet level information (e.g. packet duration [2], [7], beacon interval [25], energy pattern [8], [9], [26], energy level [10], [11]) as minimal unit to construct special pattern that can be detected by other technologies [27], [28]. Packet-level CTC usually has a low network throughput and long transmission delay.

Physical-level CTC. Existing studies in this domain try to establish direct physical layer communication via software-based signal emulation. WEBee [3] proposes the first Wi-Fi to ZigBee CTC technology by constructing the WiFi payload elaborately so that the waveform resembles that of ZigBee signals. WIDE [4] proposes a novel Wi-Fi to ZigBee CTC technology by digital emulation. Besides, a series of works in [29], [30] investigate the physical-level emulation between Bluetooth and ZigBee/WiFi. Physical-level CTC can achieve the maximum transmission rate defined by the standard, but also faces some problems, such as emulation error, asymmetric link [28], and emerging security threats [31]–[34].

B. Parallel Communication

B^2W^2 [35], [36] propose a communication framework that enables concurrent communication among Wi-Fi and Bluetooth devices. Chiron [37] presents a communication technique that enables concurrently transmitting 1 stream of Wi-Fi data and up to 4 streams of ZigBee data to commodity Wi-Fi and ZigBee devices without causing interference. PIC [12] propose a parallel inclusive communication method that leverages the unique modulation schemes of Wi-Fi and BLE for bi-directional transmission of both Wi-Fi and BLE data at the same time within the overlapped channel. PMC [38] proposes a parallel communication technique from 1 Wi-Fi transmitter to 1 ZigBee receiver and 1 Wi-Fi receiver. Amphista [39] proposes a parallel communication technique that enables a ZigBee device to send out two different pieces of information to both the Wi-Fi gateway and another ZigBee device.

To the best of our knowledge, PMC [38] and PIC [12] are the only works close to our design, but their designs require hardware modification, which prevents them from being implemented easily on COTS devices.

VII. Conclusion

In this paper, we propose a physical-level parallel inclusive communication paradigm that leverages unique bit embedding strategies to achieve parallel decoding at both Wi-Fi and ZigBee receivers within an overlapping spectrum. Our proposed scheme is a fully software-level design, which can be implemented on many existing COTS devices to enhance throughput. The developed system is able to support the maximum performance of legacy ZigBee communication while providing up to 3.75Mbps extra Wi-Fi throughput simultaneously. We extensively evaluate our system on both USRP and COTS platforms, and the experimental results demonstrate that our design outperforms most of the Wi-Fi to ZigBee CTC schemes in terms of throughput, decoding accuracy, and feasibility.

REFERENCES

[1] S. R. Department. Iot connected devices worldwide 2030. [Online]. Available: https://www.statista.com/statistics/802690/worldwide-connected-devices-by-access-technology/

[2] K. Chebrolu and A. Dhekne, "Esense: communication through energy sensing," in *Proceedings of the 15th annual international conference on Mobile computing and networking*. ACM, 2009, pp. 85–96.

[3] Z. Li and T. He, "Webee: Physical-layer cross-technology communication via emulation," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. ACM, 2017, pp. 2–14.

[4] X. Guo, Y. He, J. Zhang, and H. Jiang, "Wide: physical-level ctc via digital emulation," in *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2019, pp. 49–60.

[5] X. Guo, Y. He, X. Zheng, Z. Yu, and Y. Liu, "Lego-fi: Transmitter-transparent ctc with cross-demapping," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 2125–2133.

[6] S. Wang, S. M. Kim, and T. He, "Symbol-level cross-technology communication via payload encoding," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2018, pp. 500–510.

[7] Y. Zhang and Q. Li, "Howies: A holistic approach to zigbee assisted wifi energy savings in mobile devices," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 1366–1374.

[8] Z. Yin, W. Jiang, S. M. Kim, and T. He, "C-morse: Cross-technology communication with transparent morse coding," in *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*. IEEE, 2017, pp. 1–9.

[9] W. Jiang, Z. Yin, S. M. Kim, and T. He, "Transparent cross-technology communication over data traffic," in *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*. IEEE, 2017, pp. 1–9.

[10] X. Guo, X. Zheng, and Y. He, "Wizig: Cross-technology energy communication over a noisy channel," in *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*. IEEE, 2017, pp. 1–9.

[11] X. Zheng, Y. He, and X. Guo, "Stripcomm: Interference-resilient cross-technology communication in coexisting environments," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 171–179.

[12] Z. Chi, Y. Li, X. Liu, Y. Yao, Y. Zhang, and T. Zhu, "Parallel inclusive communication for connecting heterogeneous iot devices at the edge," in *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*. ACM, 2019, pp. 205–218.

[13] C.-J. M. Liang, N. B. Priyantha, J. Liu, and A. Terzis, "Surviving wi-fi interference in low power zigbee networks," in *Proceedings of the 8th ACM conference on embedded networked sensor systems*, 2010, pp. 309–322.

[14] J. Huang, G. Xing, G. Zhou, and R. Zhou, "Beyond co-existence: Exploiting WiFi white space for Zigbee performance assurance," in *Proceeding of the 18th IEEE International Conference on Network Protocols (ICNP)*. IEEE, 2010, pp. 305–314.

[15] X. Zhang and K. G. Shin, "Enabling coexistence of heterogeneous wireless systems: Case for ZigBee and WiFi," in *Proceedings of the 12th ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc)*, 2011, pp. 1–11.

[16] S. Gollakota, F. Adib, D. Katabi, and S. Seshan, "Clearing the RF smog: making 802.11n robust to cross-technology interference," in *Proceedings of the ACM SIGCOMM*, 2011, pp. 170–181.

[17] Q. Yan, H. Zeng, T. Jiang, M. Li, W. Lou, and Y. T. Hou, "Jamming resilient communication using MIMO interference cancellation," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 11, no. 7, pp. 1486–1499, 2016.

[18] D. Halperin, M. J. Ammer, T. E. Anderson, and D. Wetherall, "Interference Cancellation: Better Receivers for a New Wireless MAC," in *Proceeding of the 6th ACM Workshop on Hot Topics in Networks (HotNets)*, 2007.

[19] S. Gollakota, S. D. Perli, and D. Katabi, "Interference alignment and cancellation," in *Proceedings of the ACM SIGCOMM*, 2009, pp. 159–170.

[20] S. Katti, S. Gollakota, and D. Katabi, "Embracing wireless interference: Analog network coding," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 397–408, 2007.

[21] V. Iyer, V. Talla, B. Kellogg, S. Gollakota, and J. Smith, "Inter-technology backscatter: Towards internet connectivity for implanted devices," in *Proceedings of the 2016 ACM SIGCOMM*, 2016, pp. 356–369.

[22] Matlab. Rate 2/3 convolutional code in awgn. [Online]. Available: https://www.mathworks.com/help/comm/ug/rate-23-convolutional-code-in-awgn.html

[23] S. Dhaliwal, N. Singh, and G. Kaur, "Performance analysis of convolutional code over different code rates and constraint length in wireless communication," in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*. IEEE, 2017, pp. 464–468.

[24] P. Mathys. Theory and practice of error control codes. [Online]. Available: http://ecee.colorado.edu/~mathys/ecen5682/slides/convperf99.pdf

[25] S. M. Kim and T. He, "Freebee: Cross-technology communication via free side-channel," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 2015, pp. 317–330.

[26] Z. Chi, Z. Huang, Y. Yao, T. Xie, H. Sun, and T. Zhu, "Emf: Embedding multiple flows of information in existing traffic for concurrent communication among heterogeneous iot devices," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[27] W. Jiang, S. M. Kim, Z. Li, and T. He, "Achieving receiver-side cross-technology communication with cross-decoding," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. ACM, 2018, pp. 639–652.

[28] S. Wang, Z. Yin, Z. Li, and T. He, "Networking support for physical-layer cross-technology communication," in *2018 IEEE 26th International Conference on Network Protocols (ICNP)*. IEEE, 2018, pp. 259–269.

[29] W. Jiang, R. Liu, L. Liu, Z. Li, and T. He, "Bluebee: 10,000 x faster cross-technology communication from bluetooth to zigbee," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. ACM, 2017, pp. 486–487.

[30] Z. Li and Y. Chen, "Bluefi: Physical-layer cross-technology communication from bluetooth to wifi," in *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2020, pp. 399–409.

[31] S. Yu, X. Zhang, P. Huang, L. Guo, L. Cheng, and K. Wang, "Authctc: Defending against waveform emulation attack in heterogeneous iot environments," in *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, 2020, pp. 20–32.

[32] S. Yu, X. Zhang, P. Huang, and L. Guo, "Secure authentication in cross-technology communication for heterogeneous iot," in *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. IEEE, 2019, pp. 1–2.

[33] X. Zhang, P. Huang, L. Guo, and Y. Fang, "Hide and seek: Waveform emulation attack and defense in cross-technology communication," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 1117–1126.

[34] X. Zhang, P. Huang, L. Guo, and M. Sha, "Incentivizing relay participation for securing iot communication," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1504–1512.

[35] Z. Chi, Y. Li, H. Sun, Y. Yao, Z. Lu, and T. Zhu, "B2w2: N-way concurrent communication for iot devices," in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*. ACM, 2016, pp. 245–258.

[36] Z. Chi, Y. Li, H. Sun, Y. Yao, and T. Zhu, "Concurrent cross-technology communication among heterogeneous iot devices," *IEEE/ACM Transactions on Networking*, 2019.

[37] Y. Li, Z. Chi, X. Liu, and T. Zhu, "Chiron: Concurrent high throughput communication for iot devices," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2018, pp. 204–216.

[38] Z. Chi, Y. Li, Y. Yao, and T. Zhu, "Pmc: Parallel multi-protocol communication to heterogeneous iot radios within a single wifi channel," in *2017 IEEE 25th International Conference on Network Protocols (ICNP)*. IEEE, 2017, pp. 1–10.

[39] Z. Chi, Y. Li, Z. Huang, H. Sun, and T. Zhu, "Simultaneous bi-directional communications and data forwarding using a single zigbee data stream," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 577–585.